EXIT Chart Analysis for Compressive Turbo Codes

Bilal Riaz

Department of Electrical and Computer Engineering
McGill University
Montréal, Canada

May 2009

Submitted in partial fulfillment of the M. Eng. degree requirements.

Abstract

Turbo codes have achieved near Shannon limit performance in data communication over noisy channels. Recently introduced EXtrinsic Information Transfer (EXIT) Charts [15] have become an essential part of turbo code design and have also been used as a complementary design tool for the traditional bit error rate simulations. Additionally, compressive turbo codes have been shown to achieve near-entropy performance in different source coding problems [1], [46], [74]. The main objective of this thesis is an extension of EXIT charts from turbo channel codes to turbo source codes, as well as extension of this technique to analog and finite precision iterative decoders.

After the initial review of relevant literature and research results, the EXIT charts technique is extended to performance analysis of compressive turbo codes. As opposed to previous attempts at such an extension in [24] and [32], the extension derived in this thesis gives reliable results that closely match actual simulated performance of compressive turbo codes. Furthermore, a lower bound is obtained in order to illustrate the connection and differences between the compressive EXIT chart technique in this thesis and the one previously proposed in [32]. Finally, the EXIT chart technique is extended to performance analysis of analog iterative decoders and digitally implemented turbo decoding algorithms based on finite precision arithmetics.

Various numerical results have been obtained to illustrate the successful application of the derived EXIT chart technique. Firstly, the derived EXIT chart performance analysis is used for several single source compression schemes based on both parallel and serial concatenated encoders. Consequently, the EXIT chart technique developed in this thesis is further applied to distributed turbo source coding schemes for the Slepian-Wolf and Wyner-Ziv problems. Finally, the designed EXIT charts are used to analyze the performance of turbo compressive systems with finite precision decoders. The effect of finite precision arithmetics has been studied for turbo channel codes in [70] but not for turbo compressive codes. Furthermore, in all these cases, a good match between the simulated system performance (the start of the water-fall region) and the performance predicted by the EXIT chart technique is observed.

Sommaire

Les turbo codes ont atteint des performances près de la limite de Shannon dans la communication de données sur des canaux bruités. Les diagrammes EXIT (*EXtrinsic Information Transfer*) [15] récemment proposés sont devenus une partie essentielle de la conception de turbo codes et ont été utilisés en tant que substituts aux simulations traditionnelles du taux d'erreur binaire. Il a été démontré que les turbo codes compressés atteignent des performances près de l'entropie dans différents problèmes de codage de source [1], [46], [74]. L'objectif principal de cette thèse est d'étendre les diagrammes EXIT des turbo codes pour canaux aux turbo codes pour sources et également étendre cette technique aux décodeurs itératifs de taille fixé.

Après une revue initiale de littérature et recherche pertinent, la technique de EXIT charts est augmenté à l'analyse de performance de codes turbo compressés. Au contraire de la dernière tentative d'une telle augmentation [24] et [32], l'augmentation dérivée dans cette thèse donne des résultats fiables qui correspondents à la performance simulée de codes turbo compréhensives. De plus, une borne inferieure est obtenue pour illustrer la connexion et les différences entre la technique de EXIT charts de cette thèse et de celle proposé dans [32]. Finalement, la technique de EXIT charts est augmenté à l'analyse de performance de décodeurs analogues itératifs et l'implémentation numérique de décodage turbo basée sur l'arithmétique de précision finie.

Plusieurs résultats numériques ont été obtenus pour illustrer l'application réussie des dérivés de la technique du graphe EXIT. Premièrement, l'analyse de la performance du graphe EXIT dérivées est utilisée pour plusieurs seules sources de compression basée sur des encodeurs concaténés en parallèle et en série. En conséquence, la technique des graphes EXIT développée dans cette thèse a été appliquée dans codage de source turbo distribué pour les problèmes de Slepian-Wolf et Wyner-Ziv. Enfin, les graphes EXIT conçus ont été utilisés pour analyser la performance des systèmes de turbo compresseur avec les décodeurs d'une précision finie [70]. Dans tous ces cas, un bon accord entre la simulation de la performance du système (le début de la chute de la région 'water-fall') et les résultats prédits par la technique des graphes EXIT a été observé.

Acknowledgements

I would like to express my sincere gratitude and deep admiration to my advisor, Professor Jan Bajcsy, for his excellent guidance and continuous encouragements throughout the course of my Masters program. First and foremost, I would like to thank him for introducing me to the topic of data compression using turbo codes. Without him, none of this work could have been possible.

I would also like to thank my lab colleagues, Aminata Amadou Garba and Yong Jin Daniel Kim for the inspiration and support over the past years. I am grateful for the financial supports offered by my advisor, Prof. Jan Bajcsy. In addition, I would like to thank the National Science and Engineering Research Council (NSERC) for awarding me with the CGS-M scholarship and hence providing me with financial support from May 2008 and lasting till the third week of September 2008.

I would also like to take this oppurtunity to thank Nortel Networks (Canada) for allowing me to finish and submit my thesis for the initial submission in October 2008 while working for them full-time. Finally, I dedicate this report to my family whose unconditional support and firm belief in me has encouraged me throughtout my degree.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 A Brief Overview of Telecommunications

Early telecommunication involved the use simple techniques (e.g., smoke signals, drums, flags etc.) to transmit information. These days, modern telecomunication primarily encompasses the use of electronic and optical transmitters in telephone, television, radio, computer and cell phone systems. An important aspect of a telecommunication system is the transmitting medium which can be wireline, wireless or optical. During the past two decades, there has been a tremendous increase in interest and research in these media because of the convenience and the need for broadband access.

The communication industry around the globe is striving for new technologies, which can help them increase data rates, reduce the operating cost and improve their dtaa services to cater to the increasing customer demand. Future digital communication systems are required to support increased user data rates, offer seamless user access to the backbone communication network and integrate multiple services, such as, downloading and uploading movies and videos, accessing electronic library resources, video conferencing, surveillance for security purposes, telemedicine and telerobotics [5], [51]. Intensive research is currently underway to develop technologies that will meet the data rate requirements of these services. Powerful data compression techniques along with error control codes are two of the most promising approaches to facilitate the above applications [51], [52].

1.2 Architectures of Digital Communication Systems

Advances in integrated circuit technology and digital communication systems over the years enabled personal data communications to become practical, economical and widespread. These systems rely on data compression and error control coding to achieve higher spectral efficiency, noise robustness and increased fidelity [51], [52]. Many different digital communication standards have been developed during the past three decades for wireless and wireline transmission of voice and data, such as, the GSM and IS-54 second generation (2G) digital wireless cellular standards using rate ½ convolutional codes along with advanced speech compression techniques. These 2G systems replaced the old first generation (1G) analog cellular phone technology based mostly on frequency modulation, such as the Advanced Mobile Phone System (AMPS), which used multiple repititions of data as well as BCH block codes to perform error correction. A continued need for advanced services led to the research, development and eventual deployment of the third generation (3G) wireless systems during the late 1990's [40] which use a high class of error correction codes known as turbo codes. Similarly, there has been a steady improvement in phone-line modem technology that resulted in standards that allow fast (up to several Mbits per second) Internet access in residential homes and businesses.

Fig. 1.1 shows a block diagram of a digital communication system architecture used for point-to-point communication in the abovementioned standards. The data source in Fig. 1.1 generates messages which are transmitted to the receiver over a physical medium. The source encoder removes unnecessary information from the source data (via compression), while exploiting the source redundancy and statistics. The channel encoder, on the other hand, adds redundancy into the data to make them more robust against errors and impairments encountered on the channel. (The channel impairments could include noise, fading, distortion and interference from other users.) The modulator essentially prepares the encoded signal to be transmitted through the physical medium by formatting the data according to the properties of the medium. This typically involves pulse shaping and up conversion from baseband to passband frequency range. The receiver modules, shown in Fig. 1.1, invert the operations performed by the corresponding transmitter blocks, i.e., de-modulator inverts the function of the modulator to get back the coded signal which is then passed through the channel decoder. This decoder removes redundunacy which is added by the channel encoder at the transmitting side. Finally, the source decoder decompresses the data before passing it on.

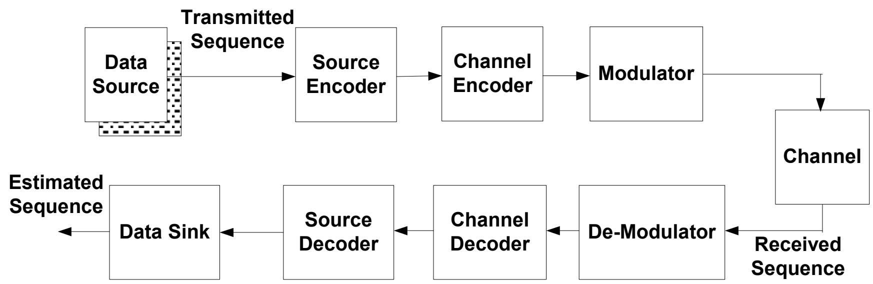


Fig. 1.1 Architecture of a typical digital communication system [27]

Another type of architecture may be utilized in a communication network, where a group of (correlated) information source nodes needs to communicate data to a common sink node as shown in Fig. 1.2. A classic example is a distributed sensor array communication system consisting of sensors which capture a common scene or acoustic signal independently. These sensors transmit their information, which is highly correlated, to a central unit (a sink or a fusion centre) that uses all the received data to form the best picture of the particular scene or estimate of the acoustic signal. If all the sensors could communicate with each other, then redundant information observed by the sensors could be eliminated. However, this would not be practical in most cases, as it would need an elaborate interconnected sensor network along with substantial bandwidth and energy requirements. Hence, the distributed source coding problem involves the design of effective compression systems which optimally compress separate source nodes data, assuming the nodes cannot communicate with each other, but are decoded jointly, as shown in Fig. 1.2.

In this context, Slepian and Wolf (SW) showed a surprising result [61] that in case of distributed compression of two discrete sources X and Y (shown in Fig. 1.2. (a)) it is theoretically possible to achieve the same data compression performance as when the sources are compressed jointly. Wyner considered a special case of the above SW problem when the joint decoder is only interested in recovering X, as shown in Fig. 1.2. (b). In this case, Y serves as side-information about X and is available at the decoder only. Again, it turns out that one can achieve the same performance with or without side information at the transmitter.

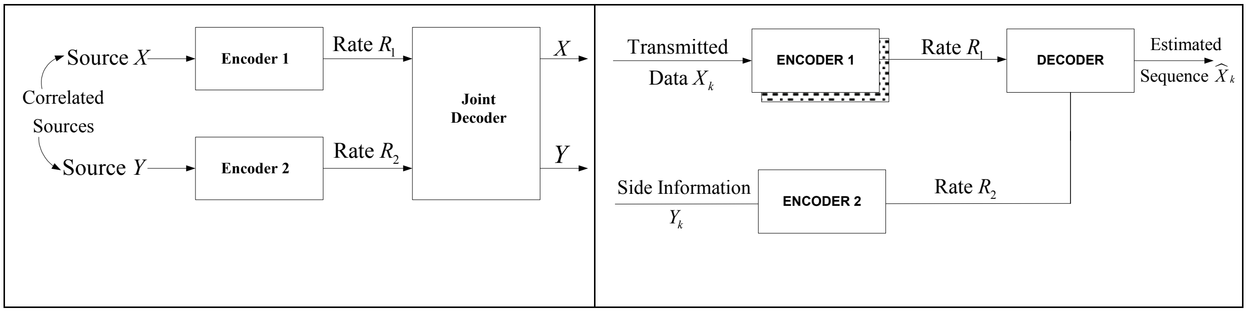


Fig. 1.2. (a) (Left) Block diagram of the distributed compression problem considered by Slepian/Wolf **(b)** (Right) Block diagram of a source coding scheme of with side information at the receiver only, as considered by Wyner

1.3 Motivation for Data Compression

Data compression (source coding) is a very important area of research nowadays, as a major part of the transmission of data over a channel is to find ways to send as little data as possible, such that it is recovered properly at the receiver. The amount of available data is increasing, due to new multimedia content generated and available from High Definition Television (HDTV), online electronic databases, streaming video portals, etc. The limited amount of practical storage space and physical bandwidth makes it a challenging problem to deliver the data to the end users. One solution to this problem is to store the data more efficiently using advanced data compression techniques as less data is easier to store, handle and transmit. For text files, 50 to 70% compression rates are possible, whereas for images, up to 90% compression can be achieved. Following is a list of some of the common data compression applications along with their data compression algorithms and codes [19], [76]:

- WinZip in Windows: Lempel-Ziv algorithm
- PDF: Lempel-Ziv, Run-length and Huffman codes
- JPEG: Huffman codes
- FAX: Run-length coding

Despite of the different compression techniques used in practice, the entropy and typical sequences are the two fundamental concepts that form the foundation of data compression. The *entropy*, further discussed in Section 2.1, is a quantity that shows a limit on how much data compression can be performed on a particular data source. *Typical sequences*, discussed in Section 2.2, are the source sequences, which occur most of the time at the output of a data source. Non-typical sequences, on the other hand, occur with very small probability and hence do not contribute much to the efficiency of data compression. One of the main approaches to data compression is to focus on effectively encoding the typical sequences, as it is done in the Lempel-Ziv algorithm [75] or in the recently proposed turbo compression schemes [1], [46], [74]. The latter technique will be the focus of this thesis and will be explained in detail in Section 3.3.2 and throughout Chapter 4.

1.4 Overview of Turbo Codes

Concatenated codes were first proposed as a method for achieving large coding gains by combining two or more simple component codes [26]. Their earlier versions have been successfully used in deep space communication systems and data storage. A high class of error correcting codes known as Turbo codes were first introduced by Berrou, Glavieux and Thitimajshima in 1993 [11] and used the concept of iterative decoding. They were the first practical coding technique that came close to approaching the Shannon capacity limit for a noisy Gaussian channel, e.g., within about 0.1 dB in [16]. They have been recently used to perform successful near-entropy data compression [1], [46], [74].

Turbo codes are built like a crossword puzzle, i.e., one encoder provides a set of clues about "rows" of the transmitted message data, then the interleaver changes the data order to a "columns" format and finally the second encoder generates the "column" clues. The iterative turbo decoding process proceeds in a manner that is similar to solving a crossword puzzle, i.e., one decoder is responsible for using the "row" clues and the other decoder is responsible for using the "column" clues. Both decoders keep exchanging their guesses about the decoded message symbols in order to successfully complete the "decoding" process. This procedure continues for a prescribed number of iterations or until the decoders converge to the same solution.

A typical turbo coding system is shown in Fig. 1.3 where π and π^{-1} are the interleaving and de-interleaving functions respectively. To generate the hints, the two

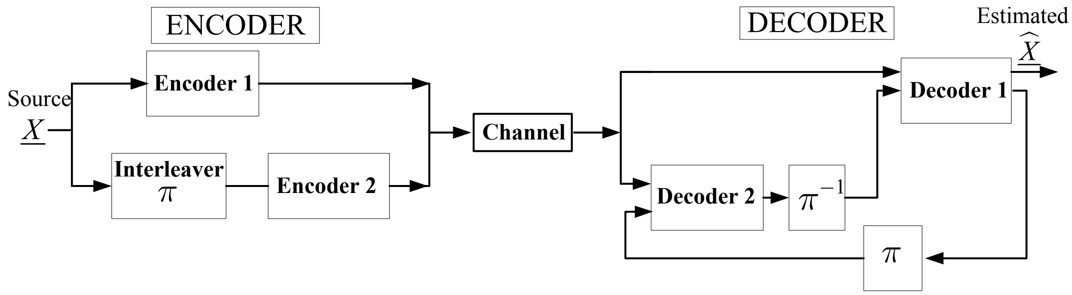


Fig. 1.3. Parallel Concatenated Turbo Encoder/Decoder system

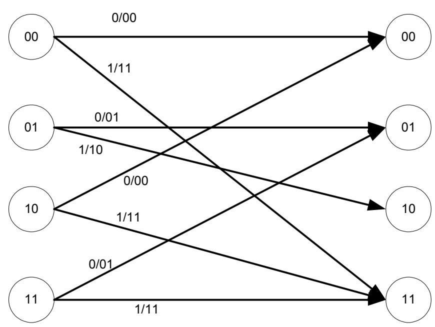


Fig. 1.4. One stage of a generic trellis structure

constituent encoders in a turbo code usually employ a trellis structure to encode the message data. An example of a trellis is shown in Fig. 1.4 which shows edges which start and end in a particular state. For example, if the input is *1* and the current state being *00*, following the edge for a *1*, the output is *11* and the destination state is *11* as can be seen from Fig. 1.4. Note that there is one trellis stage for one incoming bit in this case, thus n stages would be needed to encode/decode a sequence containing n bits. Turbo encoding and iterative decoding will be described in further detail in Section 2.4.

1.5 Thesis Overview and List of Contributions

The main objective of this thesis is the performance analysis of iteratively decoded concatenated turbo codes with the main emphasis on compressive turbo codes in different scenarios (e.g., infinite and finite precision algorithm implementation, Slepian-Wolf and Wyner-Ziv coding problems, etc.). In particular, the EXIT chart technique, originally developed for turbo channel codes in [15], is extended to various problems of interest. Consequently, selected analytic and extensive numerical results are presented.

The remainder of this thesis is structured as follows. Chapter 2 reviews the relevant material, concepts and results from turbo coding literature and information theory that will be used to develop the main results of this work. Chapter 3 gives the details of the developed EXIT chart performance evaluation technique, i.e., it shows how they can be generalized to performance analysis of compressive turbo codes. Consequently, Chapter 4 will describe the various applications of the developed EXIT chart analysis technique to single source turbo compression, Slepian-Wolf compression and Wyner-Ziv coding using turbo codes. Additionally, it will also cover the performance analysis of turbo decoder's implemented using finite precision arithmetics and analog VLSI techniques. Finally, Chapter 5 provides the concluding remarks and outlines possible directions of future research work.

Individual contributions of this thesis have been published in part in references [56] and [58] and can be listed as follows:

- Chapter 3 presents analytic and semi-analytic results that extend the EXIT charts technique to compressive turbo codes and contains the following:
    1. Review of the original EXIT chart technique for turbo channel codes, including its detailed development along with a step-by-step methodology.
    2. Generalization of the EXIT chart technique to compressive turbo codes, including detailed mutual information derivation and methodology.
    3. A detailed comparison of the derived compressive EXIT chart technique to the one proposed by Hagenauer [32] in order to illustrate their differences as well as the ability of the method derived in this thesis to provide significantly better performance evaluation results.
    4. Derivation of a lower bound on the mutual information used in the construction of compressive EXIT charts. This lower bound illustrates that the EXIT chart technique from [32] leads to inexact results and can only be interpreted as a very loose bound on actual EXIT curves.
    5. Development of EXIT charts for recently proposed analog turbo decoders. The key results show that an analog iterative turbo decoder will never leave

the region outlined by the analog EXIT curve constructed in Section 3.4.2, thus allowing one to determine convergence of such a decoder.

- Chapter 4 contains several numerical results illustrating the functionality and usefulness of the developed compressive EXIT charts in the following scenarios:
  i. Single source turbo data compression at various rates.
  ii. Slepian-Wolf coding using compressive turbo codes in a distributed setting.
  iii. Practical Wyner-Ziv coding schemes based on previously proposed extension of the Slepian-Wolf problem.
  iv. Performance analysis of finite precision iterative turbo decoders.

  In all of the above cases, the developed EXIT chart analysis led to results that matched the actual simulated performance of the studied compressive turbo codes.

Chapter 2 Preliminaries

Information theory, the mathematical theory of communication, was founded by Claude E. Shannon in 1948. Information theory gives fundamental limits on the optimum performance achievable in various communication problems such as data transmission over a noisy channel, compression of data, quantization of analog information sources, data storage, etc. However, it is the area of coding theory and techniques that is concerned with practical means to achieve these theoretical limits in practical communication.

To make this thesis self-contained and to allow in-depth performance analysis of compressive turbo coded systems, definitions and selected information theory concepts are reviewed in Section 2.1-2.3. Furthermore, Section 2.4 describes encoding and iterative decoding of turbo codes as well as their usage in different communication scenarios.

2.1 Information Measures: Entropy and Mutual Information

Since EXIT charts modeling and construction uses several concepts from information theory, these concepts will be reviewed in this subsection. This exposition is an adaptation of the material from the classic textbook by Cover and Thomas [19] and the course notes from a graduate course on information theory [41].

For a discrete random variable *X* described by its probability mass function P_X and taking values on a discrete alphabet $\mathcal{X}$, the *entropy* of a discrete random variable *X* is defined as,

$$H(X) = E\left(\log_2 \frac{1}{P_X(x)}\right) = -\sum_{x\in\mathcal{X}} P_X(x)\log_2(P_X(x)) \quad \text{bits per source symbol}. \tag{2.1}$$

The entropy *H(X)* represents the amount of uncertainty about the discrete random variable *X*, i.e., the average amount of information resolved by observation of a specific realization of *X*.

Theorem 2.1: The entropy of a discrete source X that takes values on alphabet $\mathcal{X} = \{a_1, a_2, \cdots, a_M\}$ is always non-negative and does not exceed $\log_2 M$, i.e.,

$$0 \leq H(X) \leq \log_2 M \tag{2.2}$$

where the first inequality occurs if and only if X is deterministic (note that we make the standard assumption that $0\log_2 0 = 0$) and the second inequality if and only if all source symbols are equally likely ($P_X(x = a_m) = p_m = \frac{1}{M}$, $m = 1, 2, \cdots, M$).

The non-negativity of entropy can be deduced from (2.1) using the fact that $0 \leq P_X(x) \leq 1$ and thus $0 \leq -P_X(x)\log_2(P_X(x))$ for all $x \in \mathcal{X}$. As a consequence of this theorem, we know that for any integer N

$$2^{NH(X)} \leq 2^{N\log_2 M} \tag{2.3}$$

Lemma 2.1:: The relative entropy between two probability mass functions $P_X(x)$ and $Q_X(x)$ with $x \in \mathcal{X}$ is defined as:

$$D(P \| Q) = \sum_{x \in \mathcal{X}} P_X(x)\log_2\left(\frac{P_X(x)}{Q_X(x)}\right) = E_P\left(\log_2 \frac{P_X}{Q_X}\right) \tag{2.4}$$

where the standard assumption is used that $0\log_2 \frac{0}{Q(\bullet)} = 0$ and $P(\bullet)\log_2 \frac{P(\bullet)}{0} = \infty$.

Then:

$$D(P \| Q) \geq 0 \tag{2.5}$$

with equality if and only if $P_X(x) = Q_X(x)$ for all x .

Proof of Lemma: The proof of this Lemma can be found in Appendix A.1.

Proof of Theorem: The proof of this Theorem can be found in Appendix A.1.

If we consider two discrete random variables X and Y defined on the discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$ respectively, with a joint probability mass function $P_{XY}(x,y)$, then the *joint entropy* of X and Y is given by:

$$H(X,Y) = E_{XY}\left(\log_2 \frac{1}{P_{XY}(x,y)}\right) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2(P_{XY}(x,y)) \tag{2.6}$$

Also, the *conditional entropy* of X given a specific realization $Y=y$ is defined as follows:

$$H(X \mid Y=y) = E_{X|Y=y}\left(\log_2 \frac{1}{P_{X|Y}(x \mid y)}\right) = -\sum_{x\in\mathcal{X}} P_{X|Y=y}(x \mid y)\log_2(P_{X|Y=y}(x \mid y)) \tag{2.7}$$

The (average) conditional entropy of X given Y is defined as

$$\begin{aligned} H(X \mid Y) &= E_Y\left(H(X \mid Y=y)\right) = -\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} P_Y(y)P_{X|Y}(x \mid y)\log_2(P_{X|Y}(x \mid y)) \\ &= -\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} P_{XY}(x,y)\log_2(P_{X|Y}(x \mid y)) = E_{XY}\left(\log_2 \frac{1}{P_{X|Y}(x \mid y)}\right) \end{aligned} \tag{2.8}$$

For the similar reasons as the entropy, the abovementioned conditional entropies are non-negative, i.e.,

$$\begin{aligned} &H(X \mid Y=y) \geq 0 \\ &H(X \mid Y) \geq 0 \end{aligned} \tag{2.9}$$

Theorem 2.2: The chain rule for the join entropy states that

$$H(X,Y) = H(X) + H(Y \mid X).$$

Proof of Theorem: The proof of this theorem can be found in Appendix A.1.

The chain rule directly implies some other useful identities about entropies:

$$\begin{aligned} H(X,Y) &= H(X) + H(Y \mid X) \geq H(X) \\ H(X,Y) &= H(Y) + H(X \mid Y) \geq H(Y) \end{aligned} \tag{2.10}$$

with equality if and only if Y is completely determined by X. Also if Y is a deterministic function of X

$$H(X) \geq H(Y) \tag{2.11}$$

Theorem 2.3: $H(X \mid Y) \leq H(X)$ with equality if and only if X and Y are independent. This implies that conditioning (extra or side information) reduces entropy. Note that X and Y can be interchanged without the loss of any generality.

Proof of Theorem:

$$H(X) - H(X \mid Y) = E_{XY}\left(\log_2 \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)}\right) \overset{\text{Using Lemma 2.1}}{=} D\left(P_{XY} \,\|\, P_X P_Y\right) \overset{\text{Using Lemma 2.1}}{\geq} 0$$
$$H(X) - H(X \mid Y) \geq 0 \quad \rightarrow \quad \therefore\ H(X \mid Y) \leq H(X) \tag{2.12}$$

□

Finally, the *union bound* for the joint entropy states that

$$H(X,Y) \leq H(X) + H(Y) \tag{2.13}$$

with equality if and only if X and Y are independent.

The quantity $H(X) - H(X \mid Y)$ is the rate of information transfer or the average information provided about X by observing Y. This quantity is known as the *mutual information* $I(X,Y)$. The units can be in bits.

$$\begin{aligned}
I(X,Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2\left(\frac{P_{XY}(x,y)}{P_X(x)P_Y(y)}\right) \\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2\left(\frac{P_{X|Y}(x\mid y)}{P_X(x)}\right) \\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2\left(P_{X|Y}(x\mid y)\right) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2\left(P_X(x)\right) \\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2\left(P_{X|Y}(x\mid y)\right) - \sum_{x\in\mathcal{X}} P_X(x)\log_2\left(P_X(x)\right) \\
&= -\left(-\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P_{XY}(x,y)\log_2\left(P_{X|Y}(x\mid y)\right)\right) + \left(-\sum_{x\in\mathcal{X}} P_X(x)\log_2\left(P_X(x)\right)\right) \\
&= -H(X\mid Y) + H(X) = H(X) - H(X\mid Y)
\end{aligned} \tag{2.14}$$

The above implies that the mutual information is the amount by which the uncertainty in X is removed by the knowledge of Y. Also it can be shown, due to symmetry, that $I(X,Y)=H(Y)-H(Y\,|\,X)$.

Also, from Theorem 2.3 (and (2.12)) it can be deduced that mutual information is non-negative i.e.

$$I\left(X,Y\right)=H(X)-H(X\,|\,Y)\geq 0 \tag{2.15}$$

2.2 Typical Sequences and Source Coding for a Single User

The concept of typical sequences is of great use in the field of data compression as it provides a link between the the theoretical limit of compression and the entropy of a data source. This thesis will make use of this important connection when presenting theoretical and numerical results. Furthermore, this thesis will mostly be concerned with discrete memoryless (data) sources using the following definition:

Definition 2.1: A discrete memoryless source is defined as a random vector $\underline{X}=X_1,X_2,\cdots,X_N$ where X_n are independent identically distributed (i.i.d.) RV's with alphabet $\mathcal{X}=\{a_1,a_2,\cdots,a_M\}$ and probability mass function (PMF) given by $P_X=\{p_1,p_2,\cdots,p_M\}$ where $p_m=\Pr\{X=a_m\}=p_X\left(a_m\right)$. $\underline{X}=X_1,X_2,\cdots,X_N$ is considered as the extension of the source X which outputs sequences from the set $\mathcal{X}^N$ i.e. $\mathcal{X}^N$ is the set of all the realizations of $\underline{X}$. A particular realization of the random vector $\underline{X}$ is denoted as $\underline{x}$ such that $\underline{x}=\left[x_1,x_2,\cdots,x_N\right]\in\mathcal{X}^N$.

In addition, let $N\left(a_m\,|\,\underline{x}\right)$ denote the number of times that symbol a_m appears in the sequence $\underline{x}$.

Definition 2.2: For a discrete memoryless stationary source $\underline{X}$, sequence $\underline{x}$ of length *N* is called a typical sequence for a given $\delta > 0$ if $\left| \frac{1}{N} N\left(a_m \mid \underline{x}\right) - p_m \right| \leq \frac{\delta}{M}$, $m = 1, 2, \cdots, M$. For the given source $\underline{X}$ and a given $\delta > 0$, let $T_N^{\underline{X}}(\delta)$ denote the set of all typical sequences of length *N*.

Theorem 2.4: (*AEP: Asymptotic Equipartition Property)*

For any discrete memoryless stationary source $\underline{X}$ and for any $\delta > 0$, the following three results are true for the typical sequences:

a) $\Pr\left[\underline{x} \in T_N^{\underline{X}}(\delta)\right] \underset{N \to \infty}{\rightarrow} 1$.

b) If $\underline{x}$ is a typical sequence of a discrete stationary memoryless source, then $2^{-N\{H(X)+\delta\}} \leq P_{\underline{X}}(\underline{x}) \leq 2^{-N\{H(X)-\delta\}}$ with an arbitrary small $\delta > 0$. In other words, $P_{\underline{X}}(\underline{x}) \approx 2^{-NH(X)}$.

c) $(1-\delta) \cdot 2^{N(H(X)-\delta)} \leq \left| T_N^{\underline{X}}(\delta) \right| \leq 2^{N(H(X)+\delta)}$

   where $\left| T_N^{\underline{X}}(\delta) \right|$ is the number of elements in the typical set $T_N^{\underline{X}}(\delta)$.

This AEP implies that the source *X* outputs most of the time only about $2^{NH(X)}$ *nearly* equally probable length-*N* sequences from all possible $M^N = 2^{N \log_2 M}$ sequences that can be generated.

Proof of Theorem: The AEP is a consequence of the law of large numbers and its proof is included in Appendix A.1.

Definition 2.3: A $\left(2^{NR}, N\right)$ *block source code*, with integer *N>0*, is specified by an encoder *g* which is a mapping of sequences from $\mathcal{X}^N$ to the set of integers

$\{0,1,2,\cdots,2^{NR}-1\}$ and a decoder *f* such that $f:\{0,1,2,\cdots,2^{NR}-1\}\rightarrow\mathcal{X}^N$. In other words, the decoder *f* maps the set of integers $\{0,1,2,\cdots,2^{NR}-1\}$ back to the sequences in $\mathcal{X}^N$.

The probability of error of a code is defined as $P_e^{(N)}=P\left[f\left(g\left(\underline{x}\right)\right)\neq\underline{x}\right]$. For a given source, a rate R is said to be achievable if there exists a sequence of $\left(2^{NR},N\right)$ source codes with $P_e^{(N)}\rightarrow 0$ as $N\rightarrow\infty$.

Theorem 2.5: *(Source Coding Theorem)*

Given a discrete memoryless source *X* with a prescribed compression rate $R^*>H(X)$ bits and any error probability target $\varepsilon>0$, there exists a $\left(2^{NR},N\right)$ source code such that $R\leq R^*$ and $P_e^{(N)}=P\left[f\left(g\left(\underline{x}\right)\right)\neq\underline{x}\right]<\varepsilon$ for sufficiently large *N*.

"Interpretation": Shannon's noiseless source coding theorem says that if *R>H(X)*, the source can be encoded using a block source code with a vanishingly small probability of decoding error or failure when $N\rightarrow\infty$.

Proof of Theorem: We will construct such an encoder/decoder pair using the typical sequences $T_N^X(\delta)$ from Theorem 2.4. The following encoding procedure is used:

if $\underline{x}\in T_N^X(\delta)$, then $\underline{x}$ is mapped 1-to-1 into a distinct codeword in {0, 1, ..., 2^{NR}-1}.

if $\underline{x}\notin T_N^X(\delta)$, then $\underline{x}$ is mapped into a fixed codeword which could already be assigned to a typical sequence.

The corresponding decoder recovers the typical sequences corresponding to codeword $c\in$ {0, 1, ..., 2^{NR}-1}. but it always makes an error if a sequence is not typical. Thus, using Theorem 2.1.4:

If $\underline{x}\in T_N^X(\delta)$ then for a sufficiently large *N*, it follows that

$$\left|T_N^X(\delta)\right|\leq 2^{N\{H(X)+\delta\}} \tag{2.16}$$

Such that if $R = \lceil H(X)+\delta$, where $\lceil \bullet$ denotes the upper integer part, there are enough codewords for all the typical sequences.

The described decoder makes an error only if a non-typical sequence was generated, thus

$$\Pr\left\{\text{decoding failure}\right\} = \Pr\left\{\underline{x} \notin T_N^{\underline{X}}(\delta)\right\} \underset{N\to\infty}{\to} 0 \tag{2.17}$$

Consequently, for some large enough *N*, this source code works better that prescribed.

□

The converse to this theorem states that if *R<H(X)*, then any source code will have a decoding failure probability that will tend to *1* when $N \to \infty$. This means that *H(X)* is a tight lower bound to the achievable code rate for reliable compression because any attempt of compression below entropy will result into an error floor. This result is stated without a proof.

Fixed-length-to-fixed-length source codes obey this theorem and turbo compressive codes mentioned later are one example of such codes. It is interesting to note that there are also fixed-to-variable length codes for which a similar theorem applies [19] in which case $E(L) \geq H(X)$ where $E(L)$ is the expected length of the codewords. Huffman codes are an example of such codes constructed and used in practice. There are two other possible approaches to compression, i.e., variable-to-fixed-length codes (e.g., run-length codes used in the fax machine [19]) and variable-to-variable length codes (e.g., the Lempel-Ziv algorithm used in the Windows based Winzip software [75]).

2.3 Two Multi-User Source Coding Problems

In most sensor networks, there are two (or more) correlated source nodes that wish to communicate with a common date fusion node. The goal of the multi-user or distributed source coding problem is to compress the sources, approaching the same optimal performance which is possible if the individual sources were able to communicate with each other. The following sub-sections will look at two such multi-user source coding problems.

2.3.1 The Slepian-Wolf Problem

Slepian and Wolf (SW) showed that in the case of distributed (separate) compression of two discrete memoryless sources *X* and *Y* followed by joint decoding, it is theoretically possible to achieve the performance of a system where the encoders can communicate with each other [61][1]. Wyner considered a specialization of the SW problem i.e. the case in which the decoder is primarily interested in decoding only one source *X* and has the other source *Y* available at the decoder as the side information [71].

The Slepian-Wolf problem deals with separate encoding of correlated sources, when joint recovery is performed at the receiver, as illustrated in Fig. 2.1 (a). Let *X* and *Y* be two discrete memoryless sources defined on alphabets $\mathcal{X}$ and $\mathcal{Y}$ respectively. Even though the two encoders cannot communicate with each other in this problem, the rates R_X and R_Y at which the sources *X* and *Y* can be successfully compressed and decompressed belong to an achievable rate region. Before defining this achievable rate region, some preliminary details are required.

Definition 2.4: Let *X* and *Y* be discrete memoryless sources whose outputs are taken from the sets $\mathcal{X} = \left\{a_1, a_2, \cdots, a_{M_X}\right\}$ and $\mathcal{Y} = \left\{b_1, b_2, \cdots, b_{M_Y}\right\}$. The (distributed) encoder pair consists of two functions g_X and g_Y such that $\mathrm{g}_X : \mathcal{X}^N \to \{0,1,2,\cdots,2^{NR_X} - 1\}$ for the source *X* and $\mathrm{g}_Y : \mathcal{Y}^N \to \{0,1,2,\cdots,2^{NR_Y} - 1\}$ for the source *Y*. The joint decoder is a mapping $f : \{0,1,2,\cdots,2^{NR_X} - 1\} \times \{0,1,2,\cdots,2^{NR_Y} - 1\} \to \mathcal{X}^N \times \mathcal{Y}^N$.

Definition 2.5: A rate pair $\left(R_X, R_Y\right)$ is said to be achievable if there exists a sequence of $\left(2^{NR_X}, 2^{NR_Y}\right)$ codes with $P_e^{(N)} \to 0$ as $N \to \infty$, where the probability of error is defined as $P_e^{(N)} = P\left[f\left(g_X\left(\underline{x}\right), g_Y\left(\underline{y}\right)\right) \neq \left(\underline{x}, \underline{y}\right)\right]$.

[1] In this case, there are some residual decoding errors but the decoding error probability tends towards 0 for rates within the Slepian-Wolf (SW) achievability region (Theorem 2.6) [52].

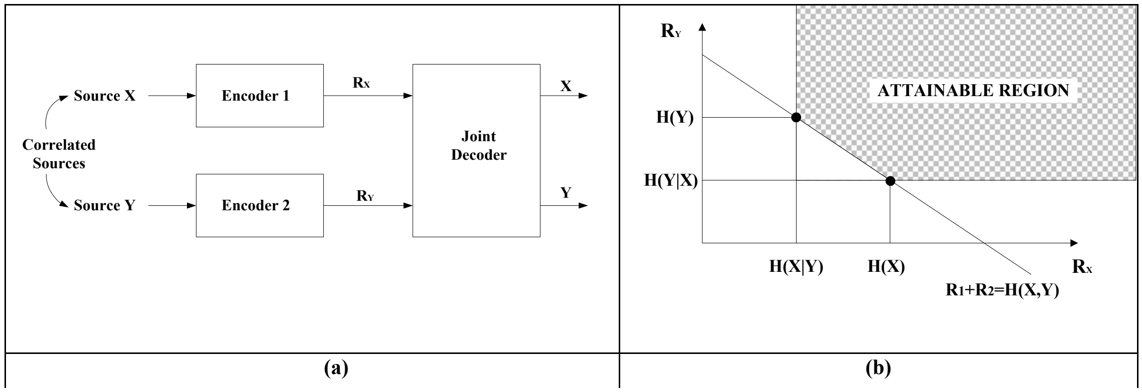


Fig. 2.1. (a) Schematic block diagram of Slepian-Wolf compression problem
(b) Attainable region in the Slepian-Wolf problem [61]

Definition 2.6: The achievable rate region in the Slepian-Wolf problem is the closure of the set of achievable rate pairs.

The Slepian-Wolf achievability region will be derived using the concept of random binning.

Theorem 2.6: The achievable rate region as defined by the SW problem [61] is illustrated in Fig. 2.1 (b) and is formally defined as the rate pairs which satisfy:

$$\begin{aligned} R_X + R_Y &\geq H(X, Y) \\ R_X &\geq H(X \mid Y) \\ R_Y &\geq H(Y \mid X) \end{aligned} \tag{2.18}$$

where $P_e^{(N)} = P\left[f\left(g_X\left(\underline{x}\right), g_Y\left(\underline{y}\right)\right) \neq \left(\underline{x}, \underline{y}\right) \right] \rightarrow 0$ as $N \rightarrow \infty$.

Proof of Theorem: Proof of this theorem is included in Appendix A.2. The converse part relies on the single-source compression theorem, i.e., if both encoders had the other source available during the encoding process, they could not compress below the respective conditional entropy. Similarly, a joint encoder could not compress below a rate *H(X,Y)*. The direct part relies the concept of joint typicality and the AEP theorem for two sources, as explained in detail in Appendix A.2.

□

From a practical point of view, one can focus on designing a Slepian-Wolf coding scheme that can operate at the corner points of the rate region in Fig. 2.1, i.e., the black dots (*H(X|Y)*, *H(Y)*) and (*H(Y|X)*, *H(X)*). Consequently, *time-sharing* these two coding schemes for α and $1-\alpha$ fraction of time will allow achieving the linear boundary of the Slepian-Wolf region. This boundary corresponds to the lowest achievable compression rate pairs in this problem. Note that coding the first mentioned rate pair, (*H(X|Y)*, *H(Y)*) , the source *Y* is entropy encoded and is used as the side information at the joint decoder to decode *X*. The source *X* has to be separately encoded at a rate close to *H(X|Y)* which is below *H(X)*, the entropy of the source X .

2.3.2 Rate Distortion Theory and the Wyner-Ziv Problem

An extension of the lossless data compression involves data representation subject to a distortion or fidelity criterion. This could include the storage or transmission of a continuous amplitude signal or a real number requiring an infinite number of bits. Under a fidelity criterion, when a certain amount of distortion is acceptable, the representation of continuous signals (or data) is possible. This kind of communication is more commonly known as *source coding with a fidelity criterion*, *rate distortion coding* or *lossy source coding* and comes under the framework of *Rate-Distortion theory*.

Rate distortion theory gives the theoretical bound for how much compression can be achieved using lossy compression methods. In other words, it addresses the problem of determining the minimal rate *R* (in bits) that is needed to communicate the data over a channel or store the data without exceeding a given distortion level. The *rate distortion theory* deals with the fundamental limits on the (vector) quantization of signals. Many existing practical compression techniques (audio, video, speech) exploit (vector) quantization procedures that use *rate distortion theory*. Motivated by this, the following definitions and theorems are presented.

Definition 2.7: Distortion measure is a non-negative function $D:\mathcal{X}\times\widehat{\mathcal{X}}\rightarrow[0,\infty)$ between the random variables $x\in\mathcal{X}$ and $\hat{x}\in\widehat{\mathcal{X}}$. The fidelity criterion (or the average distortion) is defined as $E_{X\hat{X}}\left[D\left(x,\hat{x}\right)\right]$ where E denotes the expectation. Commonly used distortion measures are the Hamming distortion:

$$D(x,\hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{if } x \neq \hat{x} \end{cases} \tag{2.19}$$

and the squared Euclidean distortion:
$$D(x,\hat{x}) = \left(x - \hat{x}\right)^2. \tag{2.20}$$

In reference to Wyner-Ziv coding, X will be the encoded source whereas $\widehat{X}$ will be the decoded or recovered version of X. $R_X(d)$ denotes the minimum rate R (in bits per source symbol) that is needed to represent source X in such a way that it can be reconstructed with an average distortion not exceeding d. As in case of lossless source coding, one can consider single source as well as multi-source rate distortion coding problems.

Wyner and Ziv considered a specific problem of rate distortion coding with side information at the decoder only [72], [73]. Fig. 2.2 shows the system model for the Wyner-Ziv problem as well as a related problem when side-information is available at both the transmitter and the receiver. When both the encoder and the decoder have access to the side information in Fig. 2.2, $R_{X|Y}(d)$ denotes the minimum rate R (in bits per source symbol) that is needed to represent source X in such a way that it can be reconstructed with an average distortion not exceeding d.

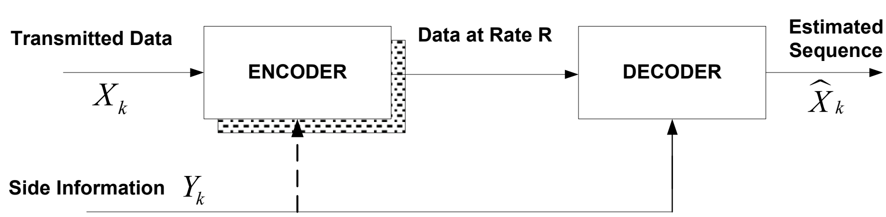


Fig. 2.2. System model for the Wyner-Ziv problem [72],[73]

Theorem 2.7: *Rate distortion function for a single source* [10], [27]

For a memoryless source X, the rate distortion functions is as follows:

$$R_X(d) = \min_{\substack{P_{\hat{X}|X,Y}(\hat{x}|x) \\ E\left[D(x,\hat{x})\right] \le d}} I(X,\hat{X}) \tag{2.21}$$

Let $\mathcal{X}, \mathcal{Y}, \widehat{\mathcal{X}}$ be sets for a pair of correlated random variables *(X,Y)* taking values in the set $\mathcal{X} \times \mathcal{Y}$. Then the *rate distortion* function $R_{X|Y}(d)$ is given by the following theorem:

Theorem 2.8: *Rate distortion function for two correlated sources* [10], [27]
For a pair of correlated memoryless sources *(X,Y)* ,

$$R_{X|Y}(d) = \min_{\substack{P_{\hat{X}|X,Y}(\hat{x}|x,y) \\ E\left[D(x,\hat{x})\right] \le d}} I(X,\hat{X} \mid Y) \tag{2.22}$$

where $I(X,\hat{X} \mid Y) > 0$ is a continuous quantity and denotes the mutual information between X and $\hat{X}$ conditioned on Y.

Theorem 2.9: *Rate distortion function for Jointly Gaussian sources X and Y*
Thus, the special case will be considered where $\mathcal{X} = \mathcal{Y} = \widehat{\mathcal{X}} = \mathbb{R}$, $D(x,\hat{x}) = (x-\hat{x})^2$ and X, Y are jointly Gaussian with zero mean and are related by the following:

$$Y = \beta(X+U) \tag{2.23}$$

where $\beta > 0$, X and U are independent Gaussian random variables, $E[X] = E[U] = E[Y] = 0$, $E[X^2] = {\sigma_X}^2$, $E[U^2] = {\sigma_U}^2$.

$$R_{X|Y}(d) = \begin{cases} \dfrac{1}{2}\log\left(\dfrac{b{\sigma_U}^2}{d}\right), & \text{for} \quad 0 < d < b{\sigma_U}^2 \\ 0, & \text{for} \quad d \ge b{\sigma_U}^2 \end{cases} \tag{2.24}$$

For brevity, the proof of $R_{X|Y}(d)$ is included in Appendix A.3.

One coding problem of interest, treated in this thesis, is related to the Wyner-Ziv problem on quantization with side-information at the receiver only (shown in Fig 2.2). In this case, $R^*(d)$ is the minimum rate in the Wyner-Ziv problem that ensures a distortion not larger than d can be achieved in reconstruction of X at the decoder[2]. Surprisingly, the authors in [73] showed that the best quantization performance achievable without the side information at the transmitter is the same as the case when the side information is available at both the encoder and decoder for jointly Gaussian sources i.e. $R^*(d) = R_{X|Y}(d)$ [73]. This is analogous to the Slepian-Wolf result for lossless coding of X given side-information Y, i.e., it says that the transmission rate cannot be lowered even if the encoder has access to the side information (It may help the complexity of the implementation.).

Theorem 2.10: $R^*(d)$ *for Jointly Gaussian X and Y* [73]

Wyner has shown in [73] that the rate distortion function $R^*(d)$ for the Wyner-Ziv problem in the case of jointly Gaussian X and Y can be expressed as follows:

$$R^*(d) = R_{X|Y}(d) = \begin{cases} \dfrac{1}{2}\log\left(\dfrac{{\sigma_X}^2{\sigma_U}^2}{\left({\sigma_X}^2+{\sigma_U}^2\right)d}\right), & \text{for} \quad 0 < d < \dfrac{{\sigma_X}^2{\sigma_U}^2}{{\sigma_X}^2+{\sigma_U}^2} \\ 0, & \text{for} \quad d \geq \dfrac{{\sigma_X}^2{\sigma_U}^2}{{\sigma_X}^2+{\sigma_U}^2} \end{cases} \tag{2.25}$$

2.4 Turbo Encoding and Decoding

In this section, the structure of turbo codes along with the process of turbo encoding and decoding is reviewed [7]. Turbo codes, proposed in [11], have been shown to operate within a fraction of one dB of the capacity limit on AWGN channels. They have become part of the 3G wireless and DVB-RCS standards [25], [63]. A typical parallel concatenated turbo system is shown in Fig. 2.3.

2.4.1 Parallel and Serial Turbo Encoding

A turbo encoder consists of parallel or serial concatenation of simpler component codes [11], [12] which are usually recursive convolutional codes. A parallel concatenated code

---

[2] For a formal statement of $R^*(d)$, please refer to Appendix A.4.

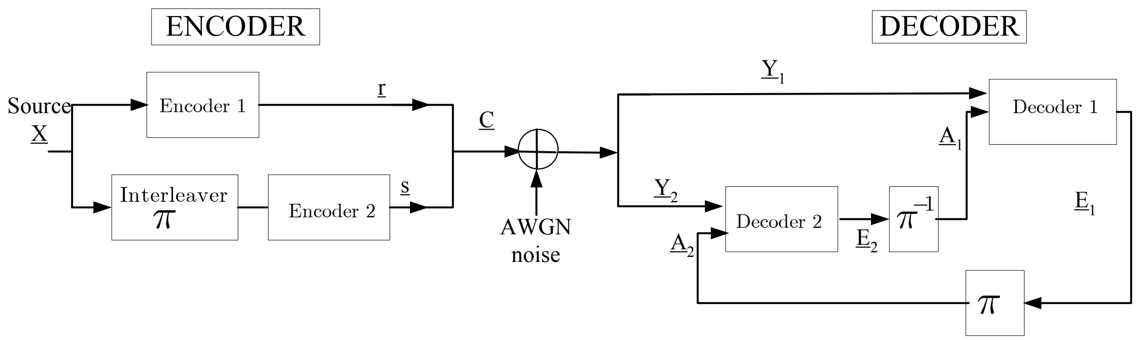


Fig. 2.3. Parallel Concatenated Turbo Encoder/Decoder system

is shown in Fig. 2.3. The message sequence $\underline{X} \in \mathbb{S}$ ($\mathbb{S}$ is the set of message sequences of length N_S bits) is encoded by the first encoder, while the interleaver π reorders the message symbols before they enter the second encoder. The interleaver can be based on a pseudorandom permutation or a pre-designed permutation, such as the *S*-random interleaver [33], which ensures that the neighboring bits in the original message sequence are mapped a certain distance apart.

The two outputs of the turbo encoder are the sequences of the coded bits $\underline{r}$ and $\underline{s}$, which are multiplexed into a single codeword $\underline{C}$ (of length N_C bits). This single codeword might also include the systematic part of the input source bits depending on the code. The recursive encoders can be represented by the feed-forward and feedback generator polynomials, $\left(G_{ff}(D), G_{fb}(D)\right)$. Fig. 2.4 illustrates the block diagram of the eight state component code used in the 3G standard [63] ($G_{ff}(D) = 1 \oplus D \oplus D^3$ and $G_{fb}(D) = 1 \oplus D^2 \oplus D^3$). The error correcting capability of a convolutional code depends on the number of errors the code can correct which relies largely on the free distance of the code[3]. Recursive convolutional codes have been shown to have a larger free distance compared to non-recursive convolutional codes. Furthermore, the overall rate of the turbo code depends on the constituent encoders being used, e.g., if the encoder from Fig. 2.4 is used, the overall rate is 1/3, i.e. one input bits gives rise to two coded bits and one systematic bit.

[3] The free distance of a code is the minimal Hamming distance between different encoded sequences. The Hamming distance between two encoded sequences is the number of positions where the two differ.

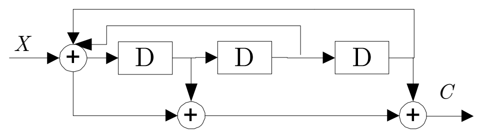


Fig. 2.4. Schematic block diagram of a recursive 8-state convolutional encoder

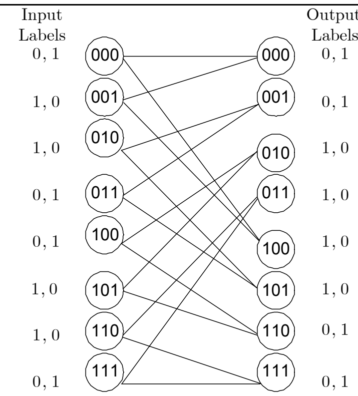


Fig. 2.5. The corresponding complete trellis diagram of the code on the left

Recursive systematic convolutional codes can also be described using a trellis representation. A trellis is a layered directed graph, in which the vertices are partitioned into sets $E_1, E_2, \ldots, E_N$, so that edges can only go from E_t to E_{t+1} in the *t*-th time instant. In a trellis, the states of the convolutional encoder are used. They are depicted as a column of vertices and are repeated at each time instant. Two of these columns form a stage also known as a *trellis stage*. A *trellis path*, on the other hand, is a series of trellis edges that an encoder traverses based on the input message sequence. The complete trellis for one stage corresponding to the encoder in Fig. 2.4 is shown in Fig. 2.5.

An important notion in trellis construction is the concept of terminating the trellis. One way to do this is by forcing the trellis to the all-zero state at the termination of the input message sequence i.e. after the last message bit arrives. This is mainly done to ensure that the encoder starts and finishes in the all-zero state and it also ensures that the last message bit is protected by several coded bits. This trellis termination is designed such that the shortest path or the smallest transitions are used to reach the all-zero state from a particular state.

Serial concatenated codes were initially used in deep space exploration and were proposed by Forney in 1966 [26]. Although these codes were serial concatenated, they did not use the iterative decoding concept which a turbo coded system uses. Also, these codes did not have an interleaver between the two serial constituent codes. A serial concatenated (turbo) code is shown in Fig. 2.6. The message source is encoded by

Encoder *1* (also known as the outer decoder) producing a coded bit sequence $\underline{C}$ which is permuted by an interleaver and consequently encoded by Encoder *2*, which is typically referred to as the inner encoder.

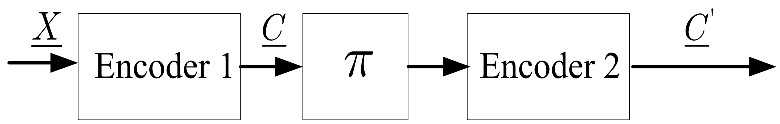


Fig. 2.6. Schematic block diagram of a serial concatenated (turbo) code.

2.4.2 Iterative Turbo Decoding

The most important part of the Turbo coded system is the iterative decoder which allows practical complexity decoding. For the turbo encoder structure shown in Fig. 2.3, optimal decoding which minimizes the bit error probability is too complex. This is due to the fact that the number of states of the joint trellis of both the encoders increases exponentially with the message block length. Thus, keeping in mind the complexity constraints, a sub-optimal structure was proposed in [11], consisting of two separate decoders connected in an iterative loop. One constituent decoder operates on the first encoder's output (and the systematic stream). The other constituent decoder operates on the second encoder's output. In this process, the two decoders exchange information on the *message bits* in case of a parallel scheme. In a serial scheme, the decoders exchange information on the first encoder's *coded bits*.

The methodology and notation used in this sub-section is based on [7], [8]. An iterative decoder for a parallel concatenated code is shown in Fig. 2.7. The two BCJR decoders are represented as δ_1 and δ_2 and are connected in an iterative loop. The output of decoder *1* is E_1 which stands for the extrinsic information, whereas the input of decoder *1* is A_1 which signifies that the input is the *a-priori* information. The same applies for decoder *2*. Thus V_j^i are the labels of the input and output of the decoders where V can be either E or A depending on the input or the output. Also, i is the time index (or the iteration count) and j is either *1* or *2* depending on the decoder. An important point worth mentioning here is that the extrinsic of one decoder is the *a-priori* of the other decoder subject to interleaving/de-interleaving. More specifically, $A_1^i = \pi\left(E_2^i\right)$ and $A_2^i = \pi^{-1}\left(E_1^i\right)$ as shown in Fig. 2.7 where $\pi(\bullet)$ is the interleaving function and $\pi^{-1}(\bullet)$ is the de-interleaving function. The turbo iterative decoding, in the *i*-th iteration can be described as

$$\begin{aligned} E_1^i &= \delta_1\left(Y_1, A_1^i\right) - A_1^i \\ E_2^i &= \delta_2\left(Y_2, A_2^{i-1}\right) - A_2^{i-1}, \end{aligned} \tag{2.26}$$

where Y_1 and Y_2 are log-likelihood ratio (LLR) vectors on the coded bits from the two encoders (Y_1 also includes LLRs on the systematic bits) and $E_1^0 = A_2^0 = [0,0,\ldots,0]$. In Fig. 2.7, the appropriate interleavering $\pi(\bullet)$ and de-interleaving $\pi^{-1}(\bullet)$ is performed. The outputs of the decoding functions are logarithmic ratios on the *a-posteriori* probabilities,

$$\begin{aligned} \delta_2\left(Y_2, A_2^{i-1}\right) &= \Pr\left(X \middle| Y_2, A_2 = A_2^{i-1}\right) \\ \delta_1\left(Y_1, A_1^i\right) &= \Pr\left(X \middle| Y_1, A_1 = A_1^i\right), \end{aligned} \tag{2.27}$$

where vectors E_1 and E_2 are known as extrinsic information. These are then fed as *a-priori* information to the next decoder. The subtraction in (2.26) makes sure that the next decoder only receives new information on the message bits. Finally, hard decisions on the message bits are performed by thresholding the output of decoder *1* by using a thresholding function $h(a) = 1$ if $a > 0$ and $h(a) = 0$ if $a < 0$, i.e.,

$$\widehat{X} = h\left(\Pr\left(X \middle| Y_1, A_1^i\right)\right). \tag{2.28}$$

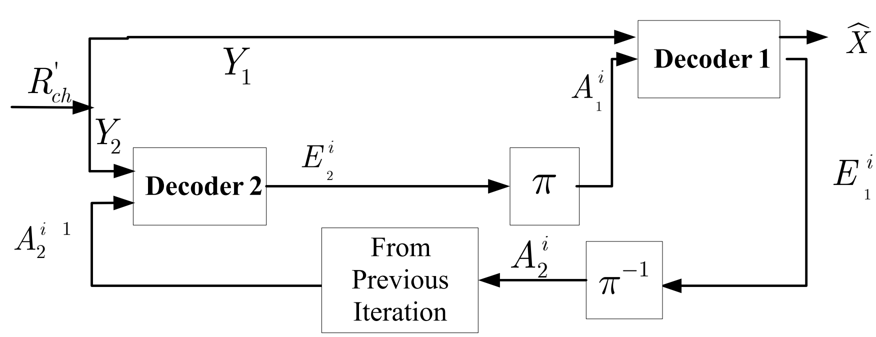


Fig. 2.7. Schematic block diagram of the parallel concatenated turbo decoder

The previous discussion focuses on parallel concatenated codes. The treatment for serial concatenated codes is a little different. Fig. 2.8 shows the schematic diagram of a decoder for a serially concatenated turbo code. The turbo decoder, in the *i*-th iteration for a serial concatenated system can be described as

$$\begin{aligned} E_1^i &= \delta_1\left(A_1^i\right) - A_1^i \\ E_2^i &= \delta_2\left(Y, A_2^{i-1}\right) - A_2^{i-1}, \end{aligned} \tag{2.29}$$

where Y is the log-likelihood ratio vector on the coded bits from the outer encoder produced by the demodulator and $E_1^0 = A_2^0 = [0,0,\ldots,0]$. The decoding functions δ_1 and δ_2 produce logarithmic ratios on the outer encoder's coded bits and message bits, respectively:

$$\begin{aligned} \delta_2\left(Y, A_2^{i-1}\right) &= \Pr\left(C \middle| Y, A_2 = A_2^{i-1}\right) \\ \delta_1\left(A_1^i\right) &= \Pr\left(X \middle| A_1 = A_1^i\right). \end{aligned} \tag{2.30}$$

The extrinsic information vectors $E_1(A_1)$ and $E_2(A_2)$ contain independent log-ratios on the coded bit sequence C from the outer encoder as shown in Fig. 2.6. The decisions on the message bits are determined like the parallel case by thresholding the output of decoder 1,

$$\widehat{X} = h\left(\Pr\left(X \middle| A_1^i\right)\right). \tag{2.31}$$

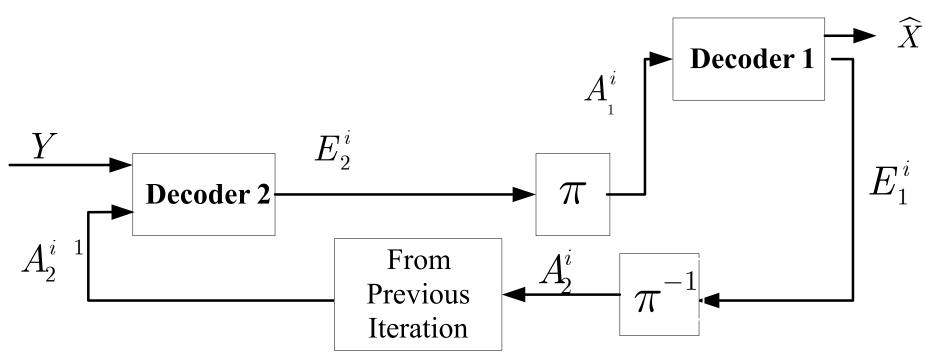


Fig. 2.8. Schematic block diagram for a serially concatenated turbo decoder

Chapter 3 EXIT Chart Analysis

This chapter explores the Extrinsic Information Transfer (EXIT) charts, one of the two recently proposed and successfully used techniques for analysis and design of turbo coded communication systems.[4] The main objectives of this chapter are as follows:

- Explain the EXIT charts and their construction methodology in detail
- Properly extend the EXIT charts method to compressive turbo codes
- Derive several new analytical results regarding the EXIT charts
- Extend the EXIT chart performance analysis to analog iterative decoders

3.1 Motivation of EXIT Charts Performance Analysis

Performance analysis is an integral part of communication system design, allowing designers to choose optimal values of system parameters and/or understand sensitivity and robustness to various imperfections and disturbances. Monte Carlo simulations are often used to study bit error rate (BER) performance of a system by simulating its model repeatedly on a computer. Nonetheless, for turbo coded systems, reliable simulations of small BER values can be extremely time consuming and do not provide insight into the performance impact of various system parameters. On the other hand, performance analysis techniques can often provide faster evaluation of BER curves and/or permit parametric sensitivity analysis.

In 2001 Ten Brink proposed a semi-analytical technique to describe the flow of extrinsic information between two constituent decoders [15]. This technique is commonly known as the Extrinsic Information Transfer (EXIT) chart. The EXIT charts offer a simple and robust graphical description of the turbo decoding process and are used to predict the SNR value corresponding to when a channel turbo decoder starts to converge,

[4] The uniform interleaver analysis [9], [23] is the other such technique and relies on the input-output weight enumerating functions of constituent encoders and analytically computed union bounds.

i.e., the start of the (turbo) waterfall region. The EXIT charts enable quick evaluation of the effects that the design parameters (constituent encoders, used decoding algorithms, etc.) have on the overall system performance. One of the main advantages of using EXIT charts is that the two constituent decoders can be analyzed separately and the two results are then combined to obtain the overall performance picture.

The basic ingredients in the construction of the EXIT charts are two extrinsic information curves formed for the two constituent decoders. The extrinsic information curves are formed by using the extrinsic information arrays that are computed and exchanged by the two constituent decoders, as explained in Section 2.4.2 and as shown in Fig. 3.1 for a parallel and serial concatenated scheme.

The EXIT chart technique involves the calculation of the mutual information of the extrinsic and the message bits in addition to the mutual information of the *a-priori* knowledge with respect to the message bits. These mutual information’s are then plotted as the Extrinsic Information Transfer (EXIT) chart. The EXIT Chart is a graphical description of the turbo decoding process and is used to predict the SNR value corresponding to the start of the waterfall region in particular among other things. In addition to the prediction of the SNR values, links have been made between EXIT Charts and the corresponding BER curves. The main advantage of the EXIT Charts is its robustness and simplicity along with the fact that it gives the designer a visual tool to analyze the performance of a particular.

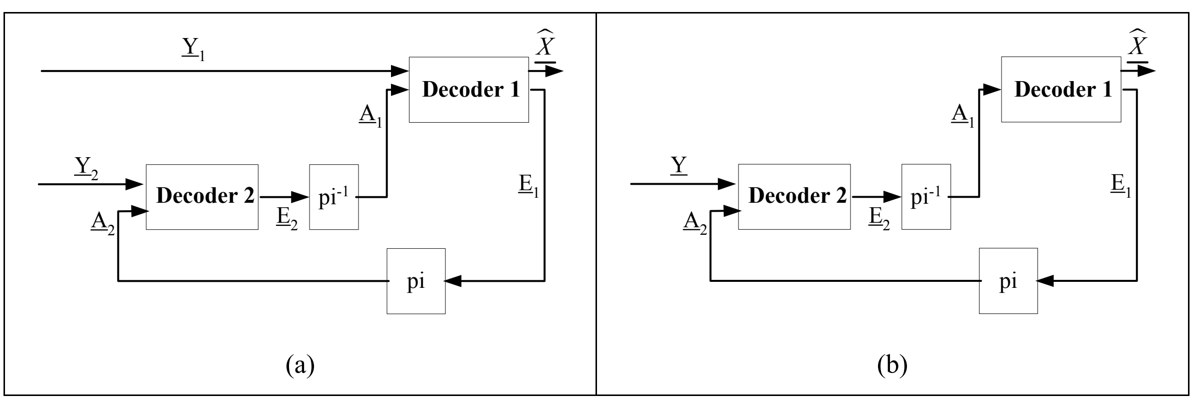


Fig. 3.1. Overall system model for an iterative decoder in a (a) parallel and (b) serial concatenated system

Section 3.2.1 will present the step by step procedure for the construction of an EXIT chart for turbo channel codes. Furthermore, Section 3.3 will cover the extension of the EXIT chart technique to turbo compressive codes and will provide several analytical results to properly characterize the mutual information quantities used in the construction of the EXIT charts. Section 3.4 will further extend the EXIT chart technique to develop an insight into the performance analysis of analog iterative decoders.

3.2 EXIT Chart Methodology for Turbo Channel Codes

3.2.1 A Step-by-Step Construction of an EXIT Chart

The construction of the EXIT chart can be divided into three steps as explained in this sub-section. Before going into the details of these steps, it is important to note that a memoryless source X is used which produces symbols *-1* and *1* with probabilities p and *(1-p),* respectively. The information in the decoding process is expressed in terms of probabilities and to make the decoding process practical, log likelihood ratios (LLR) are used e.g. for a source X with output symbols *+1* and *-1*, the LLR is defined as:

$$L(X) = \ln \frac{P(x=+1)}{P(x=-1)} \tag{3.1}$$

Thus, from this point on, the input and output of the constituent BCJR decoders will be denoted in the form of LLRs. Prior to presenting the step by step construction of the EXIT charts, it is necessary to present the LLR notation used and define an AWGN channel.

Definition 3.1: LLR Notation

The *LLR notation* has to be clarified as it is imperative to make a distinction between the actual LLRs exchange in an iterative decoder and the LLRs in the system model of a constituent decoder used to construct the EXIT chart. In an actual iterative decoder, L_{Ai} is used for *a-priori* LLR at the input of the constituent decoder i and L_{Ei} is used for *a-posteriori* LLR at the output of the constituent decoder i, where i can be either *1* or *2* depending on which decoder is being used. Also,

$$L_{Ai} = L_{Ai}(X) = \ln \frac{P(x=+1 \mid A_i)}{P(x=-1 \mid A_i)} \tag{3.2}$$

$$L_{Ei} = L_{Ei}(X) = \ln \frac{P(x=+1 \mid E_i)}{P(x=-1 \mid E_i)} \tag{3.3}$$

On the other hand, in an approximate model of a constituent decoder *i* used for the construction of the EXIT chart, $\tilde{L}_{Ai}$ and $\tilde{L}_{Ei}$ is used for the *a-priori* and *a-posteriori* LLRs of the constituent decoder *i* respectively, where *i* can be either *1* or *2*. Similarly,

$$\tilde{L}_{Ai} = \tilde{L}_{Ai}(X) = \ln \frac{P(x=+1 \mid \tilde{A}_i)}{P(x=-1 \mid \tilde{A}_i)} \tag{3.4}$$

$$\tilde{L}_{Ei} = \tilde{L}_{Ei}(X) = \ln \frac{P(x=+1 \mid \tilde{E}_i)}{P(x=-1 \mid \tilde{E}_i)} \tag{3.5}$$

Definition 3.2: Additive White Gaussian Noise (AWGN) is a model that distorts or changes the amplitude/phase of a transmitted signal and is distributed according to a Gaussian distribution which has a probability density function as shown below:

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{3.6}$$

where σ = standard deviation or and μ = mean with *x* being the random variable.

The most common noise encountered in digital communications is the Additive White Gaussian Noise (AWGN). The transmitted binary signal $y_k \in \{\pm 1\}$ is corrupted by the AWGN noise i.e.

$$z_k = y_k + n_k \tag{3.7}$$

The word white in AWGN comes from that fact that the spectral density of the noise is flat or constant in the required frequency range. In (3.7), the *k* signifies the discrete time index. The channel output PDF conditioned on the transmitted binary symbol *y* is

$$p(z \mid y) = \frac{1}{\sigma_n\sqrt{2\pi}} \exp(-\frac{(z-y)^2}{2\sigma_n^2}) \tag{3.8}$$

Step 1: Empirical Normal Distribution of Extrinsic Information Vectors

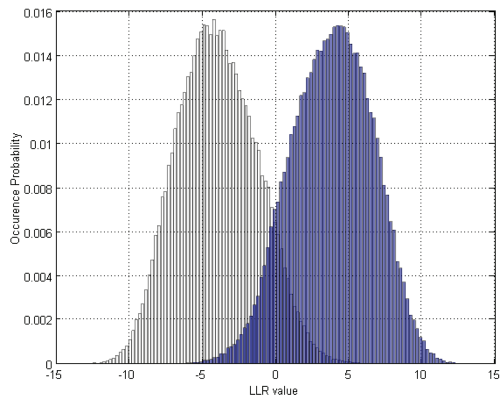


Fig. 3.2 Extrinsic information distribution of the LLR's for two constituent decoders in a turbo coded system with a rate of 1-to-3 [63] . The histograms of the LLR's are plotted after the 3rd iteration for a source with equi-probable symbols.

Fig. 3.3 (a) shows how an actual constituent BCJR decoder is used in an iterative turbo decoding system. Channel observations on the component codeword and the *a-priori* information contained in the extrinsic information array L_{A1} are used by a *single* BCJR decoder to generate the new extrinsic information array L_{E1} for the message bits. This array is then passed to the next BCJR decoder. The first step in the construction of an EXIT chart is to properly model the *a-priori* information that is contained in the extrinsic information array L_{A1}, so that the two constituent decoders can be analyzed separately. The *a-priori* input L_{A1} to the channel decoder turns out to be near-Gaussian distributed, as observed by different authors ([11], [15], [32]) and explicitely shown in Fig. 3.2.

Step 2: Replacement Model of an Actual Constituent Decoder

For the purposes of EXIT chart construction, Fig. 3.3 (b) shows the *replacement model* of the constituent BCJR decoder 1. The actual extrinsic information array L_{A1} from the original setup in Fig. 3.3 (a) will be replaced in Fig. 3.3 (b) by input message bits corrupted by Gaussian noise with a prescribed noise variance, so that

$$\widetilde{A}_1 = x + n \ ,$$

where the noise ha zero mean and a prescribed varianc $\sigma_n^{\ 2}$. In literature, this is referred to as passing the message through a a-*priori* channel with a prescribed noise variance[5].

The *a-priori* information, $\widetilde{L}_{A1}$, available at the input of the constituent BCJR decoder in Fig. 3.3 (b) has approximately the same *distribution* as the actual *a-priori* information L_{A1} available at the input of a actual constituent BCJR decoder 1 in Fig. 3.3 (a). We write this fact as

$$L_{A1} \overset{i.d.}{\approx} \widetilde{L}_{A1} \tag{3.9}$$

and similarly observe the same for the outputs of the BCJR decoders in Fig. 3.3 (a) and Fig. 3.3 (b), i.e., $L_{E1} \overset{i.d.}{\approx} \widetilde{L}_{E1}$. Note that the same procedure is applied for the other constituent BCJR decoder *2* with the subscripts indices *1* and *2* replaced with respect to each other.

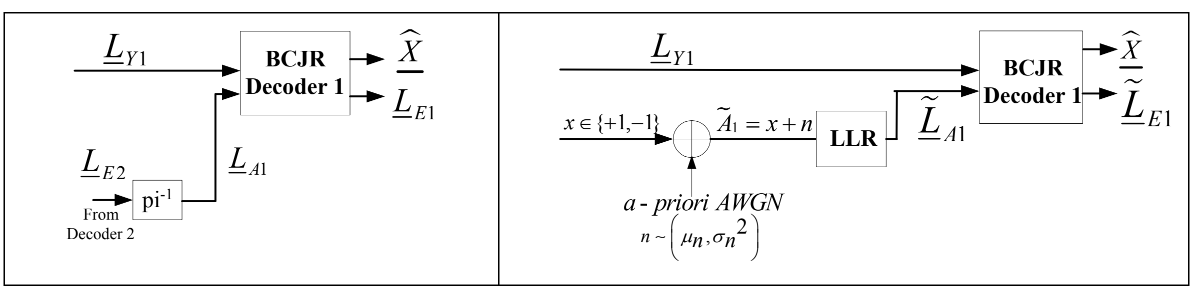


Fig. 3.3 (a) Constituent decoder 1 in an actual iterative turbo decoding system; (b) Constituent decoder 1 model used to approximate the *a-priori* information for the construction of the EXIT chart.

---

[5] The *a-priori* channel could also be a *Binary Erasure Channel* or *Binary Symmetric Channel* e.g. the source can also be modeled by a binary erasure channel (with erasure probability ε) when it is assumed that the other decoder automatically converts errors to erasures.

Step 3: Construction of an EXIT Curve for a Constituent Decoder

To quantify the amount of decoding information using a single variable, we can evaluate $I(X;L_{A1})$ and $I(X;L_{E1})$, the mutual information between the data source X and the two extrinsic information arrays at the input and the output of the constituent decoder. If $I(X;L_{E1})=0$, there is no information about X in the extrinsic information array, while if $I(X;L_{E1})=H(X)$, the decoder has the information about source message fully available. Furthermore, for different values of $I(X;L_{A1})$, we can also plot $I(X;L_{E1})$ as a function of $I(X;L_{A1})$ in order to graphically observe the improvement achieved by the constituent decoder. This gives an indication about how much information is added by the constituent BCJR decoder and made available to the next decoder.

For an actual iterative decoder, achieving the objectives stated in the above paragraph turns out to be analytically untractable and excessively computationally difficult. Consequently, we will use the replacement model from Fig. 3.3 (b) for a constituent decoder and evaluate $I(X;\widetilde{L}_{E1})$ and $I(X;\widetilde{L}_{A1})$ instead of $I(X;L_{E1})$ and $I(X;L_{A1})$. In other words, once the replacement extrinsic information arrays $\widetilde{L}_{A1}$ and $\widetilde{L}_{E1}$ have been numerically generated, the next step involves the evaluation of the mutual information between these quantities and the symbols from source X, i.e., $I(X;\widetilde{L}_{E1})$ and $I(X;\widetilde{L}_{A1})$.

As proposed in [15], these mutual information quantities can be evaluated for a given source X, constituent BCJR decoder, replacement model from Fig. 3.3 (b) and σ_n^2 by averaging over many message sequences. The following theorem from [32] presents the formula used for numerical evaluation of the mutual information in the replacement model.

Theorem 3.1: [32] Consider a binary discrete source X with equiprobable symbols, $P(X=+1)=p=0.5$, which generates a message of length N. The mutual information between X and $\tilde{L}_A$ or X and $\tilde{L}_E$ can be expressed as

$$I(X;\tilde{L}) = 1 - \frac{1}{N}\sum_{n=1}^{N}\left[1+\log_2\left(1+e^{-x_n\tilde{L}_n}\right)\right] \tag{3.10}$$

where $\tilde{L}$ denotes either $\tilde{L}_A$ (the LLR at the input of the decoder) or $\tilde{L}_E$ (the LLR at the output of the decoder), x_n is the *n*-th sample of the source and $\tilde{L}_n$ is *n*-th sample of $\tilde{L}$.

Proof of Theorem: Please refer to the proof of Theorem 3.2 in the next Section (3.3.3), as this theorem generalizes of the above result for a general binary memoryless source with 0<*p*<*1*.

To illustrate the concepts discussed in this step, Fig. 3.4 shows two generic EXIT curves obtained for the same constituent decoder using two approaches discussed above. Fig. 3.4 (a) contains the black dots corresponding to the mutual information pairs $\left[I\left(X;L_{E1}\right),I\left(X;L_{A1}\right)\right]$ that are obtained for an actual constituent decoder in an iterative decoder setup from Fig. 3.1. On the other hand, Fig. 3.4 (b), shows this EXIT curve obtained using the replacement system model explained in Step 1 and Fig. 3.3 (b). The latter curve has been obtained by changing the variance σ_n^2, so that for each value of σ_n^2, there exists a corresponding mutual information pair $\left[I\left(X;\tilde{L}_{E1}\right),I\left(X;\tilde{L}_{A1}\right)\right]$. Thus, by varying σ_n^2, numerous pairs of $\left[I\left(X;\tilde{L}_{E1}\right),I\left(X;\tilde{L}_{A1}\right)\right]$ are obtained and plotted to get the EXIT curve in Fig. 3.4 (b). Note that the same procedure is applied for the other constituent BCJR decoder *2* with the subscripts indices *1* and *2* replaced with respect to each other.

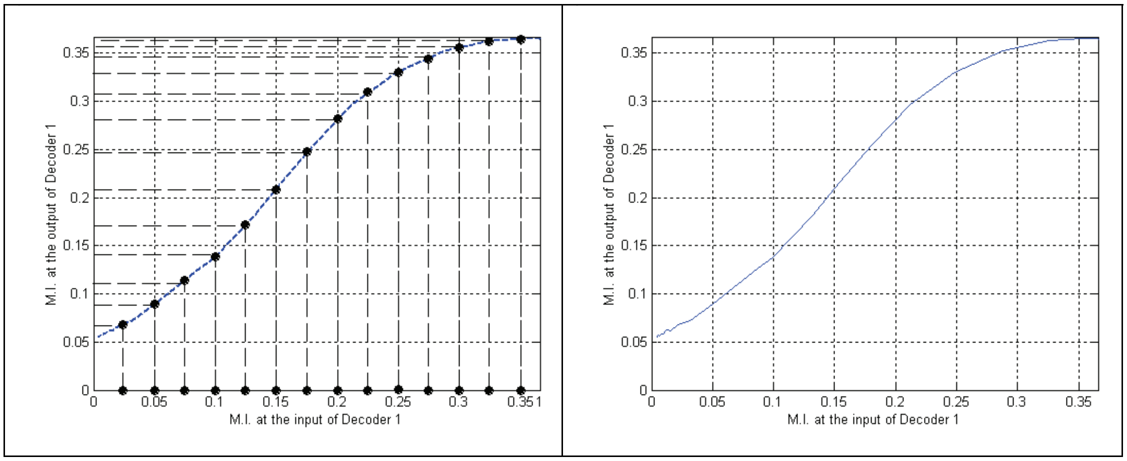


Fig. 3.4 (a) Example of a generic input/output mutual information curve for a constituent BCJR decoder 1 in an iterative decoder: $I\left(X;L_{E1}\right)$ vs. $I\left(X;L_{A1}\right)$; (b) Example of an EXIT curve for a constituent BCJR decoder 1 used in the construction of an EXIT chart: $I\left(X;\tilde{L}_{E1}\right)$ vs. $I\left(X;\tilde{L}_{A1}\right)$

Step 4: Combining Two EXIT Curves into an EXIT Chart

In an iterative turbo decoder, the above procedure can be applied to obtain an EXIT curve for each constituent decoder at a given SNR. Note that one of the curves can be flipped around the *y=x* axis, since the *a-priori* information of one decoder is the output extrinsic information of the other decoder (see Fig. 3.1). Thus, these two EXIT curves can be plotted on the same figure to form an EXIT chart, as described below and shown in Fig. 3.5. If a sufficiently large interleaver separates the two constituent decoders, so that statistical independence of neighbouring extrinsic information symbols is assured, .this resulting EXIT chart can be used to trace mutual information evolution throughout the iterative decoding process is measured with under the following conditions ([15], [32]). Note that in the case of parallel concatenated codes, the two constituent encoders are usually identical, hence both EXIT curver characteristics are often symmetrical about the $y = x$ line.

If an open tunnel forms between the two EXIT curves in an EXIT chart, the iterative decoding trajectory passed through this tunnel. A widely open tunnel indicates the turbo decoder will converge within a small number of iterations If the EXIT curves of the two decoders intersect, then the tunnel is closed and the turbo decoder fails to

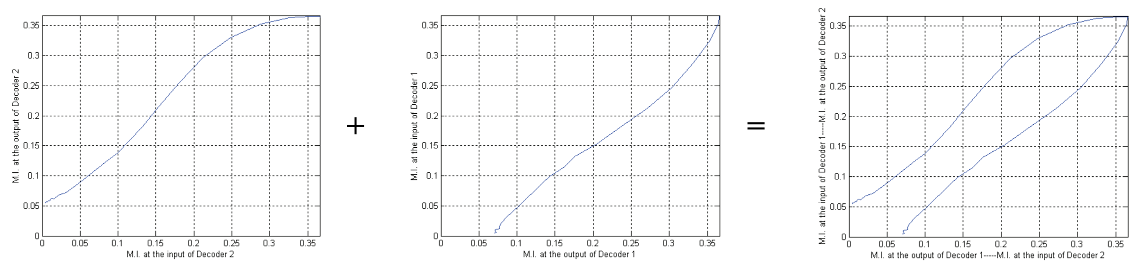


Fig. 3.5 Example of an EXIT chart (with an open tunnel) consisiting of two EXIT curves

converge i.e. the message sequence is not decoded successfully. Thus, the SNR value corresponding to the start of the waterfall region is where the two EXIT curves barely touch each other. In this case, it takes a large number of iterations for the decoder to pass through this threshold area of the EXIT chart. The EXIT chart methodology can be easily adapted to a serial turbo scheme, as explained in Section 3.2.2.

An example of the combination of two generic EXIT curves to form an EXIT chart is shown in Fig. 3.5. Note that this procedure allows separate investigation and analysis the two constituent decoders (two codes), an major advantage in the sense that the two decoders do not have to be decoded iteratively to plot the EXIT Chart and understand the behaviour during the iterative decoding process.

3.2.2 Extension of the EXIT Chart Methodology to Serial Turbo Codes

The steps for EXIT chart construction mentioned in Section 3.2.1 are for parallel concatentated schemes. The same procedure can be applied to the EXIT chart construction for serial concatenated turbo codes, as constituent decoders are again individually studied using EXIT curves, but there are some differences stated as follows:

- In a parallel iterative decoder, the constituent decoders exchange information on the source message X. On the other hand, in serial iterative decoder, the constituent decoders exchange information on the coded symbols C as shown in Fig. 2.6. Thus, instead of passing X through an AWGN channel, C is passed throught an AWGN channel. Furthermore, successful decoding of C implies successful decoding of X.

- In case of a parallel concatenated scheme both the decoders having access to the channel information Y_1 and Y_2 as shown in Fig. 3.1 (a) and for a serial scheme the decoder *2* (inner decoder) has access to the channel information as shown in Fig. 3.1 (b). The channel information of the outer decoder i.e. Decoder *1*, as shown in Fig. 3.1, is non-existent since C does not pass through a channel.
- For the serial case, $I\left(C;\tilde{L}_{E1}\right)$ is plotted as a function of $I\left(C;\tilde{L}_{A1}\right)$ to obtain the EXIT curve for the outer constituent decoder *1*. Similarly, for the inner constituent decoder *2*, $I\left(C;\tilde{L}_{E2}\right)$ is plotted as a function of $I\left(C;\tilde{L}_{A2}\right)$ to obtain another EXIT curve. The two EXIT curves for the constituent decoders are then combined using a similar procedure as described in Step 4 in Section 3.2.1. Note that the same mutual information formula given in (3.10) can be used to obtain the mutual information quantities in the serial case as well, with X being replaced with C.

3.2.3 Summary of the EXIT Chart Methodology

The EXIT chart analyzes the constituent decoder pairs in a turbo coded system separately, thus there is a need to somehow model the constituent single encoder/decoder pair as explained in Step 1 and 2 in Section 3.2.1. The replacement model for a single constituent decoder was given in Step 2. As mentioned before, in the case of a parallel concatenated system, the consitutuent decoders exchange information on the source symbols x whereas for a serial concatenated system, the constituent decoders exchange information on the coded symbosl c. Fig. 3.6 to Fig. 3.7 show the replacement model for the parallel and serial case respectively. Consequently, using these replacement models and applying Steps 3 and 4 mentioned in Section 3.2.1, the corresponding EXIT chart can be obtained.

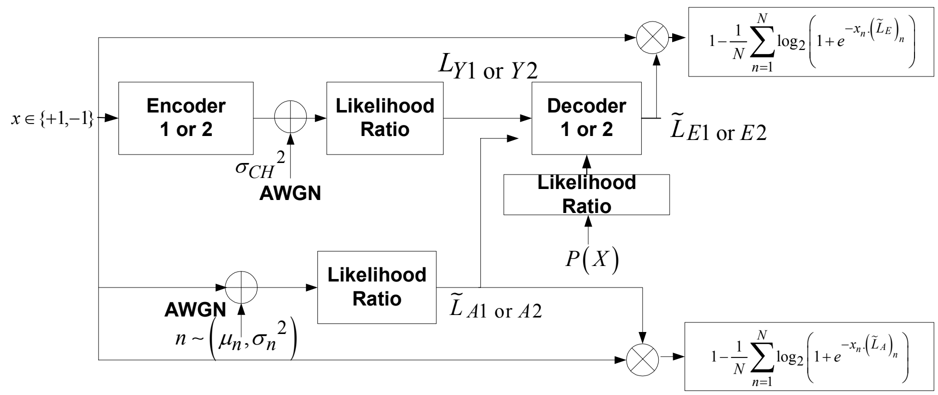


Fig. 3.6 Block diagram, based on the replacement model, that can be used to generate an EXIT curve for a constituent encoder/decoder in a *parallel* concatenated system

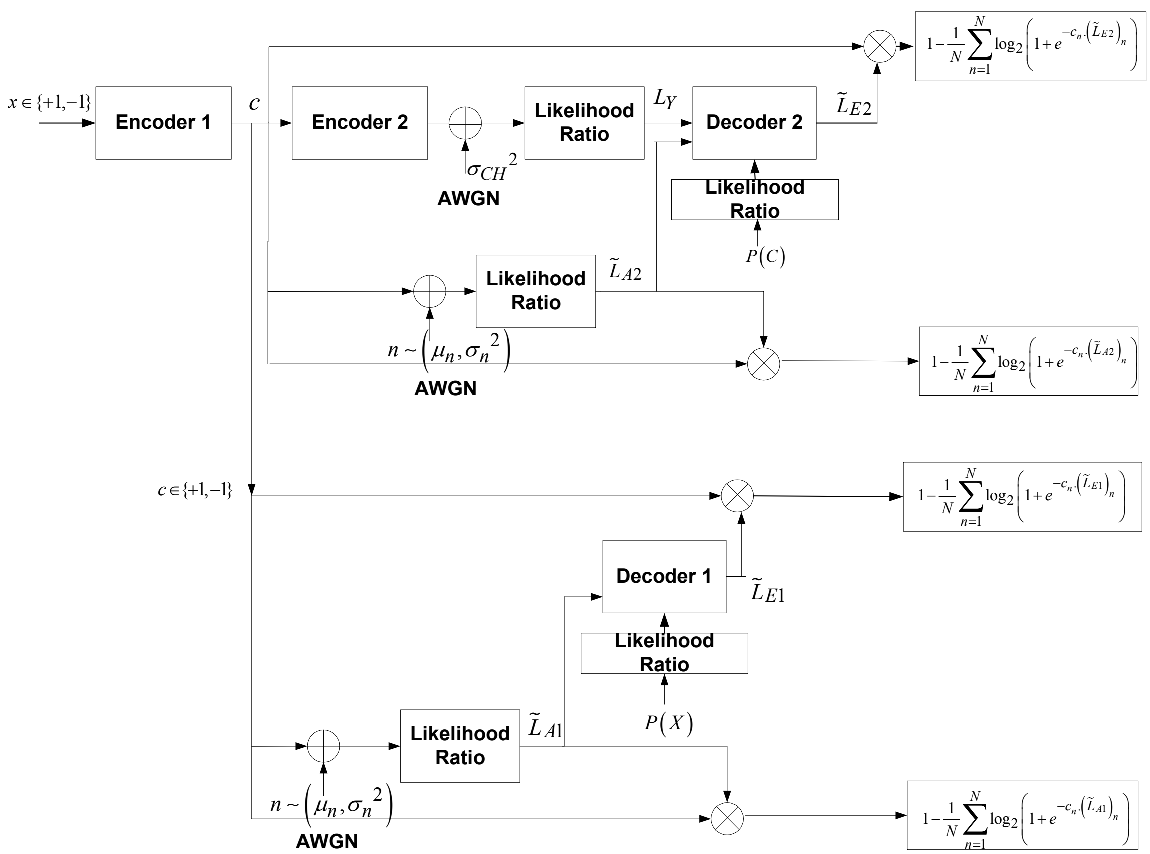


Fig. 3.7 Block diagram, based on the replacement model, that can be used to generate an EXIT curve for a *outer* and *inner* constituent encoder/decoder in a *serial* concatenated system

The EXIT Chart technique shows the corresponding iterative decoding trajectory. Initially, the trajectory starts at a mutual information of zero and subsequently rises to a new value based on the extrinsic information produced by the first constituent decoder, which is equal to the mutual information on the *a-priori* information given to the second decoder. The second constituent decoder decodes its channel observations with the help of the *a-priori* information provided by the first decoder. The resulting mutual information computed on the extrinsic information increases and ideally should be more than the mutual information on the *a-priori* knowledge provided at the input. This process is repeated for each iteration until the data is decoded successfully or the EXIT tunnel is closed in which case the iterative trajectory gets stuck.

3.3 Extension of EXIT Charts to Compressive Turbo Codes

3.3.1 Motivation

This section presents an extension of the EXIT chart technique to the performance analysis of recently proposed compressive turbo codes. The mutual information formula presented in Theorem 3.1 is for turbo channel codes and is used to construct EXIT curves that form an EXIT chart. Hagenauer in [32] has attempted to extend this result to turbo compressive codes and presented the following mutual information formula for binary memoryless sources with non equiprobable symbols:

$$I(X;\tilde{L}) = H_b(p) - \frac{1}{N}\sum_{n=1}^{N}\left[1 + \log_2\left(1 + e^{-x_n \tilde{L}_n}\right)\right] \tag{3.11}$$

where $P(X=+1) = p$, $\tilde{L} = \begin{cases} \tilde{L}_A & \text{if the LLR is at the input of the decoder} \\ \tilde{L}_E & \text{if the LLR is at the output of the decoder} \end{cases}$, $H_b(p)$ is the binary entropy function, x_n is the n-th sample of the source X and $\tilde{L}_n$ is n-th sample of $\tilde{L}$.

Nonetheless, the above formula turns out to be inaccurate, as shown later in this section. Consequently, a generalized version of Theorem 3.1 will be derived for biased sources, resulting into an improved formula

$$I(X;\tilde{L}) \cong H_b(p) - \frac{1}{N}\sum_{n=1}^{N}\left[p\log_2\left(1 + \frac{(1-p)\cdot e^{-x_n \tilde{L}_n}}{p}\right) + (1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n \tilde{L}_n}}{(1-p)} + 1\right)\right] \tag{3.12}$$

where again $P(X=+1)=p$, $\tilde{L}=\begin{cases}\tilde{L}_A & \text{if the LLR is at the input of the decoder}\\ \tilde{L}_E & \text{if the LLR is at the output of the decoder}\end{cases}$, *$H_b(p)$* denotes the binary entropy function, x_n is the *n*-th sample of the source *X* and $\tilde{L}_n$ is the *n*-th sample of $\tilde{L}$.

Furthermore, numerical results will be presented to illustrate good EXIT chart results obtained using the above generalized formula in Section 3.3.5. A comparison with the results based on the previously proposed formula (3.12) from [32] will be given. (Note that the formulas given in (3.11) and (3.12) are for parallel concatenated codes and can be easily adapted to serial concatenated codes by replacing *X* with *C* as mentioned in Section 3.2.2).

3.3.2 Previous Work on Single Source Turbo Compression

Recently, turbo codes have been designed to perform data compression at rates close to the source entropy using two approaches [1], [46], [74]. The authors in ([1], [74]) proposed Turbo compression based on heavily punctured linear turbo channel codes while the authors in [46] constructed non-linear Latin-square-based compressive encoders[6]. For both approaches, near-entropy simulated performance was reported. The simulations show that the fixed length to fixed length compression schemes achieves near lossless source coding at rates close to the source entropy. Data compression using turbo codes has gained wide acceptance over the past couple of years.

Constituent encoders used by the authors in [46] output *k* binary symbols for every *n* binary input symbols, where $2k<n$. The ratio of the total number of output symbols to the number of input symbols is chosen to be less than the source entropy *H(X)*. An example of the FSM Latin square based encoders is shown Fig. 3.8. There are three input bits at each time instant and one output bit is generated by one encoder, thus $k = 1$ and $n = 3$ and the overall compressive code rate for the compressive turbo encoder is

[6] An *n* by *n* Latin square is an *n* by *n* matrix such that any of the rows or columns of that matrix do not contain any number more than once. They consist of *n* sets of the numbers from *1* till *n*.

3:2. The FSM encoder is based on two transition matrices: an input transition matrix and an output transition matrix where the $(i,j)^{th}$ entry corresponds to the k-symbol input sequence when the encoder makes a transition from states i to j. The trellis structures of the encoders for 3-bit input ($k=3$, $n=1$) are shown below:

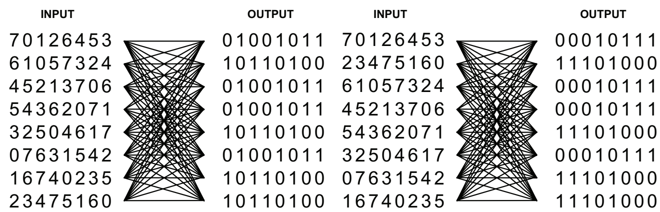


Fig. 3.8. Trellises of two rate 3-to-1 compressive FSM encoders from [46]. (Input symbols are shown in octal notation, while output symbols are binary.)

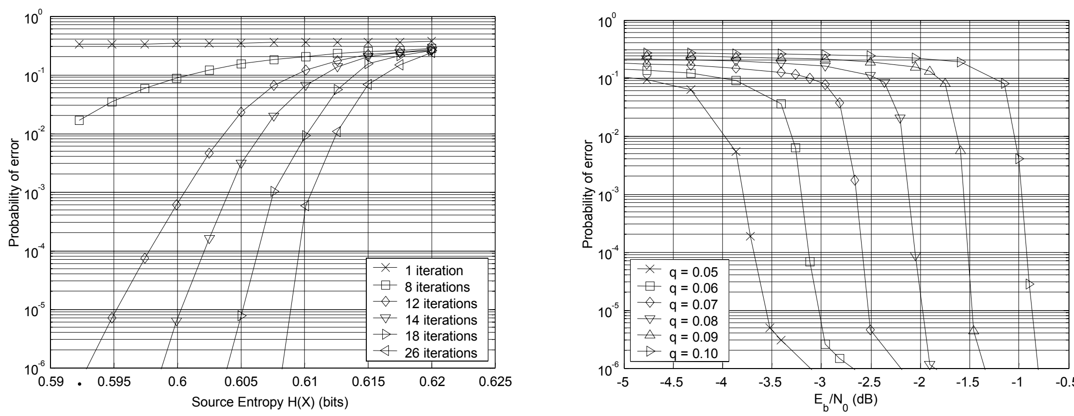


Fig. 3.9. Performance with perfect knowledge of compressed data (left) and noisy measurement of compressed data (right) – compressive turbo encoder of rate 3-to-2 [46]

The result of this particular turbo compression scheme is shown in Fig. 3.9. As shown by the figure on the left, after *26* decoding iterations, the error rate reaches 10^{-5} for source with entropy up to *0.608* bits. Thus, the rate of the encoder, which is *0.667* bits/source symbol, is only *10* percent above the entropy rate of the source. Also, the noise robustness of the proposed scheme is shown by the graph on the right in Fig. 3.9. As can be seen, for a lower probability of error, the source entropy has to be decreased for a fixed E_b/N_0. Furthermore, for fixed source entropy, E_b/N_0 has to be increased to achieve a lower probability of error.

For a source bias $q = 0.05$, $H(X) = 0.2863$ bits/symbol and for $q = 0.10$, $H(X) = 0.469$ bits/symbol. As the entropy of the source is decreased by varying q from *0.10* to

0.05, E_b/N_o can be lowered by *2.7* dB for a fixed error rate of 10^{-5}. This results from the fact that a source with a bias of $q=0.05$ and *0.10* needs an E_b/N_o of *-3.6* dB and *0.9* dB respectively to achieve an error rate of 10^{-5}.

Similarly, serial compression can also be performed using Turbo codes by serially concatenating two codes such that the combined rate is compressive. The authors in [21] used a *1*-to-*2* code serially concatenated with a *3*-to-*1* code to get an overall rate of *3*-to-*2*. The *3*-to-*1* code is the one used in the parallel turbo compression scheme with an overall rate of *3*-to-*2* as shown in Fig. 3.8. Furthermore, for a serial compressive code of rate *4*-to-*2*, a *1*-to-*2* code can be concatenated with a *4*-to-*1* code used in the parallel compressions case of rate *4*-to-*2*.

The authors in [31] investigate the issue of using turbo codes for source coding by applying ten Brink's EXIT chart analysis and examine how this technique can be used to select the most efficient match of component codes and puncturing matrices to compress discrete memoryless sources. An encoding algorithm is presented which aims to perform lossless source coding, i.e. perfect reconstruction at the decoder. This is achieved by enabling the encoder to puncture the parity bits on a step by step basis (*decremental redundancy*) as long as the compressed block can still be decoded error free by the test decoder at the source compression side. Furthermore, the authors in [33] propose a lossless compression technique for binary memoryless sources using short block length turbo codes as a large block length lossless turbo source encoder offers compression rates close to the source entropy but with large latency. The focus is on the design of the parity interleaver for different compression rates and this is accomplished by replacing the square shape puncturing array with a rectangular shape array that allows finer puncturing and hence improved compression rates. A new puncturing scheme that outperforms both structured and pseudo-random puncturing methods introduced in the literature, for sources with entropy close to 0.5.

The compression of non-binary sources using turbo codes is studied in [75] by transforming the non-binary input symbols into sequences of bits, and then source coded using punctured turbo codes, with the puncturing adjusted to achieve the desired compression rate. The authors in [75] use this turbo compression scheme to compress a pair of correlated non-binary sources with the correlation now known neither at the encoder nor at the decoder. The correlation model between sources is estimated jointly with the iterative decoding process. Compared with the case in which the correlation is known at the decoder, no significant performance loss is observed. The performance of the proposed scheme is close to the Slepian–Wolf theoretical limit.

Moreover, conventional parallel (turbo) and serial concatenated convolutional codes can be used to compress binary sources where conventional refers to codes already used in channel coding [43]. The compression is done in [43] by considering the coded bits of a channel coding trellis as the uncompressed input information with the compressed output being the syndrome bit at each trellis stage. The simulated results show that for the binary case, the practically lossless compression achieved by conventional turbo codes is higher than that of the currently available nonconventional turbo schemes and close to the theoretical limit. The focus in [43] is on compression of an equi-probable memoryless binary source with side information at the decoder where the approach is based on modeling the correlation as a channel and using syndromes. The performance achieved is seen to be better than the recently published results using nonconventional turbo codes and is closer to the Slepian-Wolf limit.

The authors in [64] compare the performance of three different methods for the transmission of non-uniform sources over AWGN and Rayleigh channels. One of the methods is the classical one considering separation between source and channel coding which includes using Huffman codes as the source encoder followed by a systematic turbo encoder. The two other methods are based on source-controlled channel decoding, where data is not compressed prior to transmission and redundancy is exploited at receiver. The second method uses non-systematic turbo encoding [77] and uses the source statistics at the receiver. On the other hand, the third method is based on a special class of nonsystematic turbo codes and makes use of unequal energy allocation based on the source non-uniformity. The three methods make use of turbo codes as the channel code. Simulation results in [64] show that, in some cases and in terms of bit error rate, it may be more advantageous not to compress data prior to transmission.

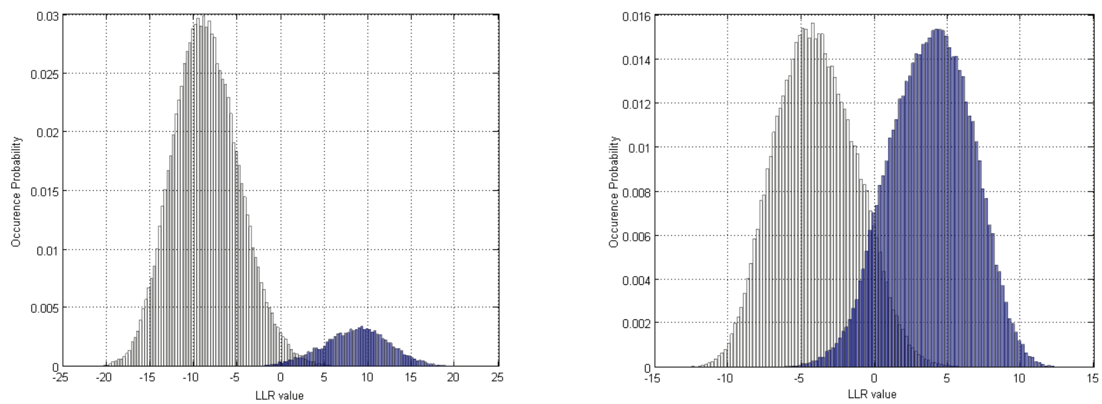


Fig. 3.10 Extrinsic information distribution of the LLR's for two constituent decoders in a parallel turbo coded system. The histograms of the LLR's are plotted after the 3rd iteration
(a) (Left) Compression scheme with a rate of *3*-to-*2* for a source bias of *P(X=-1)=0.9*.
(b) (Right) Channel coding scheme with a rate of *1*-to-*3* [63] for a source with equi-probable symbols.

An efficient structured binning scheme is presented in [65] for the compression of binary sources with parallel concatenated convolutional codes, or turbo codes. The novelty in the proposed scheme is the introduction of a syndrome former and an inverse syndrome former to efficiently and optimally exploit an existing turbo code without the need to redesign or modify the code structure and/or decoding algorithms. The authors focus on the distributed source coding problem and the simulation results reveal good performance which is close to theoretic limit.

As shown in Fig. 3.10 (a) based on an actual simulation of a compressive turbo code [46], the extrinsic LLR's exchanged between the constituent decoders are still Gaussian for compressive turbo codes. For comparison, Fig. 3.10 (b) shows the extrinsic information distribution when source symbols are equiprobable. The only difference between Fig. 3.10 (a) and Fig. 3.10 (b) is that the conditional normal distributions of the LLR's are weighted by the unequal source symbol probabilities.

3.3.3 Derivation of the Main Result

It is important to note that a memoryless source X produces symbols $+1$ and -1 with probabilities p and $(1-p)$, respectively. The information in the decoding process is expressed in terms of probabilities and to make the decoding process practical, these are kept as log likelihood ratios (LLR). For example, for a source *X* with output symbols *+1* and *-1*, the LLR is defined as:

$$L(X) = \ln \frac{P(x=+1)}{P(x=-1)}. \tag{3.13}$$

For the communication channel as well as the *a-priori* modeling channel discussed above, the probability density function of the output of the channel is represented by a continuous distribution denoted by $f(y|x)$, where y is the output from the channel and x is the channel input symbol. In this case, posterior probabilities (given channel observation y) are used in the decoding process, e.g., in the log-likelihood format

$$L_Y(X) = \ln \frac{P(x=+1 \mid y)}{P(x=-1 \mid y)} \tag{3.14}$$

Theorem 3.2: This theorem is divided into three parts for clarity:

a) The mutual information between a discrete binary random variable X such that $P_X(x) = \begin{cases} p & \text{if} \quad x=+1 \\ 1-p & \text{if} \quad x=-1 \end{cases}$ and $\tilde{L}_A$, the input log likelihood ratio of the source symbols in the replacement model (shown in Fig. 3.3 (b)), can be expressed as

$$I(X;\tilde{L}_A) = H_b(p) - E\left\{ p\log_2\left(1+\frac{(1-p)\cdot e^{-x\tilde{L}_A}}{p}\right) + (1-p)\cdot\log_2\left(\frac{p\cdot e^{-x\tilde{L}_A}}{(1-p)}+1\right)\right\} \tag{3.15}$$

b) The mutual information between a discrete binary random variable X such that and $\tilde{L}_A$, as defined in part (a) can be simplified as:

$$I(X;\tilde{L}_A) \cong H_b(p) - \frac{1}{N}\sum_{n=1}^{N}\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n(\tilde{L}_A)_n}}{p}\right) + (1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n(\tilde{L}_A)_n}}{(1-p)}+1\right)\right] \tag{3.16}$$

c) In the case of $\tilde{L}_E$, the LLR of the source symbols at the output of the constituent decoders, the mutual information between a discrete binary random variable X such that $P_X(x) = \begin{cases} p & \text{if} \quad x=+1 \\ 1-p & \text{if} \quad x=-1 \end{cases}$ and $\tilde{L}_E$, the output log likelihood ratio of the source symbols in the replacement model (shown in Fig. 3.3 (b)), can be expressed as:

$$\begin{aligned} I(X;\widetilde{L}_E) &\cong H_b(p) - E\left\{ p\log_2\left(1+\frac{(1-p)\cdot e^{-x\widetilde{L}_E}}{p}\right) + (1-p)\cdot\log_2\left(\frac{p\cdot e^{-x\widetilde{L}_E}}{(1-p)}+1\right)\right\} \\ &\cong H_b(p) - \frac{1}{N}\sum_{n=1}^{N}\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n(\widetilde{L}_E)_n}}{p}\right) + (1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n(\widetilde{L}_E)_n}}{(1-p)}+1\right)\right] \end{aligned} \quad (3.17)$$

Proof of Theorem:

(a): Initially, recall the replacement model for an actual decoder and its input a-priori information in LLR format:

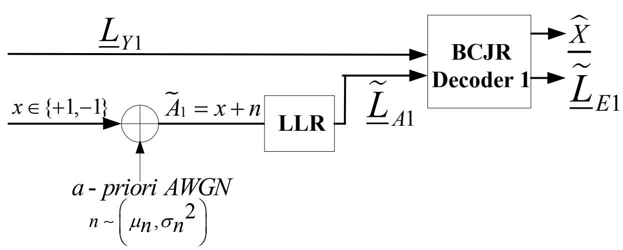


Fig. 3.11 Constituent decoder 1 model used to approximate the *a-priori* information for the construction of the EXIT chart

Fig. 3.12 shows how the discrete source X is modelled as a Gaussian random variable. The next step is to find the distribution of the *a-priori* LLR $\widetilde{L}_A$. Due to the diagram, the *a-priori* channel is subject to the addition of the AWGN noise of mean *0* and variance σ_n^2.

$$f(\widetilde{A}\,|\,x) = \frac{1}{\sqrt{2\pi\sigma_n^2}}\exp\left[\frac{-(\widetilde{A}-x)^2}{2\sigma_n^2}\right] \quad (3.18)$$

where σ_A is the standard deviation of the noise for the “*a-priori*” modeling channel.

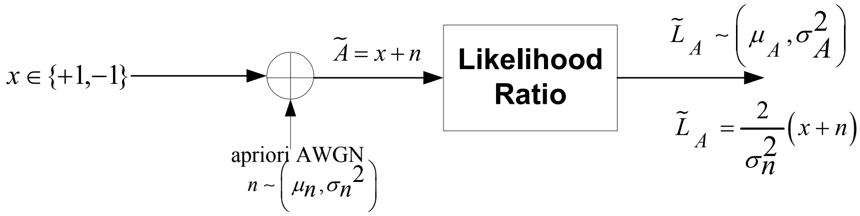


Fig. 3.12. Equivalent replacement model of the extrinsic LLR’s, originally defined in Section 3.2.1 in Fig. 3.3 (b)

From the replacement model, using the definition of the LLR of $\tilde{L}_A$, we obtain

$$\begin{aligned}\tilde{L}_A &= \ln\left(\frac{f(\tilde{A}\,|\,x=+1)}{f(\tilde{A}\,|\,x=-1)}\right)\\ &= \ln\left\{\frac{\exp\left[-(\tilde{A}-(+1))^2 \big/ 2\sigma_n^2\right]}{\exp\left[-(\tilde{A}-(-1))^2 \big/ 2\sigma_n^2\right]}\right\}\\ &= 2\tilde{A}\big/\sigma_n^2\end{aligned} \tag{3.19}$$

where we have used (3.18) and $e^A/e^B = e^{A-B}$ to simplify the expressions. Consequently.

$$\tilde{A} = x + n \qquad \text{and} \qquad \tilde{L}_A = \frac{2\tilde{A}}{\sigma_n^2}, \tag{3.20}$$

so $\tilde{L}_A$ given X=x is known to be Gaussian distributed with mean $E\left(\widetilde{L_A}\right) = \dfrac{2x}{\sigma_n^2} = \dfrac{\sigma_A^2 x}{2}$

and a variance $\sigma_A^2 = \mathrm{var}\left(\widetilde{L_A}\right) = \dfrac{4}{\sigma_n^2}$ and has the following conditional distribution:

$$f\left(\tilde{L}_A \mid X = x\right) = \frac{1}{\sqrt{2\pi\sigma_A^2}} \exp\left[-\left(\tilde{L}_A - \frac{\sigma_A^2}{2}x\right)^2 \Big/ 2\sigma_A^2\right] \quad \text{for} \quad x \in \{-1, +1\} \tag{3.21}$$

Hence, using (3.21), we obtain

$$\begin{aligned}\frac{f(\tilde{L}_A \mid X=-1)}{f(\tilde{L}_A \mid X=+1)} &= \frac{\cancel{\frac{1}{\sqrt{2\pi\sigma_A^2}}} \exp\left[-\left(\tilde{L}_A - \frac{\sigma_A^2}{2}(-1)\right)^2 \Big/ 2\sigma_A^2\right]}{\cancel{\frac{1}{\sqrt{2\pi\sigma_A^2}}} \exp\left[-\left(\tilde{L}_A - \frac{\sigma_A^2}{2}(+1)\right)^2 \Big/ 2\sigma_A^2\right]}\\ &= \exp\left\{\left[-\left(\tilde{L}_A - \frac{\sigma_A^2}{2}(-1)\right)^2 + \left(\tilde{L}_A - \frac{\sigma_A^2}{2}(+1)\right)^2\right] \Big/ 2\sigma_A^2\right\}\\ &= \exp\left\{-2\tilde{L}_A\sigma_A^2 / 2\sigma_A^2\right\}\\ &= \exp\left\{-\tilde{L}_A\right\}\end{aligned} \tag{3.22}$$

Thus, getting back to the characterization of the mutual information between a discrete source X which outputs binary symbols $x \in \{-1,+1\}$, and a continuous random variable Y, the following formula is used:

$$\begin{aligned}
&I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \\
&I(X;Y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \left(f(y|x)\cdot f(x)\right)\log_2 \frac{f(y|x)}{f(y)}\,dxdy \\
&\left| Note: f(y) = P(X=+1)\cdot f(y|X=+1) + P(X=-1)\cdot f(y|X=-1) \right| \\
&= \sum_{x=\pm 1}\int_{-\infty}^{+\infty} f(y|X=x)\cdot P(X=x).\log_2 \frac{f(y|X=x)}{P(X=+1)\cdot f(y|X=+1)+P(X=-1)\cdot f(y|X=-1)}\,dy
\end{aligned} \tag{3.23}$$

Consequently, using the source statistics $P_X(x) = \begin{cases} p & \text{if} \quad x=+1 \\ 1-p & \text{if} \quad x=-1 \end{cases}$ in (3.23) becomes:

$$\begin{aligned}
I(X;Y) &= \sum_{x=\pm 1}\int_{-\infty}^{+\infty} f(y|X=x)P(X=x)\log_2 \frac{f(y|X=x)}{pf(y|X=+1)+(1-p)f(y|X=-1)}\,dy \\
&= \sum_{x=\pm 1} P(X=x)\int_{-\infty}^{+\infty} f(y|X=x)\cdot\log_2 \frac{f(y|X=x)}{p\cdot f(y|X=+1)+(1-p)\cdot f(y|X=-1)}\,dy \\
&= p\int_{-\infty}^{+\infty} f(y|X=+1)\cdot\log_2 \frac{f(y|X=+1)}{p\cdot f(y|X=+1)+(1-p)\cdot f(y|X=-1)}\,dy + \\
&\quad (1-p)\cdot\int_{-\infty}^{+\infty} f(y|X=-1)\cdot\log_2 \frac{f(y|X=-1)}{p\cdot f(y|X=+1)+(1-p)\cdot f(y|X=-1)}\,dy
\end{aligned} \tag{3.24}$$

For the construction of the EXIT charts, the mutual information has to be calculated between the source X and the extrinsic LLR's $\tilde{L}$ where in (3.24), $\tilde{L} = y$ where $\tilde{L}$ can be either $\tilde{L}_A$ or $\tilde{L}_E$. $\tilde{L}_A$ is the *a-prori* LLR (at the input) and $\tilde{L}_E$ is the *a-posteriori* LLR (at the output). At first, the focus will be on $\tilde{L}_A$ which will be extended to $\tilde{L}_E$ later on. Thus, (3.24) can be written as:

$$\begin{aligned}
I(X;\tilde{L}_A) &= p\int_{-\infty}^{+\infty} f(\tilde{L}_A|X=+1)\cdot\log_2 \frac{f(\tilde{L}_A|X=+1)}{p\cdot f(\tilde{L}_A|X=+1)+(1-p)\cdot f(\tilde{L}_A|X=-1)}\,d\tilde{L}_A + \\
&\quad (1-p)\cdot\int_{-\infty}^{+\infty} f(\tilde{L}_A|X=-1)\cdot\log_2 \frac{f(\tilde{L}_A|X=-1)}{p\cdot f(\tilde{L}_A|X=+1)+(1-p)\cdot f(\tilde{L}_A|X=-1)}\,d\tilde{L}_A
\end{aligned} \tag{3.25}$$

Simplifying the terms inside the $\log_2(\bullet)$:

$$
\begin{aligned}
I(X;\tilde{L}_A) = p\int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\log_2 \frac{1}{p+(1-p)\cdot\dfrac{f(\tilde{L}_A \mid X=-1)}{f(\tilde{L}_A \mid X=+1)}} d\tilde{L}_A + \\
(1-p)\cdot\int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=-1)\cdot\log_2 \frac{1}{p\cdot\dfrac{f(\tilde{L}_A \mid X=+1)}{f(\tilde{L}_A \mid X=-1)}+(1-p)} d\tilde{L}_A
\end{aligned}
\tag{3.26}
$$

Substituting $\tilde{L}_A$ for $-\tilde{L}_A$ in the second part of (3.26), the result is:

$$
\begin{aligned}
I(X;\tilde{L}_A) = p\int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\log_2 \frac{1}{p+(1-p)\cdot e^{-\tilde{L}_A}} d\tilde{L}_A + \\
(1-p)\cdot\int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\log_2 \frac{1}{p\cdot e^{-\tilde{L}_A}+(1-p)} d\tilde{L}_A
\end{aligned}
\tag{3.27}
$$

Eq. (3.27) can be simplified to:

$$
\begin{aligned}
I(X;\tilde{L}_A) &= \int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[-p\log_2\left(p+(1-p)\cdot e^{-\tilde{L}_A}\right)\right] d\tilde{L}_A + \\
&\quad \int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[-(1-p)\cdot\log_2\left(p\cdot e^{-\tilde{L}_A}+(1-p)\right)\right] d\tilde{L}_A \\
&= \int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[-p\log_2 p\cdot\left(1+\frac{(1-p)}{p}\cdot e^{-\tilde{L}_A}\right)\right] d\tilde{L}_A + \\
&\quad \int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[-(1-p)\cdot\log_2 (1-p)\cdot\left(\frac{p}{(1-p)}\cdot e^{-\tilde{L}_A}+1\right)\right] d\tilde{L}_A \\
&= \int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[-p\log_2 p - p\log_2\left(1+\frac{(1-p)}{p}\cdot e^{-\tilde{L}_A}\right)\right] d\tilde{L}_A + \\
&\quad \int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[-(1-p)\cdot\log_2 (1-p) - (1-p)\cdot\log_2\left(\frac{p}{(1-p)}\cdot e^{-\tilde{L}_A}+1\right)\right] d\tilde{L}_A
\end{aligned}
\tag{3.28}
$$

The integrals in (3.28) can be rearranged to give:

$$
\begin{aligned}
I(X;\tilde{L}_A) &= \int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[-p\log_2 p-(1-p)\cdot\log_2(1-p)\right]d\tilde{L}_A + \\
&\int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[-p\log_2\left(1+\frac{(1-p)}{p}\cdot e^{-\tilde{L}_A}\right)-(1-p)\cdot\log_2\left(\frac{p}{(1-p)}\cdot e^{-\tilde{L}_A}+1\right)\right]d\tilde{L}_A \\
&= \left[-p\log_2 p-(1-p)\cdot\log_2(1-p)\right]\int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot d\tilde{L}_A - \\
&\int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[p\log_2\left(1+\frac{(1-p)}{p}\cdot e^{-\tilde{L}_A}\right)+(1-p)\cdot\log_2\left(\frac{p}{(1-p)}\cdot e^{-\tilde{L}_A}+1\right)\right]d\tilde{L}_A
\end{aligned}
\tag{3.29}
$$

Thus, noting that $H_b(p) = -p\log_2 p-(1-p)\cdot\log_2(1-p)$, (3.29) can be simplified as:

$$
\begin{aligned}
I(X;\tilde{L}_A) &= H_b(p) - \\
&\int_{-\infty}^{+\infty} f(\tilde{L}_A \mid X=+1)\cdot\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-\tilde{L}_A}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-\tilde{L}_A}}{(1-p)}+1\right)\right]d\tilde{L}_A
\end{aligned}
\tag{3.30}
$$

(b): Before proceeding with the proof of this part of the Theorem, the following definition has to be introduced:

Definition 3.3: *(Source Ergodicity)*

Qualitatively, a process is ergodic if its statistical characterization can be inferred from an observation of its realization. Mathematically, let A be a probability space and $Y \in A$. Let Y_k denote the k^{th} sample of Y. Then, Y is said to be ergodic if

$$
\frac{1}{n}\sum_{k=1}^{n} Y_k \underset{n\to\infty}{\to} E\{Y\}
\tag{3.31}
$$

with probability *1* where $E(\bullet)$ is the expectation. Note that this is a consequence of the law of large numbers.

To construct the EXIT Charts, the mutual information has to be calculated between the source information X and the LLR's at the input of the constituent decoders which are denoted as $\tilde{L}_A$. Thus, the mutual information between X and the Log Likelihood ratios $\tilde{L}_A$ simplifies to (using Definition 3.3**):**

$$\begin{aligned} I(X;\tilde{L}_A) &= H_b(p) - E\left\{ p\log_2\left(1+\frac{(1-p)\cdot e^{-x\tilde{L}_A}}{p}\right) + (1-p)\cdot\log_2\left(\frac{p\cdot e^{-x\tilde{L}_A}}{(1-p)}+1\right)\right\} \\ &\cong H_b(p) - \frac{1}{N}\sum_{n=1}^{N}\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n(\tilde{L}_A)_n}}{p}\right) + (1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n(\tilde{L}_A)_n}}{(1-p)}+1\right)\right] \end{aligned} \quad (3.32)$$

The expectation is replaced by a time average by invoking the ergodicity of the source, where X is the input message source, N is the total number of samples taken from the source X, where x_n is the n^{th} sample of the source X for $n=1, 2, \ldots, N$, $H_b(p)= -p*\log_2(p) - (1-p)*\log_2(1-p)$ is the binary entropy function, $\tilde{L}_A$ is the *a-priori* probability LLR of the source symbol X with $\left(\tilde{L}_A\right)_n$ being the n-th sample of $\tilde{L}_A$.

(c): Consequently, to find the mutual information between X and $\tilde{L}_E$ i.e. $I\left(X;\tilde{L}_E\right)$, (3.32) can be used and $\tilde{L}_A$ can be replaced with $\tilde{L}_E$. This is possible as $\tilde{L}_E$ also turns out to be approximately conditionally normal distributed as shown in Fig. 3.2 and further explained in Section 3.2.1. Furthermore, due to the fact that $L_{E2} = L_{A1}$ and $L_{E1} = L_{A2}$ in an actual iterative decoder, $\tilde{L}_A$ can be replaced with $\tilde{L}_E$. Moreover, eq. (3.32) can be used to approximately find $I\left(X;\tilde{L}_E\right)$ as the expectation is replaced by the time average using the source ergodicity theorem, signifying that the mutual information can be measured from a large number of samples without knowing the *exact* distribution of the extrinsic LLR's $\tilde{L}_E$ ([15], [32]). Consequently, the result is:

$$I(X;\tilde{L}_E) \cong H_b(p) - E\left\{ p\log_2\left(1+\frac{(1-p)\cdot e^{-x\tilde{L}_E}}{p}\right) + (1-p)\cdot\log_2\left(\frac{p\cdot e^{-x\tilde{L}_E}}{(1-p)}+1\right)\right\} \quad (3.33)$$

where $H_b(p)$ is the binary entropy function, x_n is the n-th sample of the source X and $\left(\tilde{L}_E\right)_n$ is the n-th sample of $\tilde{L}_E$.

It is important to note that the author in [32] gives the inaccurate expression for $I\left(X;\widetilde{L}_E\right)$ as an equality rather than as an approximation. Additionally, the exact expression for $I\left(X;\widetilde{L}_E\right)$ can only be found by knowing the exact statistical distribution of $\widetilde{L}_E$. This is rather tedious and involves intensive mathematical analysis. It will later be observed that the constructed EXIT chart, using the approximate $I\left(X;\widetilde{L}_E\right)$, quite accurately describes the behaviour of turbo iterative decoding systems. Thus, for the sake of brevity, the approximate expression for $I\left(X;\widetilde{L}_E\right)$ is used in the construction of the EXIT charts.

Furthermore, similar to part (b) of this Theorem, the expectation can replaced by a time average by invoking the ergodicity of the source (Definition 3.3),

$$I(X;\widetilde{L}_E) \cong H_b(p) - \frac{1}{N}\sum_{n=1}^{N}\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n\left(\widetilde{L}_E\right)_n}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n\left(\widetilde{L}_E\right)_n}}{(1-p)}+1\right)\right] \quad (3.34)$$

□

Corollary 3.2: For a binary discrete source X with equiprobable symbols with $P(X=+1)=p=0.5$, the mutual information is:

$$I(X;\widetilde{L}) \cong H_b(0.5) - \frac{1}{N}\sum_{n=1}^{N}\left[1+\log_2\left(1+e^{-x_n\widetilde{L}_n}\right)\right] = 1-\frac{1}{N}\sum_{n=1}^{N}\left[1+\log_2\left(1+e^{-x_n\widetilde{L}_n}\right)\right] \quad (3.35)$$

Where $\widetilde{L} = \begin{cases} \widetilde{L}_A & \text{if the LLR is at the input of the decoder} \\ \widetilde{L}_E & \text{if the LLR is at the output of the decoder} \end{cases}$, $H_b(p)$ denotes the binary entropy function, x_n is the n-th sample of the source X and $\widetilde{L}_n$ is the n-th sample of $\widetilde{L}$.

Proof of Corollary:

$$I(X;\widetilde{L}) \cong H_b(p) - \frac{1}{N}\sum_{n=1}^{N}\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n\widetilde{L}_n}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n\widetilde{L}_n}}{(1-p)}+1\right)\right]$$

Substituting $p = 0.5$ above and noting that $H_b(0.5) = 1$

$$=1-\frac{1}{N}\sum_{n=1}^{N}\left[0.5\log_2\left(1+\frac{(0.5)\cdot e^{-x_n\tilde{L}_n}}{0.5}\right)+(0.5)\cdot\log_2\left(\frac{0.5\cdot e^{-x_n\tilde{L}_n}}{(0.5)}+1\right)\right] \tag{3.36}$$

$$=1-\frac{1}{N}\sum_{n=1}^{N}\left[\log_2\left(1+e^{-x_n\tilde{L}_n}\right)\right]$$

It can also be deduced that for *p=0.5*:

$$p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n\tilde{L}_n}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n\tilde{L}_n}}{(1-p)}+1\right)=\log_2\left(1+e^{-x_n\tilde{L}_n}\right) \tag{3.37}$$

□

Theorem 3.3: The mutual information $I(X;\tilde{L})$ where $\tilde{L} = \tilde{L}_E$ or $\tilde{L}_A$ is lower bounded by the right hand side of eq. (3.38).

$$I(X;\tilde{L}) \geq H_b(p) - \frac{1}{N}\sum_{n=1}^{N}\left[1+\log_2\left(1+e^{-x_n\tilde{L}_n}\right)\right] \tag{3.38}$$

where X is a binary discrete memoryless source with $P(X = +1) = p$, $\tilde{L} = \begin{cases} \tilde{L}_A & \text{if the LLR is at the input of the decoder} \\ \tilde{L}_E & \text{if the LLR is at the output of the decoder} \end{cases}$, *$H_b(p)$* denotes the binary entropy function, x_n is the *n*-th sample of the source X and $\tilde{L}_n$ is the *n*-th sample of $\tilde{L}$.

(Note: This result was inaccurately given as an equality in ([24], [32]) for binary discrete sources which did not have equiprobable symbols.) It is interesting to note that the lower bound above is achieved with equality for the case mentioned in Corollary 3.2 i.e. for a binary discrete source X with equiprobable symbols.

To be more precise, the expression in eq. 3.38 is an *exact* lower bound for the case of $\tilde{L}_A$ and an *approximate* lower bound for the case of $\tilde{L}_E$. This is a direct consequence of Theorem 3.2 which states that the distribution of $\tilde{L}_A$ is exactly Gaussian, whereas the distribution for $\tilde{L}_E$ is known to be approximately Gaussian. For the sake of brevity and to prove this theorem, $\tilde{L}$ will be used where $\tilde{L} = \tilde{L}_E$ or $\tilde{L}_A$.

Proof of Theorem[7]:

Using the convexity of a function result (Appendix B.1), it is known that

$$f\left(\alpha \boldsymbol{a}+(1-\alpha)\boldsymbol{b}\right)\leq \alpha f\left(\boldsymbol{a}\right)+(1-\alpha)f\left(\boldsymbol{b}\right) \tag{3.39}$$

As $\log_2(\bullet)$ is a concave function, (3.39) can be written as (without any loss of generality)

$$f\left(\alpha \boldsymbol{a}+(1-\alpha)\boldsymbol{b}\right)\geq \alpha f\left(\boldsymbol{a}\right)+(1-\alpha)f\left(\boldsymbol{b}\right) \tag{3.40}$$

Using the function inside the sum in (3.32), and comparing it to the right hand side of (3.40), it can be shown that

$$\begin{gathered} f(\bullet)=\log_2(\bullet) \\ \alpha = p \\ \boldsymbol{a}=1+\frac{(1-p)\cdot e^{-x\tilde{L}}}{p} \quad \text{and} \quad \boldsymbol{b}=\frac{p\cdot e^{-x\tilde{L}}}{(1-p)}+1 \end{gathered} \tag{3.41}$$

Substituting the above expression in the left hand expression in (3.40), the result is

$$\begin{aligned} \log_2\left[p\cdot(\boldsymbol{a})+(1-p)\cdot\boldsymbol{b}\right] &= \log_2\left[p\cdot\left(1+\frac{(1-p)\cdot e^{-x\tilde{L}}}{p}\right)+(1-p)\cdot\left(\frac{p\cdot e^{-x\tilde{L}}}{(1-p)}+1\right)\right] \\ &= \log_2\left[\left(p+(1-p)\cdot e^{-x\tilde{L}}\right)+\left(p\cdot e^{-x\tilde{L}}+(1-p)\right)\right] \\ &= \log_2\left[1+e^{-x\tilde{L}}\right] \end{aligned} \tag{3.42}$$

This implies that:

$$\log_2\left[1+e^{-x\tilde{L}}\right]\geq p\log_2\left(1+\frac{(1-p)\cdot e^{-x\tilde{L}}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-x\tilde{L}}}{(1-p)}+1\right) \tag{3.43}$$

Multiplying both sides of (3.43) by -1, the result is:

---

[7] For the proof of the above theorem, $\tilde{L}$ will be used throughout where $\tilde{L}=\tilde{L}_E$ or $\tilde{L}_A$. To be more precise, the expression in eq. 3.38 is an exact lower bound on $\tilde{L}_A$ and an approximate lower bound on $\tilde{L}_E$.

$$
\begin{aligned}
-\log_2\left[1+e^{-x\tilde{L}}\right] &\le -\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x\tilde{L}}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-x\tilde{L}}}{(1-p)}+1\right)\right] \\
-\sum_{n=1}^{N}\log_2\left[1+e^{-x_n\tilde{L}_n}\right] &\le -\sum_{n=1}^{N}\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n\tilde{L}_n}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n\tilde{L}_n}}{(1-p)}+1\right)\right]
\end{aligned}
\tag{3.44}
$$

Finally, adding $H_b(p)$ to both the sides in (3.44),

$$
\begin{aligned}
H_b(p)-\sum_{n=1}^{N}\log_2\left[1+e^{-x_n\tilde{L}_n}\right] \le H_b(p)- \\
\sum_{n=1}^{N}\log_2\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n\tilde{L}_n}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n\tilde{L}_n}}{(1-p)}+1\right)\right]
\end{aligned}
\tag{3.45}
$$

Thus,

$$
\begin{aligned}
I(X;\tilde{L}) &\cong H_b(p)-\frac{1}{N}\sum_{n=1}^{N}\left[p\log_2\left(1+\frac{(1-p)\cdot e^{-x_n\tilde{L}_n}}{p}\right)+(1-p)\cdot\log_2\left(\frac{p\cdot e^{-x_n\tilde{L}_n}}{(1-p)}+1\right)\right] \\
&\ge H_b(p)-\frac{1}{N}\sum_{n=1}^{N}\left[\log_2\left[1+e^{-x_n\tilde{L}_n}\right]\right]
\end{aligned}
\tag{3.46}
$$

with equality if and only if $P(X=+1)=p=0.5$ i.e. the binary source X outputs equi-probable symbols. This completes the proof of this theorem.

□

Thus, the mutual information is determined between the source symbols X and $\tilde{L}_A$, the log-likelihood ratio (LLR) on the *a-priori* source probability, and also between X and $\tilde{L}_E$, the extrinsic information LLR at the output of the constituent decoder.

An EXIT curve for a specific consitutent decoder i plots $I(X;\tilde{L}_{Ei})$ against $I(X;\tilde{L}_{Ai})$ where i can be either *1* or *2* depending on the constituent decoder being used. When iterative decompression is performed, the output of one constituent decoder becomes the input (*a-priori*) information to the other decoder. By appropriately flipping the input/output axis, the EXIT curves of the two constituent decoders can be combined into an EXIT chart as explained in detail in Section 3.1. The chart allows one to track the progress of iterative decoding for two concatenated decoders. In case a tunnel exists between the EXIT curves, the source data can be decompressed correctly, while a closed tunnel signifies unsuccessful decoding for a given source.

3.3.4 EXIT Chart Example for a Turbo Compressive Code

This section presents an example of an EXIT chart constituting individual EXIT curves. The code used for this analysis is a parallel turbo compressive non-linear code with a rate of *4*-to-*2* taken from [46]. The EXIT chart in Fig. 3.13 is constructed using the methodology explained in the previous sub-section. The tunnel between the EXIT curves is closed in Fig. 3.13 (c) signifying that successful decoding is not possible. It also shows the predicted iterative trajectory of the turbo decoder. On the other hand, the tunnel is open in Fig. 3.13 (b) an (c) which indicates that successful decoding is possible, as is illustrated by the fact that the iterative trajectory reaches the maximum entropy point. Thus, the wider the tunnel, the less number of iterations are needed to successfully decode the data. Note that the iterative trajectory illustrated in Fig. 3.13 (a), (b) and (c) is solely predicted by the EXIT chart and ideally should match the actual trajectory of the iterative decoder as shown later in Section 3.3.5 and also in Chapter 4.

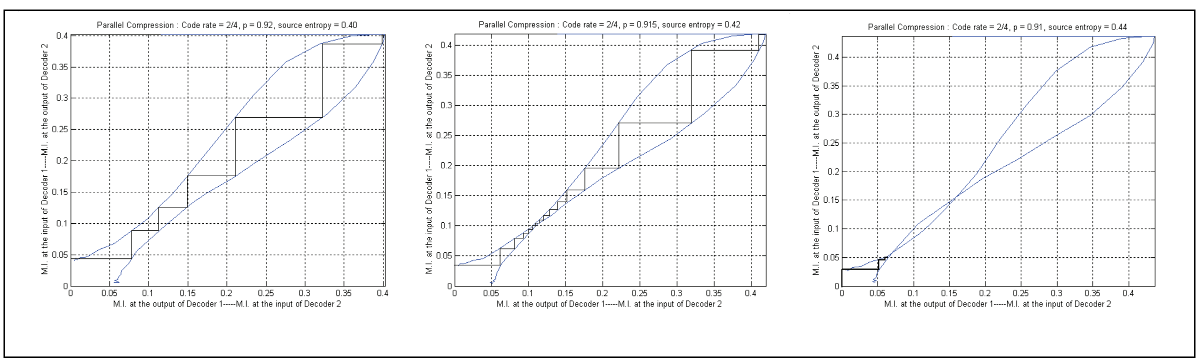


Fig. 3.13 EXIT chart for a compressive code with rate *4*-to-*2* and entropy (clockwise from top left) (a) *H(X)=0.40*, (b) *H(X)=0.42* and (c) *H(X)=0.44*. The solid line shows the iterative trajectory *predicted* by the corresponding EXIT chart for the turbo decoder

3.3.5 Differences from Published Work

This sub-section presents an example to discuss the difference in the constructed EXIT charts using the formula obtained in this thesis in Theorem 3.2 and the one specified by Haganeuer in [32]. The code used for this example/analysis is the same parallel turbo compressive non-linear code with a rate of *4*-to-*2* that was used for the example in Fig. 3.13.

As shown in Theorem 3.3, the formula given in [32] to calculate the mutual information is a lower bound and thus is not exact. Fig. 3.14 was plotted using the formula given in [32] and it can be seen that the EXIT chart extends into the negative mutual information area i.e. beyond the origin. This is not accurate since the mutual information is a non-negative function, as shown in Theorem 2.3 and eq. (2.15).

For comparison, the EXIT charts constructed using both the formula obtained in Theorem 3.2 and the one proposed in [32] are shown in Fig. 3.15. As can be seen, the difference is quite significant, especially for large values of p i.e. as p approaches 1, the disparity in both the formulas increases. This is intuitive as according to Corollary 3.2, due to the fact that both the formulas are equal if and only if $p=0.5$. Hence, the large the variation from $p=0.5$, the bigger is the deviation in the corresponding EXIT charts.

To verify the actual trajectory, a turbo coded system was simulated and the actual iterative trajectory of a turbo coded system was plotted (in terms of the mutual information) as well in Fig. 3.15, represented as the dashed lines. This was done for a *4*-to-*2* turbo compressive code with a source bias of $p=0.75$, $p=0.85$, $p=0.81$ and $p=0.925$ in Fig. 3.15(a), (b), (c) and (d) respectively. It can be seen that the actual trajectory is quite close to the one predicted by the EXIT chart built using the formulas in this thesis i.e. from Theorem 3.2 as opposed to the EXIT chart constructed using the methodology proposed in [32]. It is worth noting that in Fig. 3.15 (c), the EXIT chart constructed using the formula in this thesis (Theorem 3.2) is accurate in predicting the closed tunnel i.e. unsuccessful decoding, whereas the one constructed from the formula proposed in [32] predicts successful decoding. Note that when compared to the original methodology from [32] (dotted-dashed EXIT lines), the EXIT chart construction proposed in this thesis (solid lines) matches the *actual* simulation data (dashed lines) much more accurately, while also successfully predicting failure of the iterative decoder in Fig. 3.15 (c). This further corroborates the claim that the formula presented in this work (Theorem 3.2) is accurate.

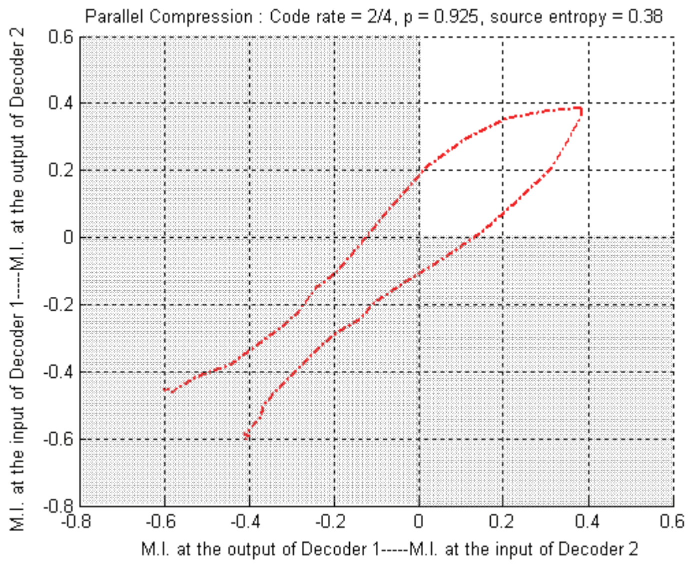


Fig. 3.14 An EXIT chart based on the original methodology proposed by Hagenauer [32] for parallel compressive turbo codes, as applied to a rate *4*-to-*2* code and a binary data source with *p=0.925*. (Note that in the gray region the mutual information calculated by this methodology is negative, thus violating the basic result of information theory about non-negativity of mutual information.)

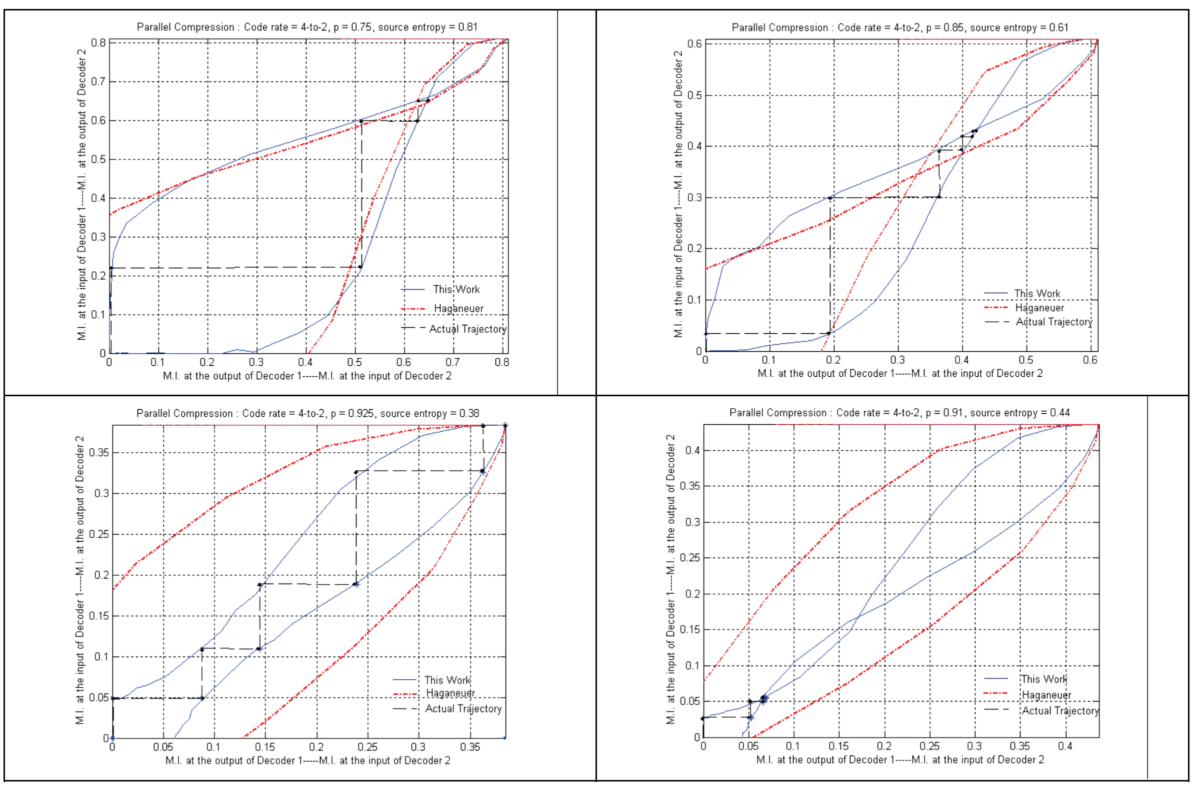


Fig. 3.15 Differences in the EXIT charts constructed in this thesis (solid lines) and the ones based on Haganeuer's methodology (dotted lines) [32], both applied to a parallel compressive turbo code of rate *4*-to-*2* and binary memoryless sources with *p=0.75*, *0.85*, *0.91* and *0.925* (clockwise from top left). The actual decoding trajectory of the iterative turbo decoder is represented by the dashed lines.

3.4 Analysis of Analog Iterative Decoders

EXIT charts offer a powerful method to study the traditional (digitally implemented) iterative decoders [15] and also allow accelerated design of good constituent encoders. Analog VLSI technology, on the other hand, can offer several advantages, when compared to digital VLSI implementation of algorithms [68]. Analog VLSI offers natural parallelism, single-transistor implementation of such operations as logarithm and multiplication, reduced size and power usage when compared to similar digital VLSI implementation. Nonetheless, any practical design of an analog VLSI decoder has to account for limited precision of implemented (analog) arithmetic operations, drift of current and/or voltage with time, temperature and time dependent computation results as the analog circuit ages, device mismatch, etc. Consequently, the analog VLSI decoding technology is still in early stages of active research, since there are issues to be worked out in order to make it practically feasible.

3.4.1 Overview of Analog Iterative Decoders

Due to their impressive error-correcting capabilities, turbo codes have been introduced into several data communication standards ([25], [63]). However, because of the iterative nature and data dependencies of the iterative decoding algorithm, digital implementations of these decoders potentially suffer from long latencies and high power consumption. To circumvent this problem, analog implementation of these decoders was suggested in ([30], [43]). The feasibility of an analog decoder is made possible by the robustness and relatively low precision requirements of iterative decoding algorithms, and also by the existence of very simple and compact analog decoding circuits. To summarize, an analog decoder has the following advantages [29]:

- *Low Power Consumption*: In a digital decoder, one wire for every bit of precision witnesses peak-to-peak voltage swings during the decoding process, whereas in an analog decoder the voltage generally does not swing the entire supply range.

Additionally, the A/D converters in the digital implementations consume a lot of power.

- *High Speed*: In a typical digital implementation of a decoder, the throughput is limited by the maximum speed of the AD converters or by the critical path of the decoder. As the analog decoder performs iterations in continuous time, there is no need for a high-speed clocking circuitry.
- *Smaller Area*: Due to the parallelism, an analog decoder can be several times smaller in terms of silicon requirements per decoded bit.

Nevertheless, despite these advantages, there are several impairments in analog VLSI that have to be taken into account to implement of practical analog iterative decoders. The first and the most important is device mismatch. This could be due to the transistors among other things e.g. current mirrors have typically an accuracy of *5-10%*, which amounts to the LLR resolution being limited to *5-7* bits [44]. The impact of various device mismatches can be accounted for by incorporating an additional (analog) Gaussian noise in the system [44]. This noise is added to the extrinsic outputs of each of the constituent decoders in the turbo coded system. Furthermore, thermal variations affect an analog decoder circuit, but it has been shown in [44] that the thermal effects do not have a noticeable impact on the performance of such decoders.

An important aspect of an analog decoder is the decoding process encountered in its VLSI implementation. Analog decoders comprise of analog circuitry and do not follow the discrete iterations performed by similar digital iterative decoders. This is due to the fact that there is a continuous flow of currents (and voltages) between the decoders until they settle down to a stable state. An analog turbo decoder is a continuous-time asynchronous network that includes a continuous time feedback loop, i.e., two constituent decoders operate concurrently with continuously updated extrinsic information arrays i.e., no back-and-forth iterations occur. After settling down, hard decisions are made on the soft outputs from the decoders using a bank of comparators and the decoded data usually matches results from traditional digital turbo decoder.

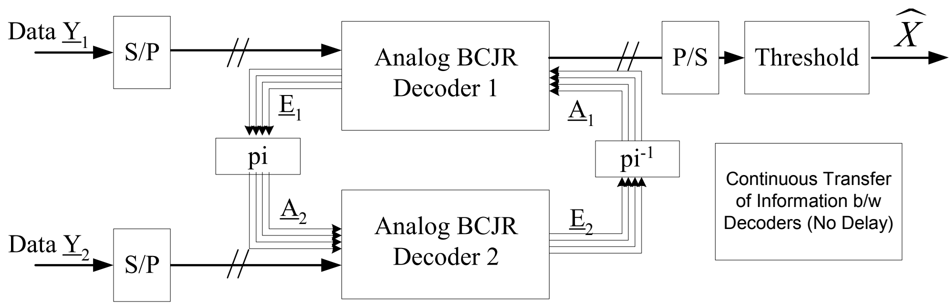


Fig. 3.16 Block diagram of an analog iterative decoder for a paralell concatenated turbo code. P/S and S/P denote the parallel-to-serial and serial-to-parallel data converters.

3.4.2 EXIT Chart Methodology for Analog Iterative Decoding

This section proposes an extension of the EXIT charts method to the performance analysis of analog iterative decoders. The focus is on how to incorporate main analog impairments and specifics (e.g., limited precision of analog circuitry, no discrete iterations, device mismatches and variations in actual analog circuits) into construction of an appropriate EXIT chart. Note that this is an "algorithmic solution" that can be useful in early design of an analog iterative decoder, i.e. choice of constituent encoders and decoders, choice of device mismatch, circuit parameters (voltage, size, substrate technology).

To analyze the performance of analog iterative decoders using EXIT charts, it is vital to bridge the gap between the analog and the digital domain. Fig. 3.17 shows a transitional model between the analog and the digital iterative decoders. The links between the two are due to the limitations that arise when implementing iterative decoders using analog circuitry. Before explaining the details of the EXIT chart extensions, it is imperative to put forth some definitions, so that the notation and convention used later is this sub-section is clear.

Definition 3.4: *Analog Decoders*

A BCJR decoder can be implemented in the analog domain (e.g. using analog VLSI circuitry) by using a sequence of trellis modules. This type of a decoder is known as the *analog BCJR decoder*. An *analog iterative decoder* (also known as analog turbo decoder) can be constructed using constituent analog BCJR decoders, as shown in Fig. 3.16.

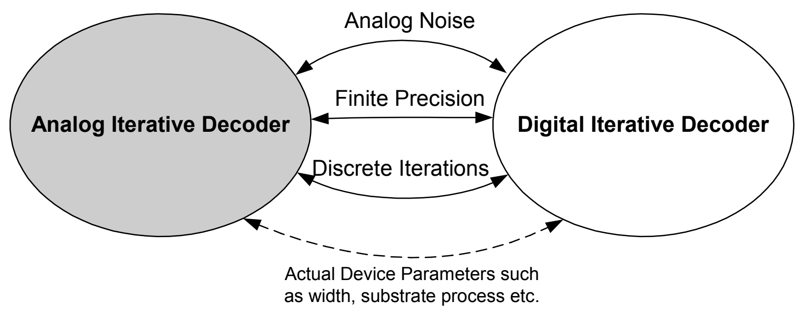


Fig. 3.17 Analog to the Digital domain transition model

Definition 3.5: *Device Mismatch*

The parameters of two identically designed devices on an integrated circuit show a random variation after fabrication. This is commonly known as *device mismatch* and causes variation in observed information quantities produced by analog decoders as well as variation of the decoding time [44]. In general, device mismatch has much more impact on analog circuits than on digital circuits.

To construct an EXIT chart for an analog iterative decoder, the following model of an analog BCJR decoder will be adopted. (Note that the BCJR (MAP) algorithm, used in this definition, is described in detail Appendix C.1)

Definition 3.6: *Computational Model for an Analog BCJR Decoder*

A *digital BCJR algorithm*, implemented for the same trellis code, will be altered to capture the effects of device mismatch in an analog BCJR decoder:

a. *Finite precision data representation* will be used for all quantities used in the BCJR algorithm to model limited accuracy of analog data in the analog BCJR decoder, i.e., the forward and backward recursion metrics $\alpha(\bullet)$ and $\beta(\bullet)$, the source/channel metric $\gamma_i(\bullet)$, the likelihood ratios for the decoded bits.

b. Variation in observed analog information quantities will be represented by *AWGN noise added to extrinsic information vectors* at the input and output of the BCJR algorithm

c. Variation of the decoding time will be modeled, to a first order approximation, by some *additional AWGN noise added during data transmission* through the channel [29]. The more the AWGN noise, the longer the decoder will take to decode the data successfully.

Parameters of the quantities in items a), b) and c) of the above definition can be calibrated based on the actual analog decoder circuit technology. For instance, the number of bits representing the integer and decimal part of the (finite precision) information quantities can be selected in such a way that it reflects the actual precision of a particular analog decoder implementation.

Definition 3.7: *Convergence Behavior and Decoding Speed of Digital and Analog Iterative Decoders*

A *digital iterative decoder* converges through the process of *discrete iterations* that consist of the exchange of extrinsic information between the two constituent decoders. In other words, a constituent decoder passes on its result after it finishes decoding to the other constituent decoder, as explained in Section 2.4.2. The other constituent decoder processes this information and then passes its result to the first constituent decoder. This procedure continues for a prescribed number of iterations or until the decoders converge to the same solution.

Analog iterative decoders do not follow discrete iterations, as the output extrinsic information is instantaneously propagated to the other constituent decoder, i.e. there is a *continuous flow of information* between the decoders, as shown in Fig. 3.16. This continuous information exchange continues until the decoders reach a steady state or a predetermined time has passed.

Consequently, the *speed of decoding* is defined as the time it takes for the BCJR or iterative decoder to successfully converge.

- In a digital BCJR decoder, this is the time it takes for the given channel output and extrinsic information input into the decoder to converge to the correct solution.

- In a digital iterative decoder, this is equivalent to the time taken by each of the iterative cycles multiplied by the number of such cycles (or iterations).
- In both an analog BCJR decoder and an analog iterative decoder, the speed of decoding is the time it takes for the analog decoder to settle (or converge) to a steady state.

Definition 3.8: *LLR Notation for Analog Iterative Decoders*

Similar to Definition 3.1, for the construction of the EXIT charts and the modeling of the constituent BCJR decoders in an analog iterative decoder, the LLR notation has to be clarified. In an actual analog iterative decoder, L_{Ai}^{Analog} is used for *a-priori* LLR at the input of the constituent decoder *i* and L_{Ei}^{Analog} is used for *a-posteriori* LLR at the output of the constituent decoder *i*, where *i* can be either *1* or *2* depending on which decoder is being used.

On the other hand, in an approximate model of a analog constituent BCJR decoder *i* used for the construction of the EXIT chart, $\tilde{L}_{Ai}^{Analog}$ and $\tilde{L}_{Ei}^{Analog}$ is used for the *a-priori* and *a-posteriori* LLR of the constituent decoder *i* respectively, where *i* can be *1* or *2*.

Definition 3.9: *Analog EXIT Chart*

An *analog EXIT chart* of an analog iterative decoder consists of two *analog EXIT curves*, each constructed for the appropriate computational model of the corresponding analog BCJR decoder. The methodology from Section 3.2.1 and 3.3 is applied to the construction of the analog EXIT curves, as shown in

Fig. 3.18, with the modifications shown in gray. The only difference is that $I\left(X,\tilde{L}_{Ai}\right)$ and $I\left(X,\tilde{L}_{Ei}\right)$ are replaced with $I\left(X,\tilde{L}_{Ai}^{Analog}\right)$ and $I\left(X,\tilde{L}_{Ei}^{Analog}\right)$ respectively as shown in Fig. 3.18.

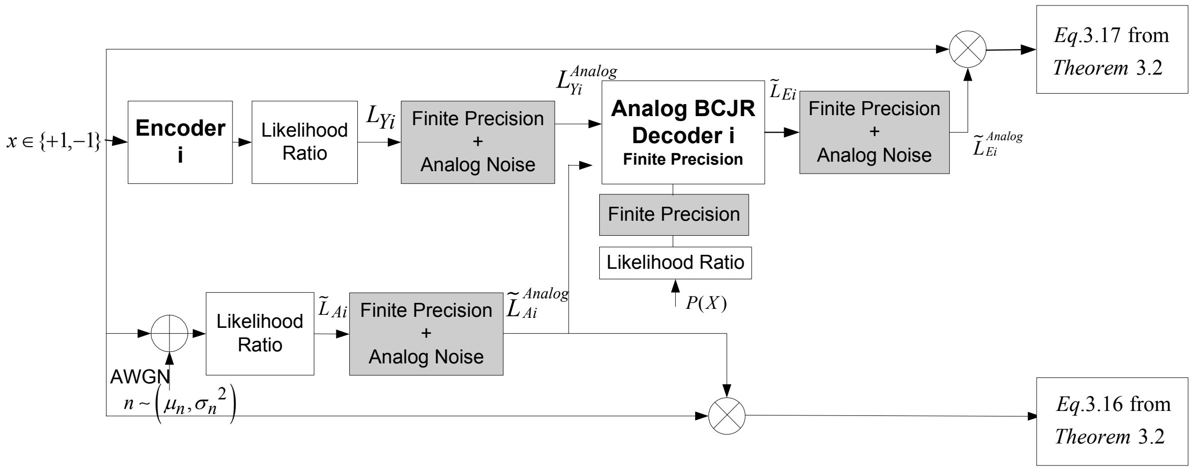


Fig. 3.18 System model for an constituent decoder used in the construction of the Analog EXIT chart for a parallel concatenated code. Note that the blocks in gray are used in the system model to incorporate AWGN noise and finite precision to fully characterize an analog decoder.

The following theorem explains how and why the defined analog EXIT charts can be used to study convergent behavior of analog iterative decoders.

Theorem 3.4: (Analog EXIT Charts)

a) If the analog EXIT chart has a closed tunnel, the analog iterative decoder does not converge.
b) Conversely, if the constituent analog BCJR decoders operate faster than the equivalent digital BCJR decoders, an open tunnel in the analog EXIT chart indicates that the analog iterative decoder will converge (successfully).

Proof of the Theorem:

A model of an *analog iterative decoder* is shown in Fig. 3.19. This model includes the *finite precision* and *analog noise* modules to capture the effects of device mismatch, as mentioned in Definition 3.6. Furthermore, to make the transition from the digital to the analog domain it is vital to make another modification to the *analog iterative decoders*, so that discrete iterations do take place. This modification is shown in Fig. 3.19 for a parallel concatenated system, where the decoders are connected with *hold modules* at their inputs

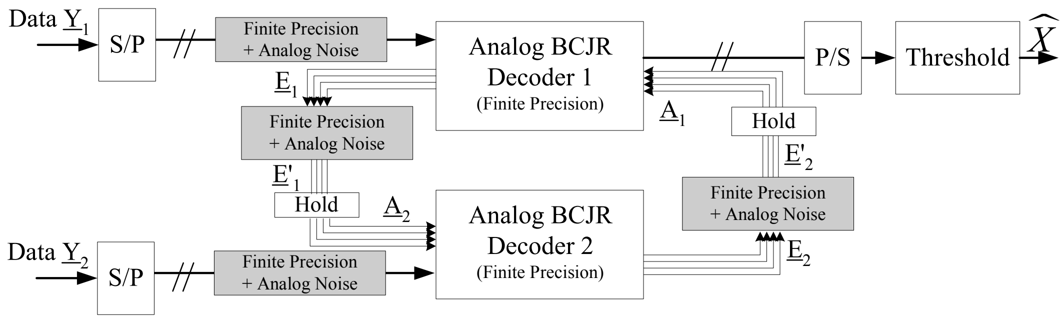


Fig. 3.19 Analog decoder in a discrete iterative setup

to disconnect them. When the hold module in on, the line is disconnected and the output is maintained at its previous level. Conversely, when the hold module is off, the data passes through normally. In other words, when decoder *1* is working, the hold module at the input of decoder *2* is off while the one in front of decoder 1 is on and vice versa. This is done to ensure that when decoder *1* is doing its job i.e. processing information, the information at the input of that decoder is not updated.

At any given time instant, the mutual information quantities $I\left(X, L_A^{Analog}\right)$ and $I\left(X, L_E^{Analog}\right)$ of an analog iterative decoder specify a point (x, y) that is located *between* the two EXIT curves of the corresponding analog constituent BCJR decoders. This is shown in Fig. 3.20. At this particular time instant, both the decoders are "pulling" the mutual information towards their EXIT curves due to the fact that there is continuous exchange of information. Thus, the result is that a path (shown by a solid curve between the two *analog EXIT curves* in Fig. 3.20) is formed which takes the two decoders to the point of stability, i.e., successful decoding. Thus, as shown in Fig. 3.20, the mutual information trajectory for an analog decoder is specified by the path in between the two EXIT curves. An important point to note here is that the process of discrete iterations of a digital iterative decoder is the extreme case scenario (in terms of *speed of decoding*) for the analog iterative decoder to converge. This is due to the fact that in the digital case, one decoder processes the information while the other one is shut off, and vice versa. This can only happen in the analog decoder if the other decoder is shut off ("frozen") which is the extreme case as both the *analog BCJR decoders* in the analog iterative decoding system function continuously. The other extreme is that there is absolutely no delay involved for the constituent *analog BCJR decoders* to process information, in which case the iterative trajectory (in terms of mutual information) resembles the dotted line between the two a*nalog EXIT curves* in Fig. 3.20. Thus, without any loss of generality, it can be assumed that an *analog iterative decoder* follows the trajectory in between these two extreme cases and that this trajectory bears a resemblance to the one shown in the solid curved line in Fig. 3.20.

It is important to note that the *speed of decoding* is related to the circuit used to implement the analog iterative decoder, thus is circuit specific, but it is known for a fact that generally an *analog BCJR decoder* operates faster than the corresponding *digital BCJR decoder* [29]. Also, note that this technique is only as good as the model i.e. the model has to be changes if the technology and/or other vital parameters are changed.

To illustrate the use of the analog EXIT charts proposed in this section, Fig. 3.21 shows as an analog EXIT chart constructed for a turbo compressive code with rate *4*-to-2. The used *finite data precision* is *7* bits for the integer part and *3* bits for the decimal part. The impact of the *analog circuitry noise* can be seen in Fig. 3.21 (a) and (b), where the analog noise is increased from *0.2* to *0.8*, respectively, and the EXIT chart tunnel closes. Consequently, successful analog iterative decoding is possible in the first case but is not possible in the second case.

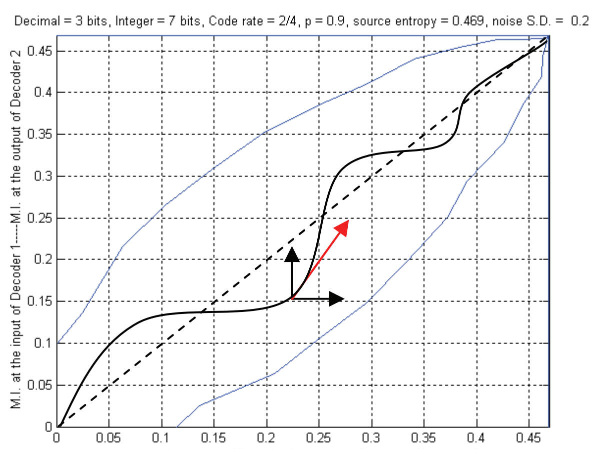
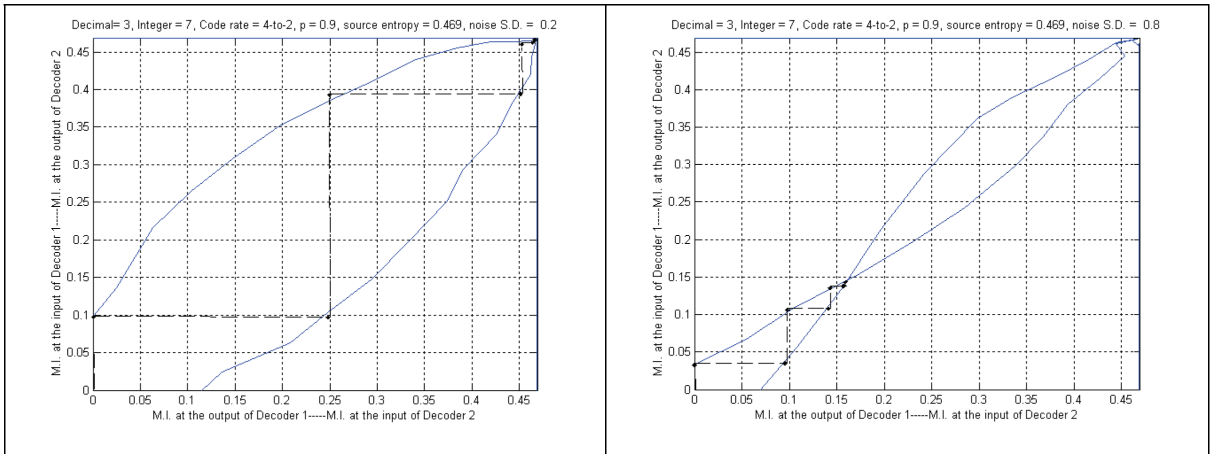


Fig. 3.20 EXIT charts trajectory for an analog iterative decoder for a parallel compressive code of rate *4*-to-*2* with finite precision (Decimal = *3* bits and Integer = *7* bits) and Gaussian analog noise S.D. *0.2*



(a) **(b)**

Fig. 3.21 EXIT charts for a parallel compressive code of rate *4*-to-*2* with finite precision (Decimal = *3* bits and Integer = *7* bits) and Gaussian analog noise S.D. (a) *0.2* and (b) *0.8* . The actual decoding trajectory of the iterative turbo decoder is represented by the dashed lines.

3.5 Chapter Summary

In this chapter, several new results have been derived and presented, in order to extend the recently proposed EXIT chart performance evaluation technique of turbo channel codes. In particular, a formula was first derived for calculation of mutual information used for the EXIT chart analysis of compressive turbo codes. The derived formula allows proper EXIT chart analysis of such compressive codes and will be further applied and extended in Chapter 4 of this thesis. Furthermore, a related lower bound has been also derived on the mutual information used in the construction of the compressive EXIT charts. Finally, the EXIT chart technique was also extended to allow performance evaluation of recently considered analog iterative decoders.

Chapter 4 Applications of EXIT Charts

This chapter presents several numerical results illustrating the usefulness of the EXIT chart technique developed in Chapter 3. Firstly, several single source turbo compression schemes are analyzed including the case with finite precision iterative decoders. Consequently, EXIT charts are used to analyze the performance of compressive turbo codes used for the Slepian-Wolf problem and practical Wyner-Ziv coding schemes. Selected numerical results are compared to simulation results in order to verify proper functioning of the developed EXIT charts. Note that the general methodology for the construction of EXIT charts was implemented in [54] and has been used in this chapter with modifications which will be further explained throughout the chapter. Additionally, the EXIT chart results presented in [54] used the same codes as the ones used in the following sections and the work in [54] was presented in part in [55] and [57].

4.1 Single Source Turbo Compression

Initially, the EXIT chart analysis of compressive turbo codes developed in Chapter 3 has been successfully applied to various single-source compression schemes. This section describes these results in detail.

4.1.1 EXIT Chart Results with Infinite Precision Arithmetics

We consider turbo compression of a single binary memoryless source X with $P(X=+1)=p$, while using both parallel concatenated turbo scheme from [46] and serial concatenated turbo scheme [21]. The EXIT charts in this sub-section are constructed using the methodology presented in Section 3.1 and 3.3. Note that the EXIT charts presented in [54] used the old formula given by Haganeuer in [32], whereas the EXIT charts presented in this sub-section for the same codes and similar source parameters are constructed using the improved formula presented in Section 3.3. For the EXIT charts shown in Fig. 4.1 and Fig. 4.2, a binary memoryless source is used with different entropies for a parallel compressive code of rate *3*-to-*2* and *4*-to-*2* (*2-to-1*) respectively. The standard deviation of the noise for the a-priori channel (from Fig. 3.6) is varied over a range from *0.05* to *2.55*. In addition to that, each of these simulations is performed for *500,000* source bits to get an average from (3.32) and (3.33) (Theorem 3.2) and the resulting mutual informations are plotted.

The EXIT charts in Fig. 4.1 a) and b) show the successful convergence of a *3*-to-*2* parallel compressive turbo code for $H(X)=0.56$ and $H(X)=0.50$ *bits* respectively. (Related BER simulations in [46] give an error of 10^{-5} for source entropy *0.609* bits.) Conversely, when $H(X)=0.57$ *bits,* the EXIT chart tunnel is closed in Fig. 4.1 c), i.e., successful decoding is not possible. Consequently, as the source entropy is decreased, the EXIT tunnel becomes wider i.e. the decoding takes less iterations as shown in Fig. 4.1 a).

Similarly, the EXIT charts in Fig. 4.2 a) and b) predict the successful convergence of a *2*-to-*1* parallel compressive turbo code for $H(X)=0.42$ and $H(X)=0.40$ *bits*. (Related BER simulations in [46] give an error of 10^{-5} for source entropy *0.456* bits.) When the entropy is increased to $H(X)=0.44$ *bits,* the EXIT chart tunnel is closed in Fig. 4.2 c), i.e., successful decoding is not possible.

Fig. 4.1 (and Fig. 4.2) show that the parallel concatenated coding scheme works for source entropies lower than $H(X)=0.56$ (and *0.42*) respectively, but for entropies higher than $H(X)=0.56$ (and *0.42*) the code does not seem to work properly. The open “tunnel” signifies that the decoding iterations are successful in recovering the sent message sequence, whereas if the “tunnel” is closed than the decoder is unable to fully recover the sent message. Note that the EXIT charts have a maximum at the source entropy as shown in Fig. 4.1 .

On the other hand, Fig. 4.3 and Fig. 4.4 show the EXIT charts for a serial concatenated turbo compression scheme. The codes used in the serial concatenation consist of a *1*-to-*2* code ([21], Appendix C.2]) followed by a *3*-to-*1* or a *4*-to-*1* code also used in the parallel turbo compression scheme above. Thus, the code rate of this serial concatenation in Fig. 4.3 and Fig. 4.4 is 3-to-2 and 4-to-2 respectively. Fig. 4.3 (a) and (b) show EXIT charts with an open tunnel signifying successful convergence with the source entropies of *H(X)=0.48* and *0.54* bits respectively. Conversely, Fig. 4.3 (c) shows an EXIT chart with a closed tunnel for a source entropy *H(X)=0.58* bits. As explained earlier, the closed tunnel suggests that successful decoding is not possible in this case. It is worthy to note that Fig. 4.3 (b) where the source entropy *H(X)=0.54* bits is the threshold whereby the EXIT tunnel is almost closed. In other words, for *H(X)>0.54* successful decoding would not be possible as the EXIT tunnel would be closed. Similarly, Fig. 4.4 (a) and (b) show EXIT charts with an open tunnel signifying successful convergence, whereas the EXIT chart in Fig. 4.4 (c) demonstrates a closed tunnel implying unsuccessful decoding.

To verify the proper functioning of the EXIT chart technique, Fig. 4.1 to Fig. 4.4 also show the actual iterative decoding trajectory along with the numerically evaluated EXIT charts, both for a parallel and serial concatenated turbo compression scheme. The actual trajectory represented by the dashed lines matches well with the numerically evaluated EXIT chart, thus confirming the reliability of the EXIT chart performance analysis technique.

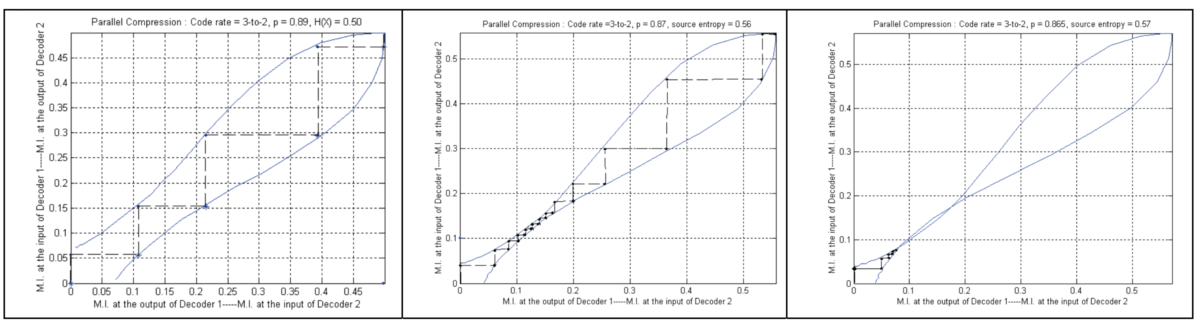


Fig. 4.1 EXIT Chart for a parallel compressive code with rate *3*-to-*2* and entropy (left to right) (a) *H(X) = 0.50*, (b) *H(X) = 0.56* and (c) *H(X) = 0.57*. The actual decoding trajectory of the iterative turbo decoder is represented by the dashed lines.

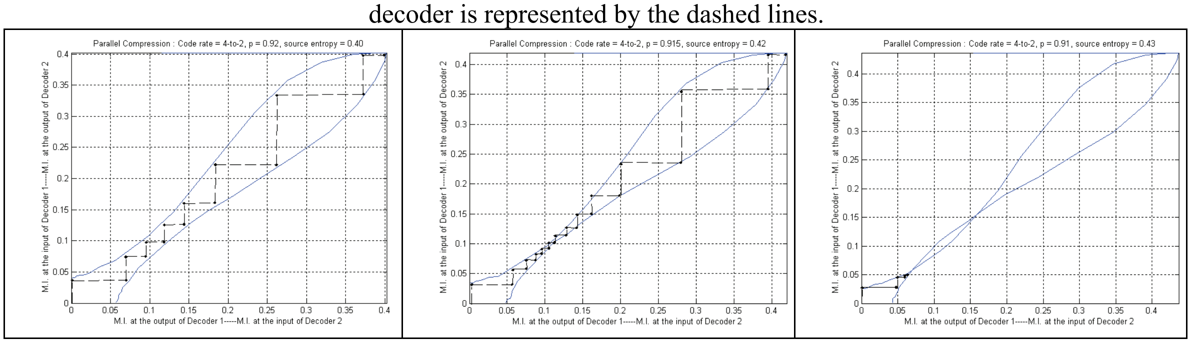


Fig. 4.2 EXIT chart for a parallel compressive code with rate *4*-to-*2* and entropy (left to right) (a) *H(X) = 0.40*, (b) *H(X) = 0.42* and (c) *H(X) = 0.44*. The actual decoding trajectory of the iterative turbo decoder is represented by the dashed lines.

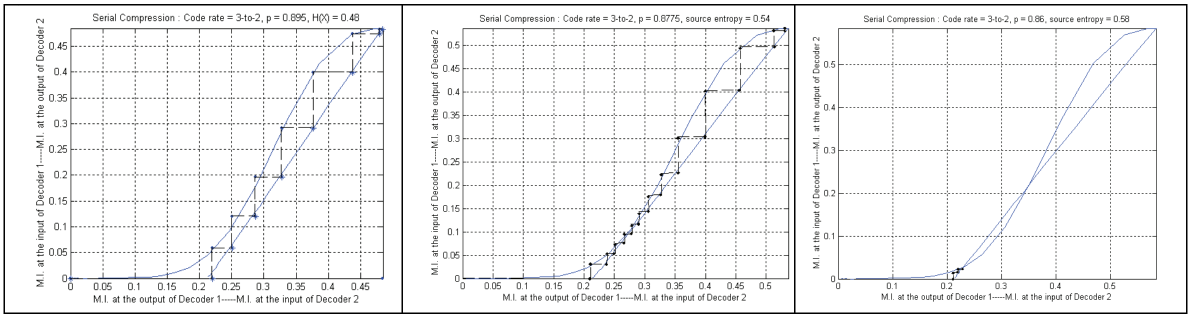


Fig. 4.3 EXIT chart for a serial compressive code with rate *3*-to-*2* and entropy (left to right) (a) $H(X) = 0.48$, (b) $H(X) = 0.54$ and (c) $H(X) = 0.58$. The actual decoding trajectory of the iterative turbo decoder is represented by the dashed lines.

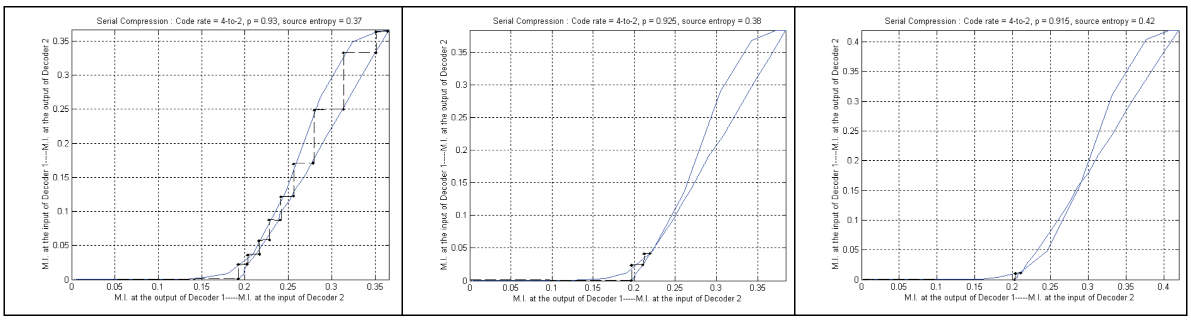


Fig. 4.4 EXIT chart for a serial compressive code with rate *4*-to-*2* and entropy (left to right) (a) $H(X) = 0.37$, (b) $H(X) = 0.38$ and (c) $H(X) = 0.42$. The actual decoding trajectory of the iterative turbo decoder is represented by the dashed lines.

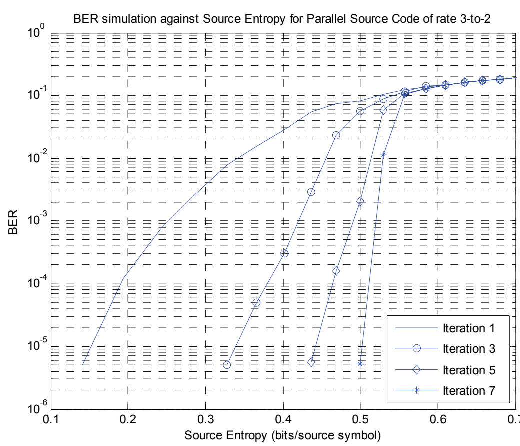
Table 4.1 shows the numerical results obtained for different code rates, compared to the ones obtained from extensive BER simulations.

Code Rate	**Source Entropy – *H(X)* (bits)**	
	EXIT Chart Numerical Results	*BER (10^{-5}) Simulation Results*
3-to-2 parallel → 3:1 and 3:1	0.560	0.609*
4-to-2 parallel → 4:1 and 4:1	0.420	0.456*
5-to-2 parallel → 5:1 and 5:1	0.320	0.343*
3-to-2 serial → 1:2 with 3:1	0.540	0.540**

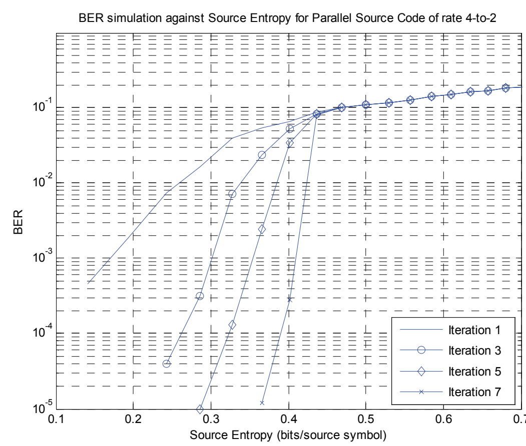
Table 4.1 Results for turbo compression of a single source. Numerically predicted starts of the BER "waterfall" region and results based on simulated BER performance are compared (*- Reference [46], [47] and **- Reference [21])

Table 4.1 compares the EXIT chart numerical results obtained in this sub-section with BER simulation results for both parallel ([46], [47]) and serial concatenated schemes [21]. Both the EXIT chart and BER results provide the value of the source entropy *H(X)* corresponding to the start of the "waterfall" region.It can be deduced that there is a good match between the EXIT charts and the BER simulations resutls for both the parallel and serial concatenated schemes.

Note that the results presented in Table 4.1 are taken from [46], [47] and [21], which are generated in quite similar but not completely identical conditions. Thus, to further corroborate the EXIT chart results, BER simulations have been performed and are shown below in Fig. 4.5 and Fig. 4.6 for parallel and serial concatenated schemes respectively. It can be seen from each of these figures that the start of the waterfall region matches well with the EXIT chart results presented earlier. For example, in Fig. 4.5 (a) the waterfall region starts at a source entropy of *H(X) = 0.56* bits/source symbol where the EXIT chart also showed that the EXIT tunnel is almost closed at of *H(X) = 0.56* bits/source symbol, thus verifying that infact there is a good match between the numerical EXIT chart results and BER simulations.

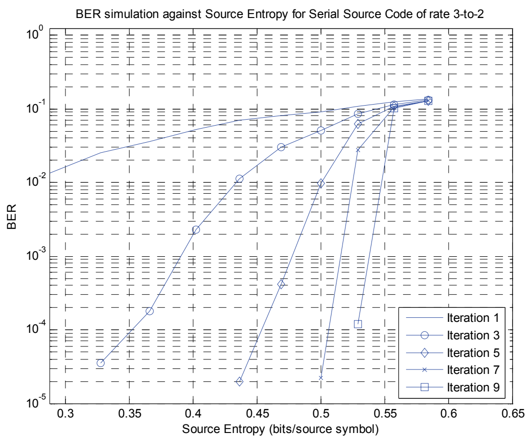


(a)

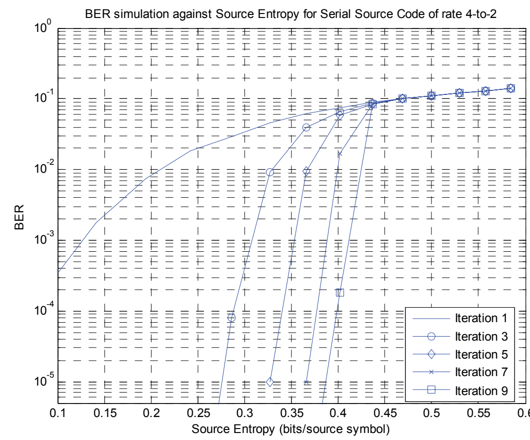


(b)

Fig. 4.5 Simulated BER curves for a (a) rate *3*-to-*2* parallel compressive turbo code for various iterations *and a (b)* rate *4*-to-*2* parallel compressive turbo code for various iterations



(a)



(b)

Fig. 4.6 Simulated BER curves for a (a) rate *3*-to-*2* serial compressive turbo code for various iterations *and a (b)* rate *4*-to-*2* serial compressive turbo code for various iterations

4.1.2 EXIT Chart Results for Finite Precision Decoders

The main objective of this sub-section is to extend the EXIT charts performance analysis technique to turbo iterative decoders with practical arithmetic limitations. This is motivated by practical implementation of the turbo decoding algorithms in FPGA and VLSI technologies, where a vital design issue is the number of bits used to represent the quantities calculated during the decoding process. Several BER simulation results have been reported regarding this issue (e.g., in [62]), but EXIT charts have not been used in this setting so far.

A binary memoryless source X is considered and its entropy $H(X)$ is varied by changing the parameter $P(X_n=+1)=p$. The encoders and decoders used are identical to the ones used in Section 3.3.2 and Section 4.1.1. The extrinsic information vectors at the output of the constituent compressive decoders are approximately conditionally normally distributed as observed in ([11], [15], [32]) and as explained in Section 3.1 and 3.3. Thus, for the purposes of constructing the EXIT curve of an individual decoder, we will use a replacement scheme shown in Fig. 4.7 which is quite similar to the one shown in Section 3.2.3 with the following changes:

- *Finite precision data representation* will be used for all quantities used in the BCJR algorithm to model limited accuracy of analog data in the analog BCJR decoder, i.e., the forward and backward recursion metrics $\alpha(\bullet)$ and $\beta(\bullet)$, the source/channel metric $\gamma_i(\bullet)$, the likelihood ratios for the decoded bits. This is modeled by the finite precision BCJR decoder shown in Fig. 4.7 gives an example of the decoder model for the case of a parallel concatenated turbo compression scheme and is analogous to Fig. 3.6. Similar models can be used for Fig. 3.7 which presented the decoder models for the outer and inner code respectively of a serial concatenated turbo compression scheme.
- The input LLR's $\tilde{L}_{Ai}$ are replaced with LLR's $\tilde{L}_{Ai}^{Finite}$ for $\tilde{A}_i = x + n$ where n is an independent Gaussian random variable with zero mean and variance $\sigma_n^{\,2}$.
- The output LLR's, $\tilde{L}_{Ei}$, are replaced with $\tilde{L}_{Ei}^{Finite}$.
- The channel output LLR's, L_{Yi}, are replaced with L_{Yi}^{Finite} .

Keeping this practical limitation in mind, the number of bits chosen to represent the integer and decimal part of these quantities has been varied in the BCJR turbo decoder implementation (Appendix C.1). The authors in [50] have proved that 6-7 bits for the extrinsic LLR's as well as the $\ln\alpha(\bullet)$, $\ln\beta(\bullet)$ and $\ln\gamma_i(\bullet)$ quantities are enough to ensure that the performance is almost identical to the system which uses "infinite" precision. The term "infinite" here refers to the maximum number of bits that can be represented in a computer simulation environment.

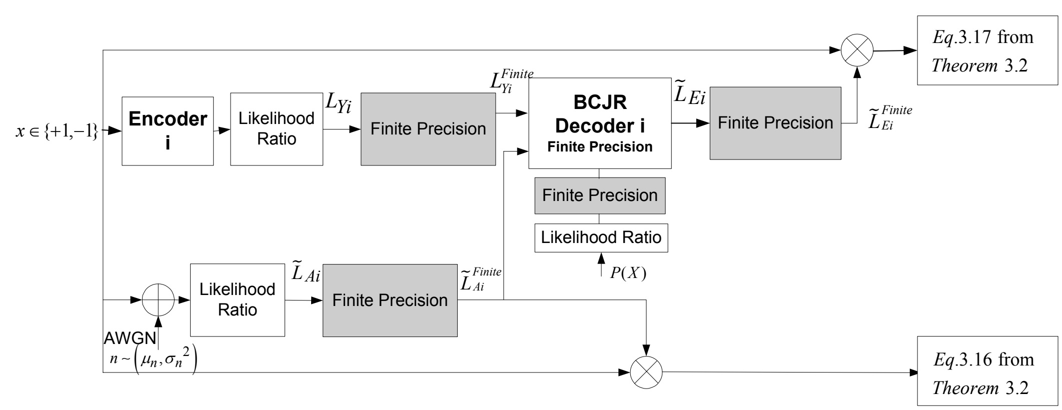


Fig. 4.7 Example of a replacement scheme used to generate EXIT curve of a constituent encoder/decoder in a parallel concatenated turbo compression scheme.

An EXIT curve for a specific decoder plots $I\left(X,\tilde{L}_{Ei}^{Finite}\right)$ against $I\left(X,\tilde{L}_{Ai}^{Finite}\right)$, i.e., the mutual information at the output of a constituent decoder versus the mutual information at the input. Thus, the EXIT chart is constructed by using the mutual information pairs $\left[I\left(X,\tilde{L}_{A1}^{Finite}\right),I\left(X,\tilde{L}_{E1}^{Finite}\right)\right]$ and $\left[I\left(X,\tilde{L}_{A2}^{Finite}\right),I\left(X,\tilde{L}_{E2}^{Finite}\right)\right]$ as explained in detail in Sections 3.1 and 3.3. Fig. 4.7 illustrates the overall replacement system model for a single constituent decoder in a parallel concatenated system. Note that the same the finite precision modules (in gray) in Fig. 4.33 can be used in Fig. 3.7, along with the finite precision BCJR decoder, to get the replacement model for the consitutent decoders in a serial concatenated system.

The EXIT charts shown in Fig. 4.8 and Fig. 4.9 have been constructed using the system specifications described in Sections 3.1 and 3.3 for a parallel 4-to-2 and a serial 4-to-2 compressive turbo code. In construction of the EXIT curves, the standard deviation of the a-priori noise (from Fig. 4.7) was varied from *0.05* to *2*. The several mutual information quantities are obtained using the equations in Theorem 3.2. The mutual information curves are plotted by averaging over *100,000* source bits.

The number of used bits being *x* for the integer part and *y* for the fractional part denoted as *(x,y)*. Fig. 4.8 shows the behavior of the 4-to-2 compressive code with entropy *H(X) = 0.400.* Fig. 4.8 a) shows the EXIT chart with infinite precision whereas Fig. 4.8 b), c) and d) use a precision of (5,3), (5,1) and (4,3) respectively. In Fig. 4.8 b) the precision is set to (5,3) and the EXIT tunnel is almost the same in comparison to the one with “infinite precision” in Fig. 4.8 a). As the number of bits for the decimal precision is decreased from 3 to 1 in Fig. 4.8 c), it is seen that the EXIT tunnel still remains unchanged. In Fig. 4.8 d), when the number of bits for the decimal part is kept to 3 and the precision of the integer part is reduced to 4 bits, it is observed that the EXIT tunnel closes i.e. successful decoding is no longer possible. This shows that the turbo decoder is much more sensitive to the integer part as compared to the decimal part. This is intuitive as most of the information is in the interger part and not in the decimal part.

Similarly, the EXIT charts in Fig. 4.9 for a serial concatentated turbo compression scheme with a code rate of *4*-to-*2* show similar behavior as the EXIT charts in Fig. 4.8 for the parallel case. Thus, analogous to the EXIT charts Fig. 4.8, Fig. 4.9 b), c) and d) use a precision of *(5,3)*, *(5,1)* and *(4,3)* respectively.

To verify the decoding trajectory of the actual iterative decoder against the predicted EXIT trajectory for the iterations, the iterative decoder system was simulated for the parallel and the serial case. The results of the actual iteration path are shown in Fig. 4.8 and Fig. 4.9, where it can be seen that the actual iteration path tends to follow the EXIT chart trend. For the case of finite precision decoders, it can be seen that the actual iteration path has a slight mismatch or offset compared to the actual path for the case of "inifinite" precision decoders shown in Fig. 4.1 to Fig. 4.4. This could be attributed to the truncation of the decoding quantities to a finite number of bits and also the fact that the interleaver depth is finite in this case, as the EXIT chart technique assumes infinite interleaver depth.

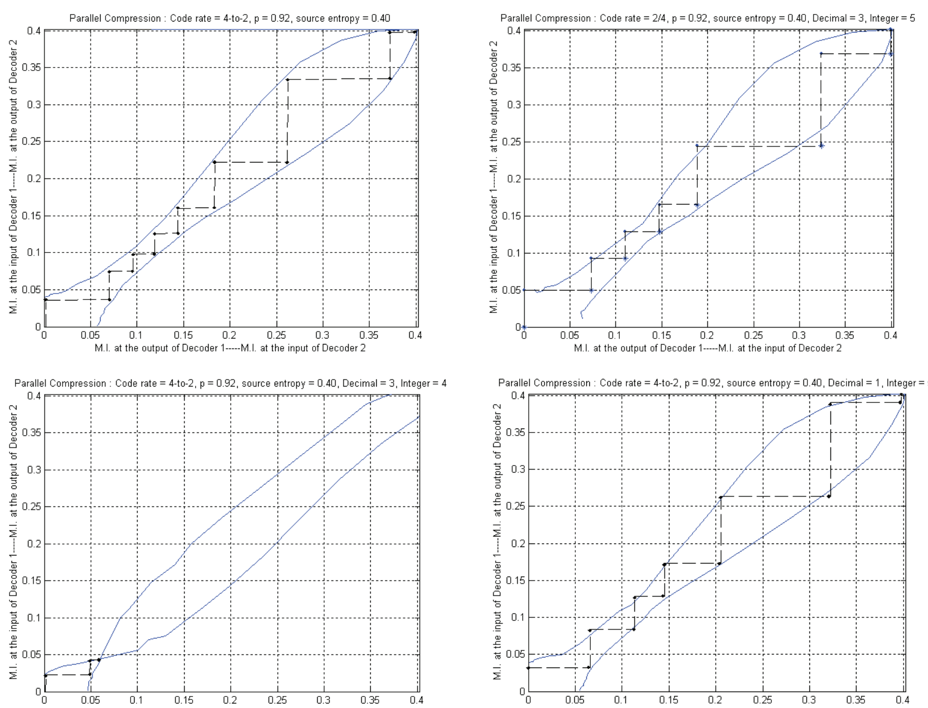


Fig. 4.8 EXIT charts for a rate *4*-to-*2* parallel compressive turbo code with *H(X) = 0.400*: (clockwise from top-left) a) "*infinite*", b) *(5,3)*, c) *(5,1)*, d) *(4,3)*. The actual decoding trajectory is represented by the dashed lines.

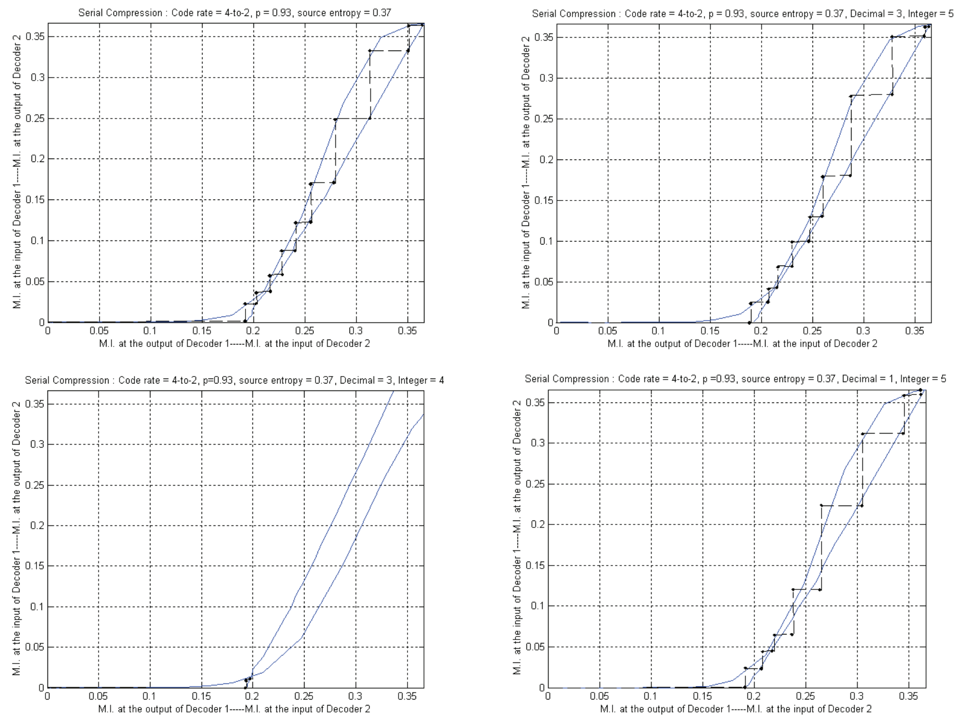


Fig. 4.9 EXIT charts for a rate *4*-to-*2* serial compressive turbo code with *H(X) = 0.37*: (clockwise from top-left) a) *infinite*, b) *(5,3)*, c) *(5,1)*, d) *(4,3)*. The actual trajectory is represented by the dashed lines.

To further corroborate the EXIT chart results, BER simulations have been performed in Fig. 4.10 for the parallel concatenated scheme. It can be seen from each of these figures that the start of the waterfall region matches well with the EXIT chart results presented earlier. For example, in Fig. 4.10 (a) the waterfall region starts at a source entropy of *H(X) = 0.46* bits/source symbol for a precision of *(5,3)* whereas for a precision of *(4,3)* the BER simulation has an error floor as confirmed by Fig. 4.8 (d).

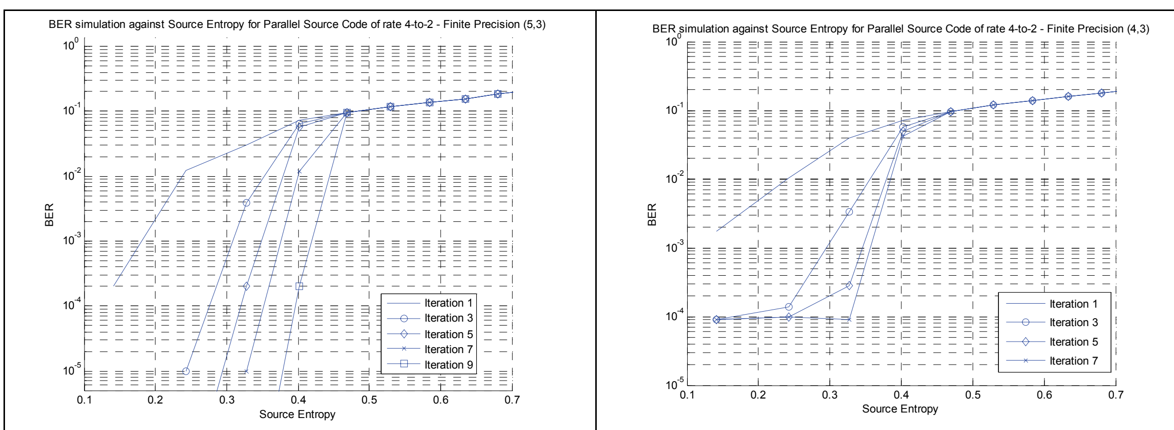


Fig. 4.10 Simulated BER curves for a rate 4-to-2 parallel compressive turbo code with a precision of (a) left (5,3), (b) right (4,3)

4.2 Slepian-Wolf Turbo Compression

This section presents numerical EXIT chart results for the performance analysis of recently proposed compressive turbo codes used in the Slepian-Wolf distributed compression setting. Section 4.2.1 reviews the EXIT chart results obtained for compression of correlated, unbiased binary memoryless sources in [54] and [57]. Furthermore, Sections 4.2.2 and 4.2.3 extend the EXIT chart technique for the Slepian-Wolf turbo compression of biased correlated sources. It is important to note that the results obtained in [54] and [57] for correlated, unbiased binary memoryless sources would not scale to biased sources using the mutual information formula from [32]. Instead, the generalized formulas from Theorem 3.2 is used and the EXIT chart numerical results for this particular setting are presented in Section 4.2.3 for turbo compression of biased correlated sources. It is noted that the numerical results obtained from the constructed EXIT charts give a good match to the BER simulation results obtained through extensive Monte-Carlo simulations. Thus, generalized EXIT charts are a strong tool which can adequately explain the behavior of the iterative decoding process for turbo codes used in the Slepian-Wolf problem.

4.2.1 EXIT Charts for SW Turbo Compression of Un-Biased Sources

The BER results shown in Fig. 3.9 are for single source compression schemes. The *3-to-2* code used above, along with various other codes of different rates, can also be used for Slepian-Wolf compression of discrete binary memoryless sources. The details of the Slepian-Wolf coding were given in Section 2.3.1. Essentially, when two encoders cannot converse with each other, the rates at which the data has to be encoded (for the two sources X and Y) is R_X and R_Y. On the other hand if X and Y are encoded together, the optimal encoding rate would be their joint entropy, denoted by $R_X + R_Y = H(X, Y) = H(X) + H(Y|X)$, as defined in Section 2.3.1 [61]. Amazingly, Slepian and Wolf have proven that in the case of separate encoding, this joint entropy is also the compression limit, and that it can be reached asymptotically.

For the performance analysis of the considered Slepian-Wolf compression scheme, a pair of unbiased binary memoryless correlated sources X and Y are considered, such that

$$\begin{aligned}\Pr(x_i = +1, y_i = +1) = \Pr(x_i = -1, y_i = -1) = \frac{p}{2} \\ \Pr(x_i = -1, y_i = +1) = \Pr(x_i = +1, y_i = -1) = \frac{1-p}{2}\end{aligned} \tag{4.1}$$

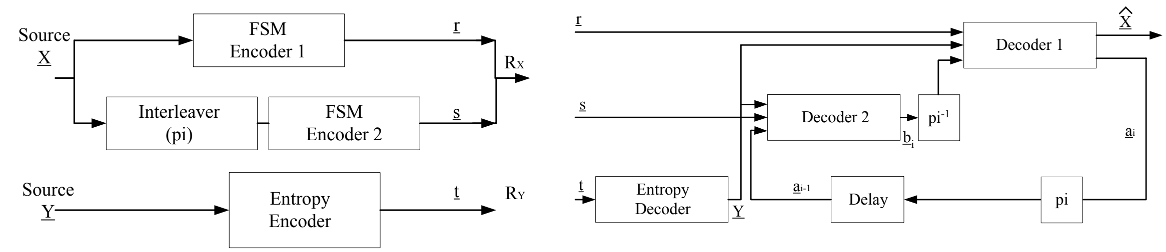
Consequently, it can be written as:

$$\begin{aligned}&\Pr(x_i = -1 \mid y_i = -1) = \Pr(x_i = 1 \mid y_i = 1) = p \\ &\Pr(x_i = -1 \mid y_i = 1) = \Pr(x_i = 1 \mid y_i = -1) = \frac{(1-p)\bullet 0.5}{0.5} = 1-p\end{aligned} \tag{4.2}$$

where x_i and y_i are the i^{th} bits of the sources X and Y respectively. The conditional entropy $H(X|Y)$ is varied by changing the correlation parameter p between the sources, as

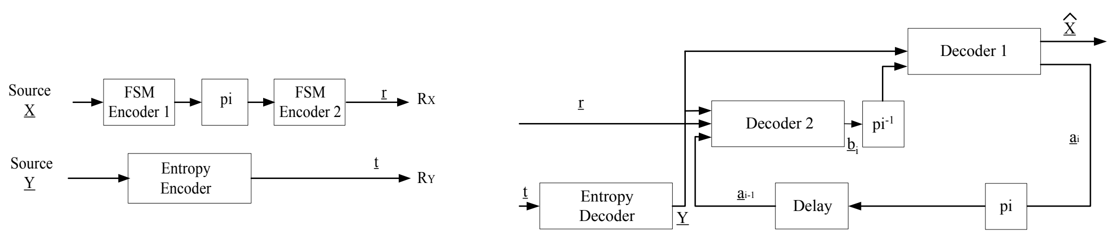
$$\left.\begin{aligned}H(X \mid Y) &= -\sum_{x\in[-1,+1]}\sum_{y\in[-1,+1]} P(x,y)\bullet\log_2(P(x \mid y)) \\ &= \quad -(p)\bullet\log_2(p) - (1-p)\bullet\log_2((1-p))\end{aligned}\right\} \tag{4.3}$$

The system used for this encoding/decoding is shown in Fig. 4.11 and Fig. 4.12 for a parallel and serial turbo compression scheme. For the parallel case, the encoder structure consists of two parallel concatenated FSM encoders whereas for the serial case, the encoders consist of serial concatenation of two FSM encoders. The FSM encoder for Y is a zero error encoder which makes it possible for Y to be used as the side information at the joint decoder to decode X. Consequently the data from source X is encoded at a rate close to $H(X|Y)$ which is below $H(X)$, as shown in Theorem 2.3, but with a low decoding probability of error this is possible according to the Slepian-Wolf theorem.



(a) (b)

Fig. 4.11. Slepian-Wolf parallel (a) encoder and (b) decoder ([48], [61])



(a) (b)

Fig. 4.12. Slepian-Wolf serial (a) encoder and (b) decoder ([21], [61])

The EXIT charts shown in Fig. 4.13 to Fig. 4.15 have been constructed using unbiased correlated memoryless sources with different conditional entropies [54]. The procedure for the EXIT chart construction is the same as explained in Section 3.1 and 3.3 with the exception that the side information $P(X)$ is replaced with $P(X|Y)$. In construction of the EXIT curves that constitute the final EXIT chart, the standard deviation of the noise for the a-priori channel is varied over a range from *0.05* to *2.55*. Each of these simulations is performed for *500,000* source bits to obtain a numerical average from the mutual information quantities in Theorem 3.2.

Initially, a *3*-to-*2* parallel compressive code is used to encode a source with *H(X) = 1* bit. Although the code rate is lower than the source entropy, the availability of the correlated source *(Y)* at the decoder makes it possible to decode the first source *(X)*. The EXIT charts in Fig. 4.13 a) and b) show the successful convergence of a *3-to-2* parallel compressive turbo code for *H(X|Y) = 0.529* and *0.563 bits* respectively. Related BER simulations, presented in [48], give an error of 10^{-5} for source conditional entropy *0.612* bits. When *H(X|Y)* increased to *0.634 bits*, the EXIT chart tunnel closed, as illustrated in Fig. 4.13 c), i.e., successful decoding was not possible.

To verify the decoding trajectory of the actual iterative decoder against the predicted EXIT trajectory for the iterations, the encoder/decoder system is simulated and the result of the actual iteration path is shown in Fig. 4.13, Fig. 4.14 and Fig. 4.15. It can be seen that the actual iteration path is quite close to the EXIT chart showing that indeed the EXIT chart is a good measure of the iterative decoding process and can be reliably used to study performance of the presented codes.

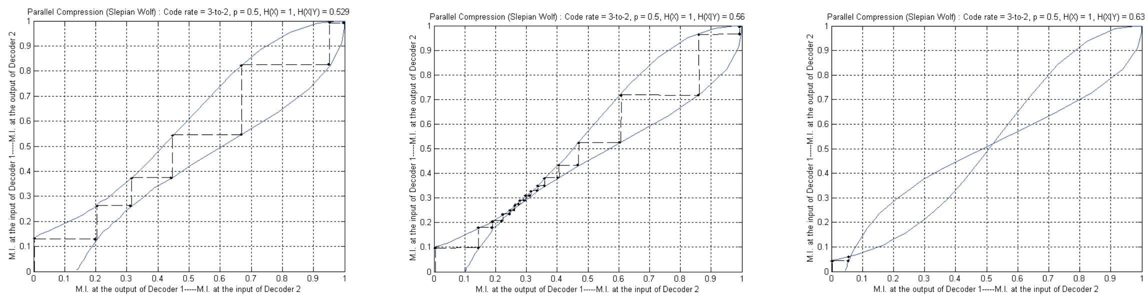


Fig. 4.13 EXIT charts for a rate *3*-to-*2* Slepian-Wolf parallel compressive turbo code including the actual decoding trajectory: (left to right) a) *H(X|Y) = 0.529* , b) *H(X|Y)=0.563*, c) *H(X|Y)=0.634*

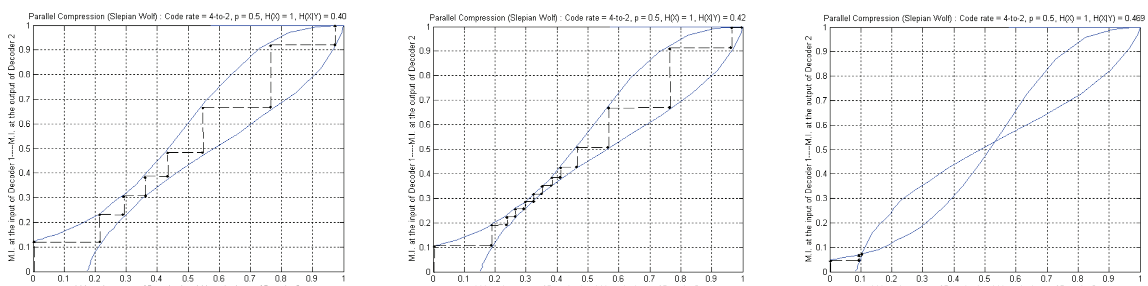


Fig. 4.14 EXIT chart for a rate *4*-to-*2* Slepian-Wolf parallel compressive turbo code including the actual decoding trajectory: (left to right) a) *H(X|Y) =0.402*, b) *H(X|Y)=0.420*, c) *H(X|Y) = 0.469*

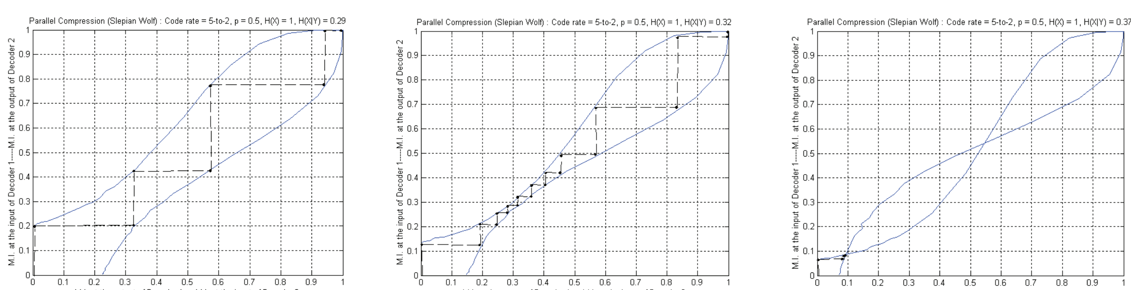


Fig. 4.15 EXIT chart for a rate *5*-to-*2* Slepian-Wolf parallel compressive turbo code including the actual decoding trajectory: (left to right) a) *H(X|Y) =0.286*, b) *H(X|Y)=0.320*, c) *H(X|Y) = 0.366*

In addition to the *3*-to-*2* compressive turbo code, two more codes have been tested as shown in Fig. 4.14 (rate *4*-to-*2* compressive code) and Fig. 4.15 (rate *5*-to-*2* compressive code). The same methodology has been used in these cases, as for the above mentioned *3*-to-*2* compressive turbo code. The EXIT charts in Fig. 4.14 a), Fig. 4.14 b), Fig. 4.15 a) and Fig. 4.15 b) show successful convergence whereas the ones in Fig. 4.14 c) and Fig. 4.15 c) demonstrate unsuccessful convergence due to the closed EXIT tunnel. The results for the considered three compressive turbo codes are summarized in Table 1, giving a good match against the BER simulations from [48].

On the other hand, Fig. 4.16 and Fig. 4.17 show the EXIT charts for a serial concatentated turbo compression scheme for the Slepian-Wolf setting with a rate of *3*-to-*2* and *4*-to-*2* respectively. The codes used in this scheme are 1-to-2 ([21], Appendix C.2) concatenated serially with a *3*-to-*1* code or a *4*-to-*1* code used in the parallel compression scheme above. Similar to the parallel compression case, the source entropy is taken to be $H(X)=1$ for this case. Fig. 4.16 (a) and (b) show the cases when the EXIT charts have an open tunnel for conditional entropies of $H(X|Y)=0.469$ and 0.560 bits respectively. As the source entropy is increases to $H(X|Y)=0.630$, the tunnel is closed and successful decoding is no longer possible. Furthermore, the EXIT charts in Fig. 4.17 (a) and (b) show an open EXIT tunnel whereas the one in Fig. 4.17 (c) demonstrate a closed EXIT tunnel due to the increased conditional entropy of $H(X|Y)=0.469$ bits. Furthermore, Fig. 4.16 and Fig. 4.17 show the actual iterative decoding trajectory to compare with the numerically simulated EXIT charts, and it can be seen that there is good match between the two.

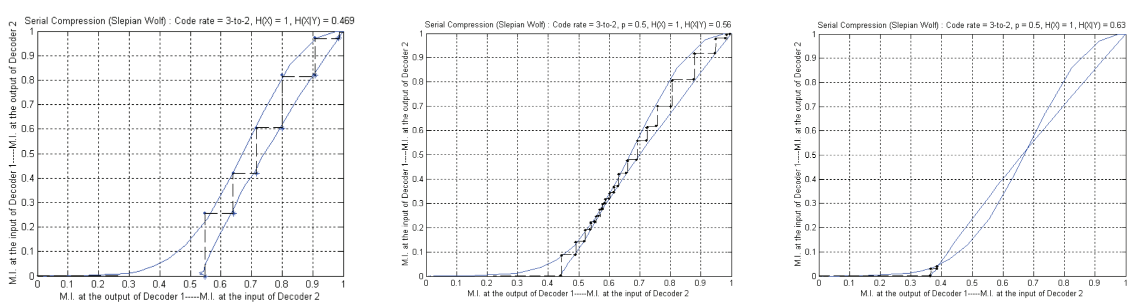


Fig. 4.16 EXIT chart for a rate *3*-to-*2* Slepian-Wolf serial compressive turbo code including the actual decoding trajectory: (left to right) a) $H(X|Y) =0.469$, b) $H(X|Y)=0.560$, c) $H(X|Y) = 0.630$

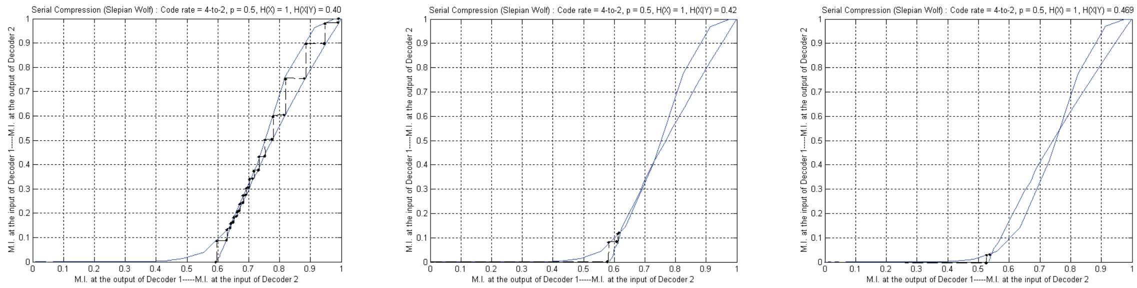


Fig. 4.17 EXIT chart for a rate *4*-to-*2* Slepian-Wolf serial compressive turbo code including the actual decoding trajectory: (left to right) a) $H(X|Y) = 0.400$, b) $H(X|Y)=0.420$, c) $H(X|Y) = 0.469$

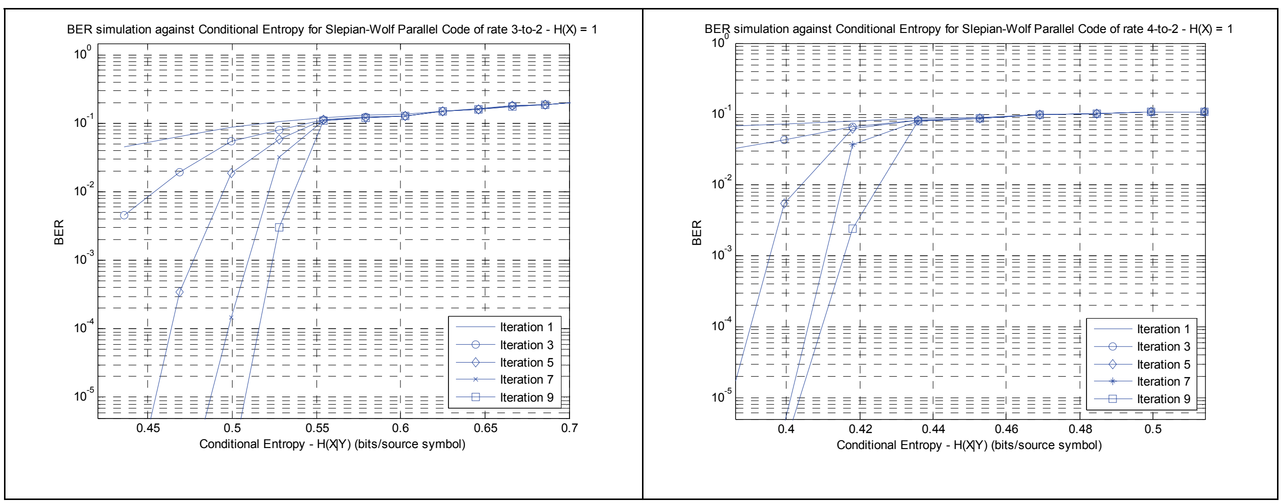
In the case of the single source compression, the EXIT charts had a maximum at the source entropy but for the Slepian Wolf compression, the source that is compressed has an entropy of $H(X)=1$ i.e. $P(x_i = +1) = 0.5$, thus the EXIT charts have a maximum at *1* as shown in Fig. 4.13 to Fig. 4.16. Using the EXIT charts designed previously and the method described above, the results for correlated sources are shown below in Table 4.2:

Code Rate	**Conditional Entropy – *H(X|Y)* (bits)**	
	*EXIT Chart Numerical Results****	*BER (10^{-5}) Simulation Results*
3-to-2 parallel → 3:1 and 3:1	0.563	0.612*
4-to-2 parallel → 4:1 and 4:1	0.420	0.450*
5-to-2 parallel → 5:1 and 5:1	0.320	0.343*
3-to-2 serial → 1:2 with 3:1	0.560	0.550**

Table 4.2 Results for distributed turbo compression for the Slepian-Wolf problem. Numerically predicted starts of the BER "waterfall" region and results based on simulated BER performance are compared. (*- Reference [47], [48], **- Reference [21], ***- Reference [54])

Table 4.2 compares the EXIT chart numerical results [54] obtained in this sub-section with BER simulation results for both parallel ([46], [47]) and serial concatenated schemes [21]. Both the EXIT chart and BER results provide the value of the conditional entropy $H(X|Y)$ corresponding to the start of the "waterfall" region. It is shown that there is a good match between these resutls for both the parallel and serial concatenated schemes.

Note that the results presented in Table 4.2 are taken from [47], [48] and [21], which are generated in quite similar but not completely identical conditions. Thus, to further corroborate the EXIT chart results, BER simulations have been performed and are shown below in Fig. 4.18 and Fig. 4.19 for parallel and serial concatenated schemes respectively. These figures show the BER simulations against the conditional entropy *H(X|Y)* for a number of iterations. As an example, in Fig. 4.18 (a), the BER simulation has been shown for a 3-to-2 parallel compressive Slepian-Wolf code with a source entropy of *H(X) = 1*. In Fig. 4.18 (a) the waterfall region starts at a source entropy of *H(X|Y) = 0.56* bits/source symbol where the EXIT chart also showed that the EXIT tunnel is almost closed at of *H(X) = 0.56* bits/source symbol. The almost closed EXIT tunnel corresponds to the start of the waterfall region in the corresponding BER simulation. It can be seen from each of these figures that the start of the waterfall region matches well with the EXIT chart results presented earlier in Fig. 4.13 to Fig. 4.17. This verifies that infact there is a good match between the numerical EXIT chart results and BER simulations.



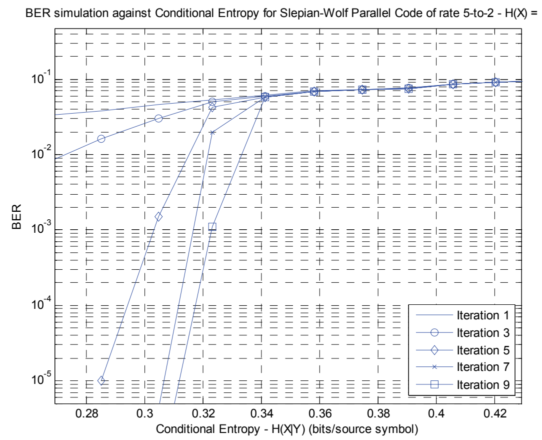
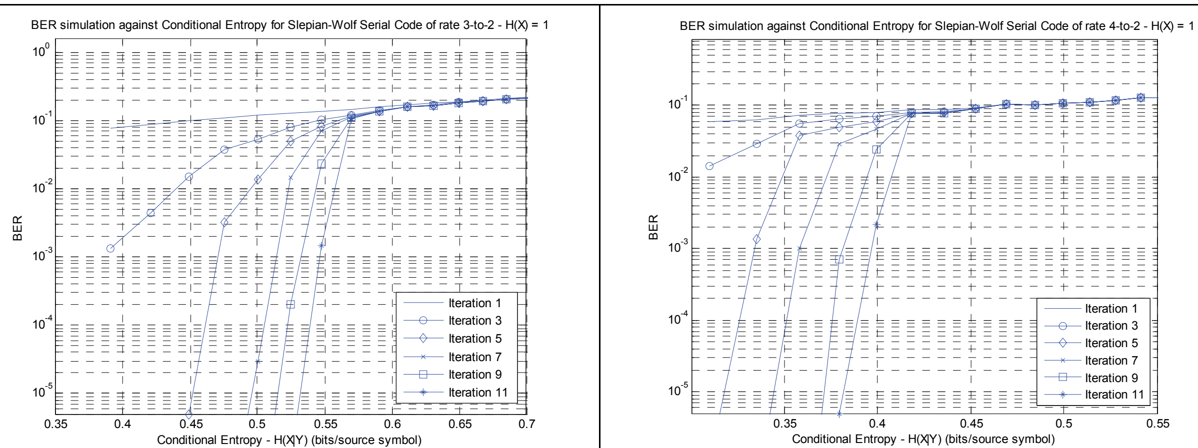


Fig. 4.18 Simulated BER curves for a parallel compressive Slepian-Wolf turbo code with a source entropy *H(X) = 1* for a rate (clockwise from top-left) (a) *3*-to-*2* , (b) *4*-to-*2* and (c) *5*-to-*2* for a number of iterations



(a) (b)

Fig. 4.19 Simulated BER curves for a serial compressive Slepian-Wolf turbo code with a source entropy *H(X) = 1* and a code rate of (a) *3*-to-*2* and (b) *4*-to-*2* for a number of iterations

4.2.2 Biased Source Model and Encoder/Decoder Structure

The main objective of this section is to extend the EXIT chart technique to biased sources in the Slepian-Wolf setting as opposed to the un-biased sources considered in Section 4.2.1. The methodology is presented on using EXIT charts in this problem and the results are illustrated for several pairs of correlated memoryless sources and turbo compression schemes, observing a good match between numerical EXIT results and the actual behavior of iterative decoders.

As explained in Section 2.3.1, the Slepian-Wolf problem deals with separate encoding of correlated sources, when joint recovery is performed at the receiver. Even though the two encoders cannot communicate with each other in this problem, the rates R_X and R_Y at which the sources X and Y can be successfully compressed and decompressed belong to an attainable region:

$$\begin{aligned} &R_X + R_Y \geq H(X, Y) \\ &R_X \geq H(X \mid Y) \\ &R_Y \geq H(Y \mid X) \end{aligned} \tag{4.4}$$

For the performance analysis of the considered Slepian-Wolf compression scheme, a pair of binary memoryless correlated sources X and Y is considered, such that the joint distribution is as follows:

$$P = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{4.5}$$

where $a = \Pr(x_i = +1, y_i = +1)$, $b = \Pr(x_i = -1, y_i = +1)$, $c = \Pr(x_i = +1, y_i = -1)$, $d = \Pr(x_i = -1, y_i = -1)$ and x_i and y_i are the i^{th} bits of the sources X and Y respectively s.t.

$\Pr(x_i = +1) = a + c$, $\Pr(x_i = -1) = b + d$, $\Pr(y_i = +1) = a + b$ and $\Pr(y_i = -1) = c + d$.

The variables a, b, c and d are chosen such that $\Pr(x_i = +1) \neq 0.5$ and $\Pr(y_i = +1) \neq 0.5$. i.e. the sources X and Y are biased with $H(X) \neq 1$ and $H(Y) \neq 1$. The conditional entropy $H(X|Y)$ is given as:

$$H(X \mid Y) = -\sum_{y \in [-1,+1]} \sum_{x \in [-1,+1]} P_{XY}(x, y) \log_2(P_{X|Y}(x \mid y)) \tag{4.6}$$

Consequently, the conditional entropy $H(X|Y)$ is varied by changing the parametes a, b, c and d to satisfy the above conditions. Fig. 4.11 shows the overall block diagram for the considered Slepian-Wolf compression scheme, originally proposed in [61]. The points of interest in the achievable region are the "corner" points in Fig. 2.1 (b), when one source is compressed close to its conditional entropy $H(X|Y)$. The other source Y is compressed close to its entropy *H(Y)*, so that the message sequence from *Y* is available error-free after its decompression at the receiver. From the symmetry of the problem (compression at rates *H(X)* and *H(Y|X)*), transmission can also be achieved at rate pairs in the remainder of the achievable region via time-sharing, originally proposed in [61].

The encoder in Fig. 4.11 (a) encodes the source *X* with two finite-state machine (FSM) encoders based on Latin Squares. The two encoders are concatenated in parallel with an interleaver in between, as shown in Fig. 4.11 (a). Both encoders output *k* binary symbols for every *n* binary input symbols, where $2k<n$ as explained for single source compression in Section 3.3.2. On the other hand, the decoder in Fig. 4.11 (b) first decompresses the source sequence from *Y* and calculates the a-posteriori probabilities for symbols $X_1, X_2, \ldots X_n$ using equation (2.27). Consequently, the iterative decoder attempts to recover the source sequence from *X* using these a-posteriori probabilities.

4.2.3 EXIT Chart Construction for Biased Sources &Numerical Results

The EXIT charts shown in Fig. 4.13 to Fig. 4.15 have been constructed using an un-biased source. The focus of this sub-section is on a pair of biased correlated sources *X* and *Y* such that $H(X)\neq 1$ and $H(Y)\neq 1$. The procedure for the EXIT chart construction in this sub-section is the same as explained in Section 3.1 and 3.3 with the exception that the side information $P(X)$ is replaced with $P(X\,|\,Y)$ as mentioned in Section 4.2.1. In construction of the EXIT curves that constitute the final EXIT chart, the standard deviation of the noise for the a-priori channel is varied over a range from *0.05* to *2.55*. Each of these simulations is performed for *500,000* source bits to obtain a numerical average from the mutual information quantities in Theorem 3.2.

As shown in Theorem 3.3, the formula given in [32] to calculate the mutual information is a lower bound and thus is not exact. Additionally, Section 3.3.5 elaborated the difference in EXIT chart results computed from the work in this thesis and the work in [32] for a single source turbo compression scheme with a biased source. Consequently, for the case of the Slepian-Wolf compression using turbo codes for binary memoryless biased sources, the formula given in [32] cannot be used. This is further corroborated by observing the example in Fig. 4.20 which was plotted using the formula given in [32] for a source entropy *H(X) = 0.51* and conditional entropy *H(X|Y) = 0.31* using a 5-to-2 parallel compressive code. It can be seen that the EXIT chart extends into the negative mutual information area i.e. beyond the origin. This is not correct since the mutual information is a non-negative function, as shown in Theorem 2.3 and eq. (2.15).

Initially, a *3*-to-*2* compressive code is used to encode a biased source in Fig. 4.21 with $H(X) \neq 1$ and $H(Y) \neq 1$. In this case, the source entropies are larger than the code rate i.e. although the code rate is lower than the source entropy, the availability of the correlated source *(Y)* at the decoder makes it possible to decode the first source *(X)*. The EXIT charts in Fig. 4.21 a) and b) show the successful convergence of a *3-to-2* parallel compressive turbo code for a source entropy of *H(X)=0.*68 with the conditional entropy H*(X|Y) = 0.53* and *H(X|Y) = 0.55 bits* respectively. When *H(X|Y)* was increased to *0.57 bits* with the same source entropy, the EXIT chart tunnel was almost closed, as illustrated in Fig. 4.21 c), i.e., successful decoding was not quite possible. This is the maximum conditional entropy threshold for which this code can function.

In addition to the *3*-to-*2* compressive turbo code, two more codes have been tested as shown in Fig. 4.22 (rate *4*-to-*2* compressive code) and Fig. 4.23 (rate *5*-to-*2* compressive code). The same methodology has been used in these cases, as for the above mentioned *3*-to-*2* compressive turbo code. The EXIT charts in Fig. 4.22 a), Fig. 4.22 b), Fig. 4.23 a) and Fig. 4.23 b) show successful convergence whereas the ones in Fig. 4.22 c) and Fig. 4.23 c) demonstrate unsuccessful convergence due to the closed EXIT tunnel. Note that for the results in Fig. 4.21, Fig. 4.22 and Fig. 4.23, the source entropy *H(X)* was always greater than the code rate and still successful decoding was possible due to the availability of the side information *Y* at the decoder.

The EXIT charts in Fig. 4.20 till Fig. 4.23 are for parallel compressive codes. On the other hand, Fig. 4.24 and Fig. 4.25 show the EXIT charts for a serial concatentated turbo compression scheme for the Slepian-Wolf setting with code rates of *3*-to-*2* and *4*-to-*2* respectively. The codes used in this scheme are similar to the ones described in Section 4.2.1 for the serial case. In Fig. 4.24 and Fig. 4.25, the source entropy *H(X)=0.68* and

0.57 bits respectively. Similar to the parallel compression case, Fig. 4.24 (a) and (b) show cases when the EXIT charts have an open tunnel for conditional entropies of *H(X|Y)=0.53* and *0.55* bits respectively. As the source entropy is increases to *H(X|Y)=0.57*, the tunnel is closed and successful decoding is no longer possible. Furthermore, Fig. 4.25 (a) and (b) show EXIT charts where successful decoding is possible whereas in Fig. 4.25 (c) successful decoding is not possible because of the closed tunnel.

To verify the decoding trajectory of the actual iterative decoder against the predicted EXIT trajectory, the encoder/decoder system is simulated and the result shown in Fig. 4.21 to Fig. 4.25 for both the parallel and serial case respectively. It can be seen that the actual iteration path (dashed line) is very close to the EXIT chart boundaries showing that indeed the EXIT chart technique is a good measure of the iterative decoding process and can be reliably used to study performance of the presented codes.

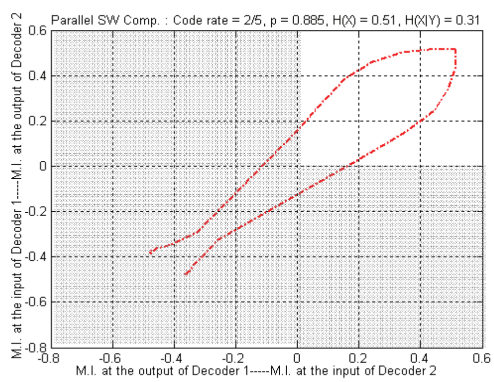


Fig. 4.20 An EXIT chart based on the original methodology proposed by Hagenauer [32] for parallel compressive turbo codes used in the SW setting, as applied to a rate *5*-to-*2* code and a binary data source with *H(X)=0.51* and *H(X|Y)=0.31*.

(Note that in the gray region the mutual information calculated by this methodology is negative, thus violating the basic result of information theory about non-negativity of mutual information.)

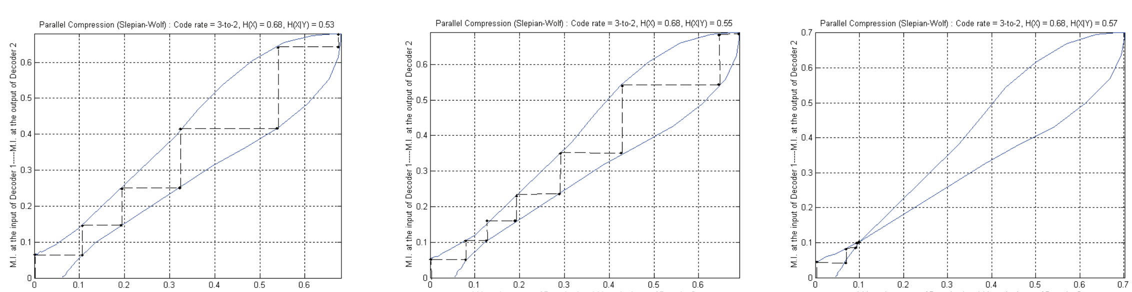


Fig. 4.21 EXIT charts for a rate *3*-to-*2* Slepian-Wolf parallel compressive turbo code with *H(X)=0.68*: and including the actual decoding trajectory (dashed line): (left-to-right) a) *H(X|Y) = 0.53*, b) *H(X|Y)=0.55*, c) *H(X|Y)=0.57*

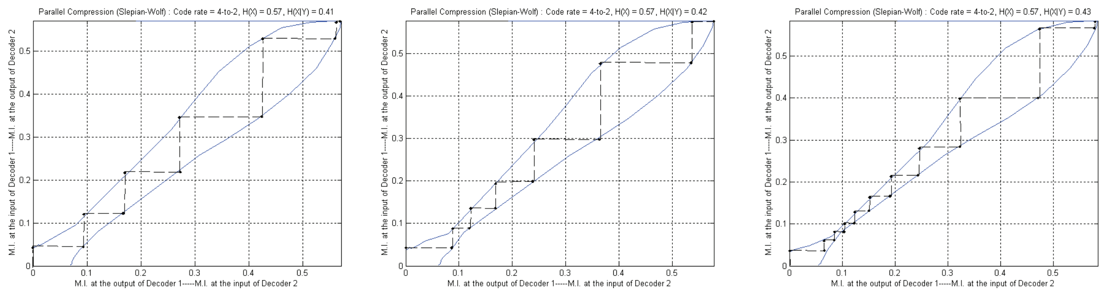


Fig. 4.22 EXIT chart for a rate *4*-to-*2* Slepian-Wolf parallel compressive turbo code with *H(X) =0.57* including the actual trajectory (dashed line):(left-to-right) a)*H(X|Y) = 0.41*, b) *H(X|Y)=0.42*, c)*H(X|Y)=0.43*

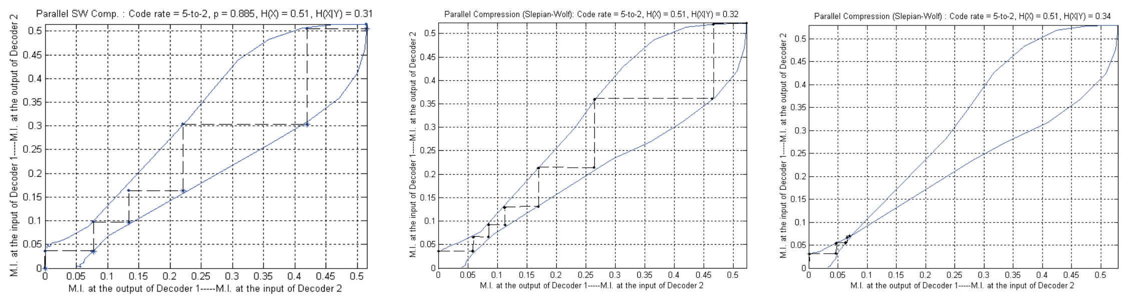


Fig. 4.23 EXIT chart for a rate *5*-to-*2* Slepian-Wolf parallel compressive turbo code with *H(X)=0.51* including the actual trajectory (dashed line): (left-to-right) a)*H(X|Y) = 0.31*, b)*H(X|Y)=0.32*, c)*H(X|Y)=0.34*

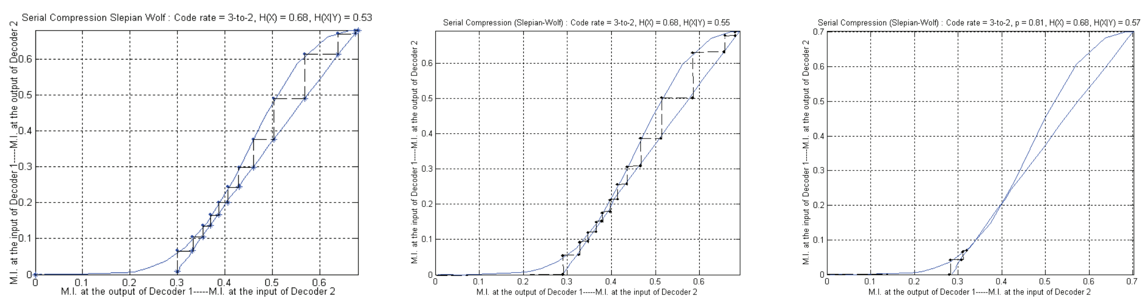


Fig. 4.24 EXIT chart for a rate *3*-to-*2* Slepian-Wolf serial comprlessive turbo code with *H(X) = 0.68* including the actual trajectory (dashed line): (left-to right) a)*H(X|Y) = 0.53*, b)*H(X|Y)=0.55*, c) *H(X|Y)=0.57*

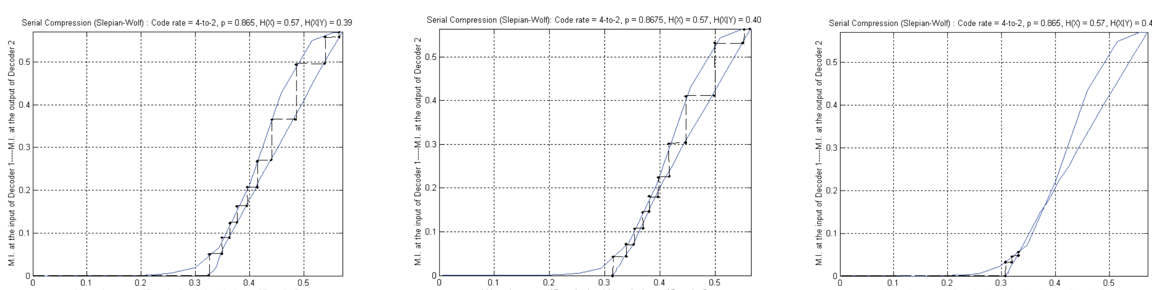
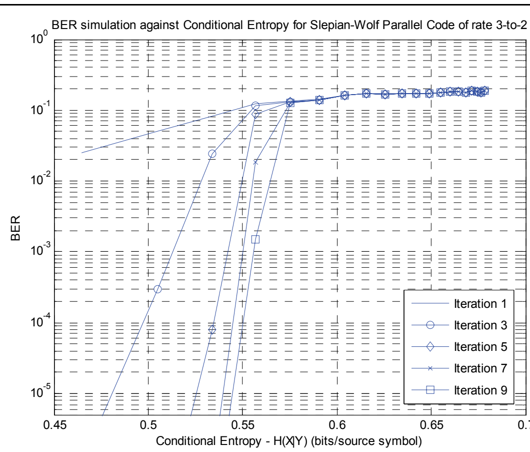
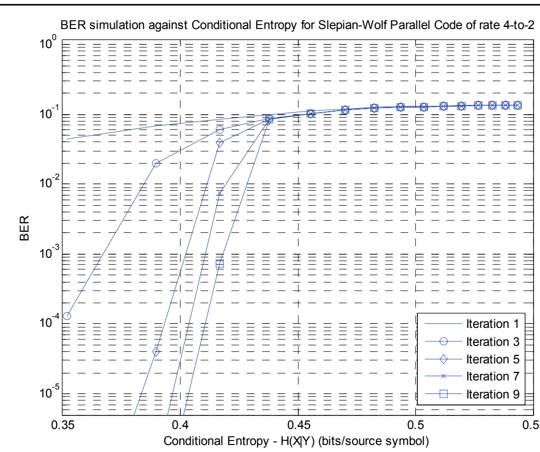


Fig. 4.25 EXIT chart for a rate *4*-to-*2* Slepian-Wolf serial compressive turbo code with *H(X) = 0.57* including the actual trajectory (dashed line): (left-to right) a)*H(X|Y) = 0.39*, b)*H(X|Y)=0.40*, c) *H(X|Y)=0.42*

To corroborate the EXIT chart results, BER simulations have been performed and are shown below in Fig. 4.26 and Fig. 4.27 for parallel and serial concatenated schemes respectively. The BER simulations shown in Fig. 4.26 (a) are for a 3-to-2 parallel compressive Slepian-Wolf code whereas Fig. 4.26 (b) shows the simulations for a 4-to-2 parallel compressive Slepian-Wolf code. Similarly, Fig. 4.27 (a) presents the BER simulations for a 3-to-2 serial compressive Slepian-Wolf code whereas Fig. 4.27 (b) shows the simulations for a 4-to-2 serial compressive Slepian-Wolf code. These figures show the BER simulations against the conditional entropy *H(X|Y)* for a number of iterations. As an example, in Fig. 4.26 (a), the BER simulation has been shown for a 3-to-2 parallel compressive Slepian-Wolf code with a source entropy of *H(X) = 0.68* bits/source symbol. In Fig. 4.27 (a) the waterfall region starts at a source entropy of *H(X|Y) = 0.57* bits/source symbol where the EXIT chart also showed that the EXIT tunnel is almost closed at of *H(X) = 0.57* bits/source symbol. The almost closed EXIT tunnel corresponds to the start of the waterfall region in the corresponding BER simulation. It can be seen from each of these figures that the start of the waterfall region matches well with the EXIT chart results presented earlier in Fig. 4.21 to Fig. 4.25. This verifies that infact there is a good match between the numerical EXIT chart results and BER simulations.

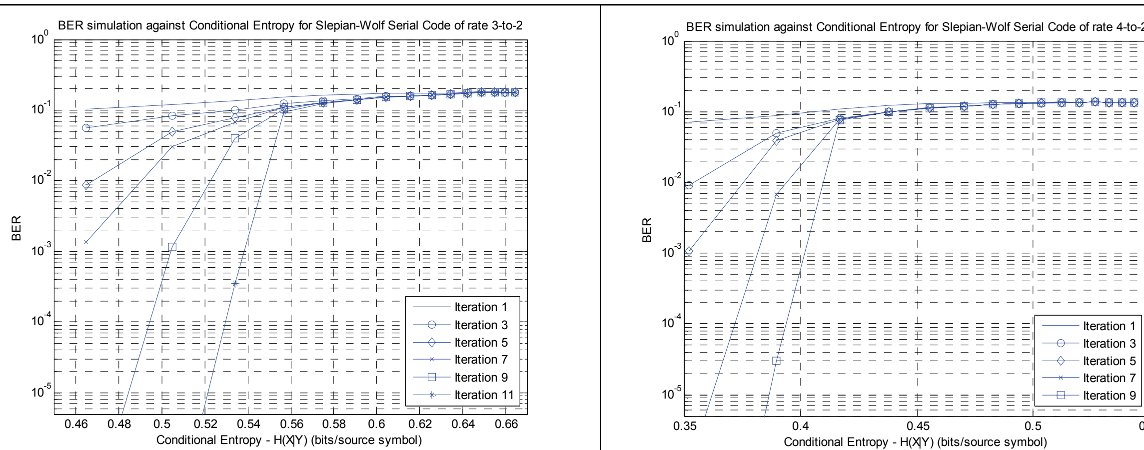


(a)



(b)

Fig. 4.26 Simulated BER curves for a parallel compressive Slepian-Wolf turbo code for a (a) source entropy H(X)=0.68 and a code rate *3*-to-*2* *and (b)* source entropy H(X)=0.57 with a code rate *4*-to-*2* for a number of iterations



(a) (b)

Fig. 4.27 Simulated BER curves for a serial compressive Slepian-Wolf turbo code for a (a) source entropy H(X)=0.68 and a code rate *3*-to-*2* *and (b)* source entropy H(X)=0.57 with a code rate *4*-to-*2* for a number of iterations

4.3 Wyner-Ziv Coding of Correlated Sources

This sub-section deals with the extension of the EXIT charts to analyze the performance of a previously proposed Wyner-Ziv lossy coding schemes based on compressive turbo codes [47], [49]. The studied scheme and several other previously proposed practical Wyner-Ziv schemes rely on the related Slepian-Wolf problem of distributed lossless compression of correlated sources. The focus of Section 4.3.1 will be on one such practical scheme.

4.3.1 Practical Wyner-Ziv Coding using Compressive Turbo Codes

The focus of this sub-section is on the implementation of a practical Wyner-Ziv coding scheme using a scalar quantizer followed by a Slepian-Wolf encoder [47], [49]. The Slepian-Wolf encoder consists of a parallel concatenation of encoders based on Latin Squares. This is not the ideal Wyner-Ziv coding scheme as the scalar quantizer is separate from the Slepian-Wolf encoder, but it saves on complexity. In the ideal case, the quantizer and the encoder are designed together.

A particular case of Wyner-Ziv deals with jointly Gaussian sources where $\mathcal{X} = \mathcal{Y} = \widehat{\mathcal{X}} = real$, $D(x,\hat{x}) = (x-\hat{x})^2$ and X, Y are jointly Gaussian with zero mean and are related by the following (without any loss in generality):

$$Y = \beta(X+U) \tag{4.7}$$

where $\beta > 0$, and X and U are independent with E[X]=E[U]=0, $E[X^2] = \sigma_X{}^2$, $E[U^2] = \sigma_U{}^2$ and $E[Y] = 0$. As explained in Section 2.3.2, for this case $R^*(d)$ is equal to:

$$R^*(d) = R_{X|Y}(d) = \begin{cases} \frac{1}{2}\log\left(\frac{b\sigma_U{}^2}{d}\right), & \text{for} \quad 0 < d < b\sigma_U{}^2 \\ 0, & \text{for} \quad d \geq b\sigma_U{}^2 \end{cases} \tag{4.8}$$

The rates of the compressive codes are 3:2 and 4:2. The notation *x:y* means that *x* input bits give *y* output bits. Thus, *x* gives the resolution of the quantizer where 2^x is the number of quantization levels. Also note that the quantizer used in Fig. 4.28 is a Lloyd Max quantizer. The rate distortion bound in Fig. 4.30 for Wyner-Ziv coding is plotted using *R=2* bits/sample as the output is always *2* bits per input sample.

The estimator at the end is responsible for recovering X from the quantized version of X which is X_Q (assuming it was recovered error free) and the side information Y. The aim is to minimize the conditional expected distortion $E\left[D(X,\widehat{X}) \mid X_Q = x_Q, Y = y\right]$. For a squared error distortion, this reduces to the conditional mean which is $\hat{x} = E\left[X \mid X_Q = x_Q, Y = y\right]$[8].

At the decoder, $P(X_Q = x_Q \mid Y = y)$ is fed as the side information to aid the decoder in the decoding process. This is needed as the conditional entropy $H(X_Q|Y)$ is less than $H(X_Q)$. A plot of the conditional entropy against the SNR of the correlation in dB (which is defined as $10\log_{10}\left(\frac{\sigma_X^2}{\sigma_U^2}\right)$) is shown in Fig. 4.29. Note that the values of $H(X_Q|Y)>2$ for values of the correlation SNR less than *3* and *10* dB for codes *3:2* and *4:2*

[8] The proof and the expression for the conditional mean are included in Appendix B.

respectively, as shown in Fig. 4.29. This implies that decoding is not possible for these SNR values irrespective of the side information.

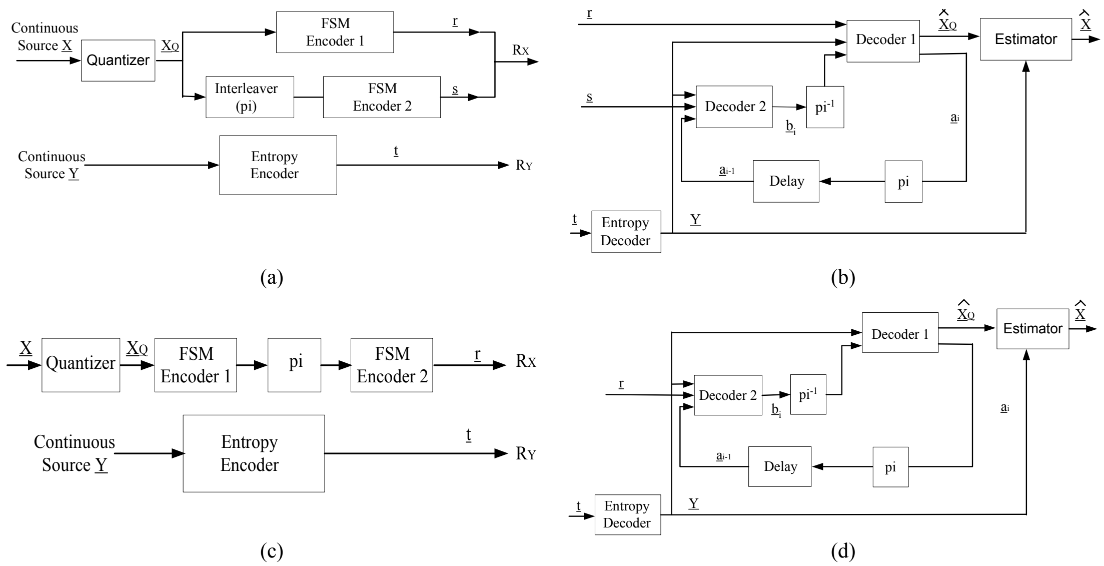


Fig. 4.28. Schematic block diagram for the Wyner-Ziv problem (a) parallel encoder+quantizer (b) parallel decoder+estimator (c) serial encoder+quantizer and (d) serial decoder+estimator

The results in Fig. 4.30 show the plot of the normalized distortion versus the correlation SNR. As can be seen, there is a difference between the optimum *R(d)* curve and the individual *8* level, *16* level and *32* level curves. An important point to note is that as the SNR gets high enough, the code with the higher rate performs better. This is logical as higher code rate means more quantizer levels and thus more resolution to represent the input. On the other hand, at low SNR values the higher code rate does not perform well. This could be attributed to the fact that for low SNR's the conditional entropy increases and therefore one needs a lower rate to encoder the source i.e. less compression is needed. In other words, taking the example of the *4*-to-*2* code, as the correlation SNR is decreased, correspondingly the conditional entropy $H(X_Q|Y)$ increases and approaches the rate which is *2*. This can be confirmed by observing Fig. 4.29. Thus, as the gap between the conditional entropy and the rate decreases, the performance also degrades.

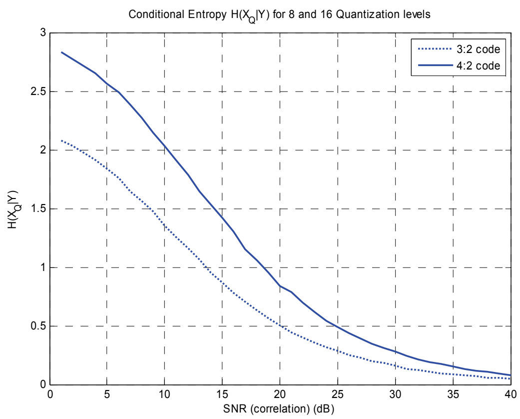


Fig. 4.29. Plot of Conditional Entropy $H(X_Q|Y)$ vs. correlation SNR (dB) for various quantizer levels (*3* and 4)

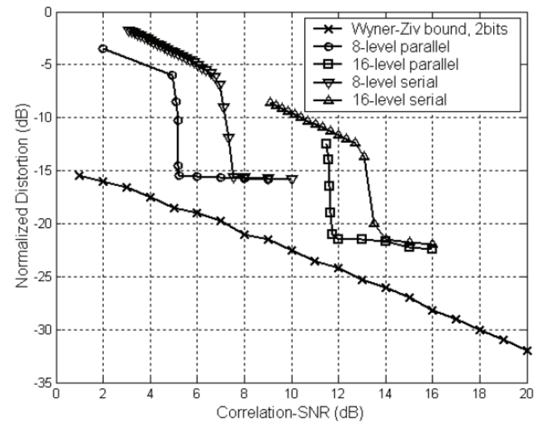


Fig. 4.30. Distortion versus correlation curves for the Rate *2* bits/source symbol and different quantizer resolutions along with the rate distortion bound ([22], [49])

It is also interesting to note the points where these codes have the best possible performance. Notice that after some point (high SNR values), the distortion does not seem to decrease. Thus, e.g. for the rate *3*-to-*2* code, the best point is just when this curve gives the lowest distortion which is *5* dB and *7.5* dB for the parallel and serial schemes respectively. Similarly, for the *4*-to-*2* code, the best point is *12* dB and *14* dB for the parallel and serial concatenated scheme respectively.

4.3.2 Encoder/Decoder Structure

The rates of the compressive codes used are *3*-to-*2* and *4*-to-*2*. The notation *x:y* means that *x* input bits give *y* output bits. Thus, *x* gives the resolution of the quantizer where 2^x is the number of quantization levels. Also note that the quantizer used in Fig. 4.28 is a Lloyd Max quantizer. The output is always *2* bits per input sample and the rate distortion bound in Fig. 4.30 for Wyner-Ziv coding is plotted using $R=2$ bits/sample.

The sub-section will focus on the performance analysis of a practical Wyner-Ziv coding scheme. This is quite similar to what was seen in Section 4.1.2, with the exception that now the side information is continuous instead of discrete but X_Q is finite. In other words, at the decoder, $P(X_Q = x_Q \mid Y = y)$ is fed as the side information to aid the decoder in the decoding process. This is needed as the conditional entropy $H(X_Q|Y)$ is less than $H(X_Q)$. Therefore, the focus will be on the encoder/decoder model shown in Fig. 4.31.

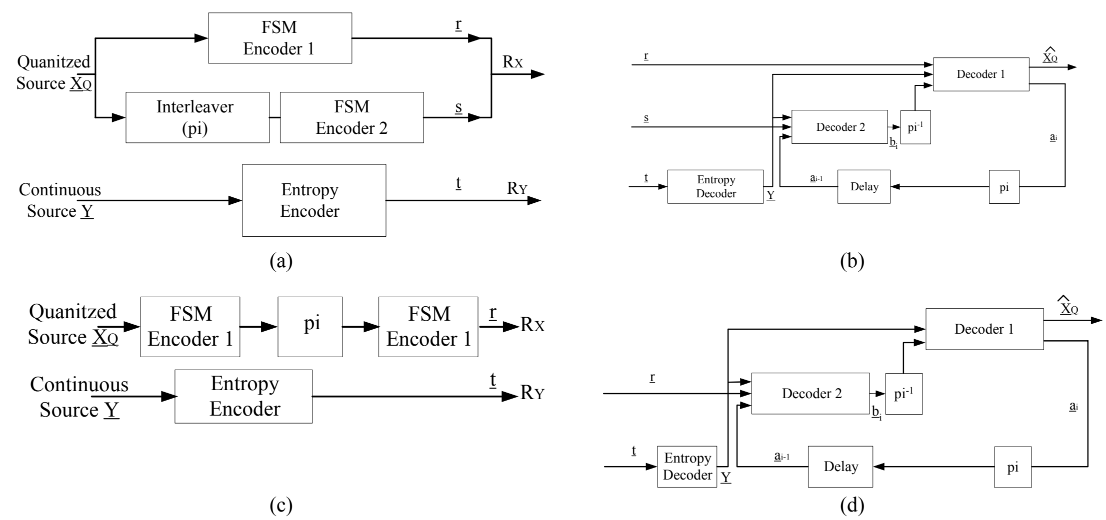


Fig. 4.31 Schematic block diagram for the extended Slepian-Wolf turbo compression scheme used in a practical Wyner-Ziv coding scheme for a (a) parallel encoder (b) parallel decoder (c) serial encoder and (d) serial decoder

4.3.3 EXIT Chart Construction and Numerical Results

The EXIT charts shown in Fig. 4.32 to Fig. 4.35 have been constructed using the system specifications described in the previous section and different pairs of correlated binary memoryless sources are used with distinct entropies. The procedure for the EXIT chart construction is the same as explained in Section 3.1 and 3.3 with the exception that the side information $P(X_Q)$ is replaced with $P(X_Q \mid Y)$. In construction of the EXIT curves that constitute the final EXIT chart, the standard deviation of the noise for the a-priori channel is varied over a range from *0.05* to *2.55*. Each of these simulations is performed for *500,000* source bits to obtain a numerical average from the mutual information quantities in Theorem 3.2 and resulting mutual information curves are plotted.

Initially, a *3*-to-*2* parallel compressive code is used to encode a quantized source X_Q as shown in Fig. 4.31. For this rate, Fig. 4.32 (a) and (b) show the EXIT charts with an open tunnel for conditional entropies $H(X_Q|Y)=1.40$ and *1.50* bits respectively. The correlation SNR for Fig. 4.32 (a) and (b) is *9.8* and *6.4* dB respectively. It can be observed that as long as the conditional entropy $H(X_Q|Y)<2$, the source can be decoded successfully whereas once $H(X_Q|Y)>2$ as shown in Fig. 4.32 (c), the EXIT tunnel is closed signifying that successful decoding of the quantized source is no longer possible.

In addition to the *3*-to-*2* parallel compressive turbo code, Fig. 4.33 presents the results for a rate *4*-to-*2* parallel compressive code. The same methodology has been used in these cases, as for the above mentioned *3*-to-*2* compressive turbo code. The EXIT charts in Fig. 4.33 a), Fig. 4.33 b) show successful convergence whereas the one in Fig. 4.33 c) demonstrate unsuccessful convergence due to the closed EXIT tunnel.

The same methodology is applied to serial concatenated schemes as shown in Fig. 4.34 and Fig. 4.35 for a code rate of *3*-to-*2* and *4*-to-*2* respectively. The EXIT charts in Fig. 4.34 a), b) and Fig. 4.35 a), b) show successful convergence whereas Fig. 4.34 c) and Fig. 4.35 c) demonstrate unsuccessful convergence due to the closed EXIT tunnel.

To verify the decoding trajectory of the actual iterative decoder against the predicted EXIT trajectory, the encoder/decoder system is simulated and the results are shown in Fig. 4.32 to Fig. 4.35. The actual iteration path (dashed line) is very close to the EXIT chart boundaries showing that indeed the EXIT chart technique is a good measure of the iterative decoding process and can be reliably used to study these codes.

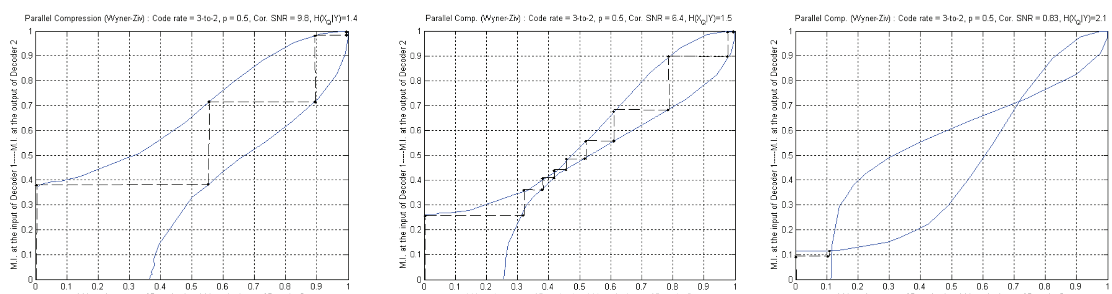


Fig. 4.32 EXIT charts for a rate *3*-to-*2* Slepian-Wolf parallel compressive turbo code including the actual decoding trajectory (dashed line): (left-to-right) a) $H(X_Q|Y) = 1.40$, *Correlation SNR=9.8* dB; b) $H(X_Q|Y)=1.50$, *Correlation SNR=6.4* dB; c) $H(X_Q|Y)=2.10$, *Correlation SNR=0.83* dB.

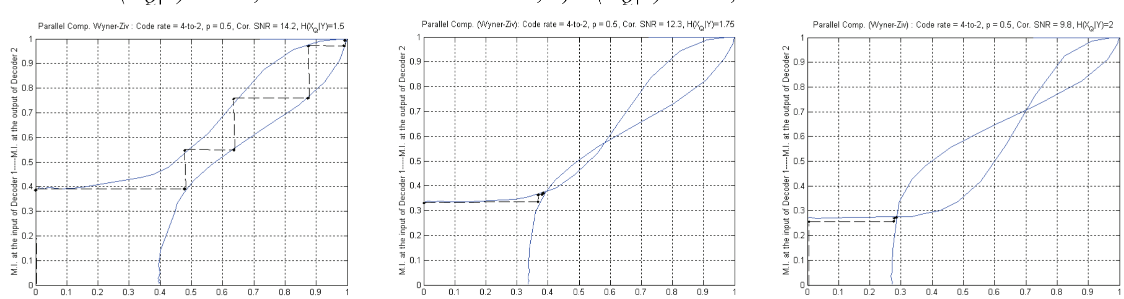


Fig. 4.33 EXIT chart for a rate *4*-to-*2* Slepian-Wolf parallel compressive turbo code including the actual decoding trajectory (dashed line): (left-to-right) a) $H(X_Q|Y) =1.50$, *Correlation SNR=14.2* dB; b) $H(X_Q|Y)=1.75$, *Correlation SNR=12.3* dB; c) $H(X_Q|Y) = 2.00$, *Correlation SNR=9.8* dB

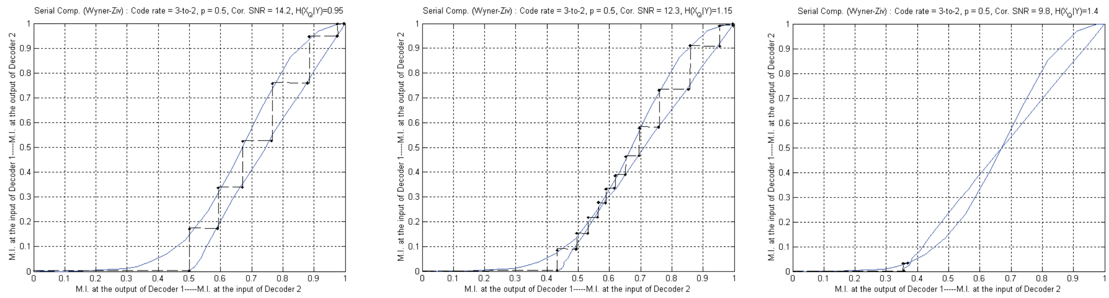


Fig. 4.34 EXIT charts for a rate *3*-to-*2* Slepian-Wolf serial compressive turbo code including the actual decoding trajectory (dashed line): (left-to-right) a) $H(X_Q|Y) = 0.95$, *Correlation SNR=14.2* dB; b) $H(X_Q|Y)=1.15$, *Correlation SNR=12.3* dB; c) $H(X_Q|Y)=1.40$, *Correlation SNR=9.8* dB

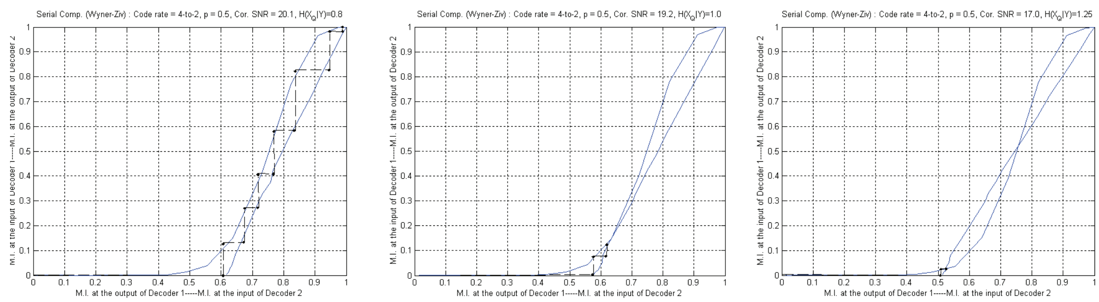


Fig. 4.35 EXIT charts for a rate 4-to-*2* Slepian-Wolf serial compressive turbo code including the actual decoding trajectory (dashed line): (left-to-right) a) $H(X_Q|Y) = 0.80$, *Correlation SNR=20.1* dB; b) $H(X_Q|Y)=1.00$, *Correlation SNR=19.2* dB; c) $H(X_Q|Y)=1.25$, *Correlation SNR=17.0* dB

4.4 Chapter Summary

Several numerical results have been presented in this chapter in order to illustrate the usefulness of the EXIT chart technique developed in Chapter 3. Firstly, this chapter looked at several turbo compression schemes of a single source assuming infinite and finite precision implementation of constituent decoders. Presented results showed a good match between the numerically evaluated EXIT chart results and the BER simulations. Consequently, the EXIT charts were extended to analyze the performance of Slepian-Wolf and Wyner-Ziv problems of source coding based on compressive turbo codes. EXIT charts were again shown to perform well in predicting the convergence behavior of the iterative decoders.

Chapter 5 Conclusion

5.1 Thesis Motivation and Objectives

The telecommunication industry around the globe is striving for new technologies as future wireless cellular systems are required to support increased user data rates and offer seamless user access to the backbone communication network [5], [51]. Powerful data compression techniques and error control codes are two of the most promising approaches that facilitate applications in the abovementioned systems and services [51], [53].

The main focus of this thesis was on the performance analysis of turbo compressive codes using EXIT charts, a method that was introduced by Stephan ten Brink in 2001 for turbo channel codes. The major advantages of the EXIT chart technique is the fact that they can be computed quickly, thus making it much more convenient than the traditionally simularted BER performance curves. Furthermore, the EXIT charts allow analyzing the constituentt decoders in an iterative decoder separately and have become an essential part of turbo code design.

5.2 Main Research Accomplishments

This thesis presented both analytical and simulation results, addresing extension of the EXIT chart technique from turbo channel codes to linear and non-linear turbo compressive codes. To make the thesis self-contained, Chapter 2 and the first part of Chapter 3 gave an introduction to the used coding and information theory tools as well as an overview of the EXIT chart technique used for turbo channel codes. Chapter 3 presented analytic results related to extension of EXIT charts to compressive turbo codes, while Chapter 4 used these results to obtain numerical results in various compressive scenarios (e.g., single source compression, Slepian-Wolf coding, Wyner-Ziv distributed source coding).

In Chapter 3, a formula was first derived for calculation of the mutual information used for the EXIT chart analysis of compressive turbo codes. The derived formula was then compared in detail to the one recently proposed by Hagenauer in [32] in order to illustrate that the derived formula improves significantly the performance analysis of compressive turbo codes. Consequently, this thesis showed that the original mutual information expression provided in [32] was not accurate, but can be shown to provide only a loose lower bound on the mutual information used in the construction of the EXIT charts. Finally, Chapter 3 extended the EXIT chart technique to performance analysis of analog iterative decoders, since such decoders offer several potential advantages (lower power consumption, faster computation. etc.) when compared to digitally implemented turbo decoding algorithms [44].

Chapter 4 of this thesis included numerical results illustrating the successful applications of the derived EXIT chart technique to turbo compressive codes under various scenarios. Firstly, the EXIT charts were used for performance analysis of several single source compression schemes based on both parallel and serial concatenated encoders. A good match between the simulated system performance and the performance predicted by the EXIT chart technique was observed. Consequently, the EXIT chart technique developed in this thesis was further extended to distributed source coding schemes which can be encountered in various network applications, such as sensor networks, distributed video surveillance, etc. The EXIT charts were used to analyze the performance of Slepian-Wolf and Wyner-Ziv coding schemes based on compressive turbo codes applied to biased correlated data sources. Several numerical EXIT chart results were compared to simulated performance of these coding schemes, illustrating proper functioning of the designed EXIT charts in these two settings. Finally, the designed EXIT charts were used to analyze the performance of turbo compressive systems with finite precision in decoders. Again, the designed EXIT charts were shown to work well in terms of their ability to predict simulated performance results.

5.3 Future Research Directions

There is ample room for future research in the research area explored by this thesis, i.e., EXIT chart performance analysis of iterative (turbo) decoding schemes. The following list contains a brief description of some possible future research directions:

- **EXIT analysis of compressive LDPC codes:** Following success of turbo codes in data compression applications, LDPC (Low Density Parity Check) codes [27] have been recently extended and applied to data compression coding [42]. Consequently, EXIT charts technique derived in this thesis and density evolution could be explored as a means to analyze performance and possibly design compressive LDPC codes.
- **BER estimation based on EXIT charts:** The original EXIT charts have been used to estimate BER performance of turbo channel codes and one can envision extending this methodology in order to utilize compressive EXIT charts to estimate BER of compressive turbo codes.
- **Design of constituent encoders using compressive EXIT charts:** The original EXIT charts have also been successfully used to design constituent encoders of capacity-achieving turbo channel codes [15]. Again, the compressive EXIT charts could be used to design entropy-achieving turbo compressive codes.
- **Further insights into analog iterative decoders:** EXIT charts, developed in Section 3.4, could also be used for both additional analysis and design of analog iterative decoders. In particular, one could use them to take into account various analog circuit specifics (e.g. the technology parameters, substrate process) and search for encoders that are more suitable to analog decoding. Furthermore, EXIT chart analysis for analog decoders can be extended to stochastic decoders.
- **EXIT charts through a fully closed form mutual information expression:** The key formulas, obtained in Section 3.3.3 (Theorem 3.2), require numerical evaluation (an average taken over many samples) in order to determine an EXIT curve of a constituent decoder. Achieving this objective via a closed form expression would be more accurate and less time-consuming, thus making the EXIT chart technique even more precise and practically applicable.

Appendices

Appendix A Information Theory Tools

Note that the following information theory tools and proofs have been included in order for this thesis to be self contained and the reader does not necessarily have to go through these.

A.1 Information Measure Proofs

Proof of Lemma 2.1:: The relative entropy between two probability mass functions $P_X(x)$ and $Q_X(x)$ with $x \in \mathcal{X}$ is defined as:

Let $Z = \{x : P_X(x) > 0\}$ be the support of $P_X(x)$. Then it follows:

$$\begin{aligned}
-D(P \| Q) &= -\sum_{x \in Z} P_X(x) \log_2 \left(\frac{P_X(x)}{Q_X(x)} \right) = \sum_{x \in Z} P_X(x) \log_2 \left(\frac{Q_X(x)}{P_X(x)} \right) \\
&\overset{(a)}{\leq} \log_2 \sum_{x \in Z} P_X(x) \frac{Q_X(x)}{P_X(x)} = \log_2 \sum_{x \in Z} Q_X(x) \\
&\leq \log_2 \sum_{x \in \mathcal{X}} Q_X(x) = \log_2 1 = 0 \\
&\therefore D(P \| Q) \geq 0
\end{aligned} \tag{A.1}$$

where (a) results from the Jensen's inequality which states that if *f* is a convex function and X is a random variable, then

$$E[f(X)] \geq f(E[X]) \tag{A.2}$$

□

Proof of Theorem 2.1: This theorem can be proved using Lemma 2.1 where $P_X(x)$ with $x \in \mathcal{X}$ and $Q_X(x = a_m) = \dfrac{1}{M}$, $m = 1, 2, \cdots, M$ for $\mathcal{X} = \{a_1, a_2, \cdots, a_M\}$.

$$
\begin{aligned}
& D(P \| Q) = \sum_{x \in \mathcal{X}} P_X(x) \log_2\left(\frac{P_X(x)}{Q_X(x)}\right) \geq 0 \\
& \sum_{x \in \mathcal{X}} P_X(x) \log_2\left(P_X(x)\right) - \sum_{x \in \mathcal{X}} P_X(x) \log_2\left(Q_X(x)\right) \geq 0 \\
& -H(X) - \sum_{x \in \mathcal{X}} P_X(x) \log_2\left(Q_X(x)\right) = -H(X) - \sum_{x \in \mathcal{X}} P_X(x) \log_2\left(\frac{1}{M}\right) \geq 0 \qquad \text{(A.3)} \\
& -H(X) - \log_2\left(\frac{1}{M}\right) \sum_{x \in \mathcal{X}} P_X(x) = -H(X) - \log_2\left(\frac{1}{M}\right) \geq 0 \\
& -H(X) \geq -\log_2(M) \qquad \rightarrow \qquad \therefore H(X) \leq \log_2(M)
\end{aligned}
$$

with equality if and only if $P_X(x) = Q_X(x)$ for all x i.e. $P_X(x = a_m) = p_m = \dfrac{1}{M}$ for $m = 1, 2, \cdots, M$. This completes the proof of Theorem 2.1.

□

Proof of Theorem 2.2: The proof of this theorem is as follows:

$$
\begin{aligned}
& H(X,Y) = E_{XY}\left(\log_2 \frac{1}{P_{XY}(x,y)}\right) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P_{XY}(x,y) \log_2(P_{XY}(x,y)) \\
& \mathit{Note}: P_{XY}(x,y) = P_{X|Y}(x \mid y) P_Y(y) \\
& H(X,Y) = -\sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{XY}(x,y) \log_2(P_{X|Y}(x \mid y)) - \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{XY}(x,y) \log_2(P_Y(y)) \\
& \qquad = -\sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{XY}(x,y) \log_2(P_{X|Y}(x \mid y)) - \sum_{y \in \mathcal{Y}} P_Y(y) \log_2(P_Y(y)) \\
& \qquad = E_{XY}\left(\log_2 \frac{1}{P_{X|Y}(x \mid y)}\right) + E_Y\left(\log_2 \frac{1}{P_Y(y)}\right) = H(X \mid Y) + H(Y)
\end{aligned}
$$

□

Proof of Theorem 2.4: Each part of this Theorem is proven separately as follows:

Proof of part (a):

Before proceeding with the proof, a random variable has to be defined:

$$Z_n^{(m)} = \begin{cases} 1, & \text{if } \ x_n = a_m \quad \text{for} \quad m = 1,2,\cdots,M \\ 0, & \text{else} \qquad\qquad \text{for} \quad n = 1,2,\cdots,N \end{cases}$$

$$\frac{1}{N} N\left(a_m \mid \underline{x}\right) = \frac{1}{N}\sum_{n=1}^{N} Z_n^{(m)} \quad \text{which is a R.V.}$$

$$\mathrm{E}\left\{Z_n^{(m)}\right\} = p_m, \qquad \mathrm{Var}\left\{Z_n^{(m)}\right\} = p_m - {p_m}^2 = p_m\left(1 - p_m\right)$$

$$\mathrm{E}\left\{\frac{1}{N} N\left(a_m \mid \underline{x}\right)\right\} = \frac{1}{N}\sum_{n=1}^{N} E\left\{Z_n^{(m)}\right\} = p_m$$

$$Var\left\{\frac{1}{N} N\left(a_m \mid \underline{x}\right)\right\} = \frac{1}{N}\mathrm{Var}\left\{Z_n^{(m)}\right\} = \frac{p_m\left(1 - p_m\right)}{N} \tag{A.4}$$

Using the Chebysev Inequality and letting $\delta>0$:

$$\Pr\left\{\left|\frac{1}{N} N\left(a_m \mid \underline{x}\right) - p_m\right| > \frac{\delta}{M}\right\} \le \frac{p_m\left(1 - p_m\right)}{N} . \frac{M^2}{\delta^2}$$

$$\therefore \Pr\left\{\left|\frac{1}{N} N\left(a_m \mid \underline{x}\right) - p_m\right| \le \frac{\delta}{M}\right\} \ge 1 - \frac{p_m\left(1 - p_m\right)M^2}{N\delta^2} \underset{\substack{N\to\infty \\ for\ any\ \delta>0}}{\longrightarrow} 1$$

Proof of part (b):

$$\frac{1}{N}\log_2 P_{\underline{X}}(\underline{x}) = \frac{1}{N}\sum_{n=1}^{N}\log_2 P_X(x_n) = \frac{1}{N}\sum_{m=1}^{M} N\left(a_m \mid \underline{x}\right)\log_2 p_m$$

Since $\underline{x}$ is a typical sequence we have that for $m = 1,2,\cdots,M$, and a given $\delta > 0$, there exists an N_0 such that for all $N \ge N_0$, the following holds:

$$
\begin{aligned}
& p_m - \frac{\delta}{M} \le \frac{1}{N} N\left(a_m \mid \underline{x}\right) \le p_m + \frac{\delta}{M} \\
& \therefore \sum_{m=1}^{M}\left(p_m + \frac{\delta}{M}\right)\log_2 p_m \le \frac{1}{N}\log_2 P_{\underline{X}}(\underline{x}) \le \sum_{m=1}^{M}\left(p_m - \frac{\delta}{M}\right)\log_2 p_m \\
& -H(X) + \frac{\delta}{M}\sum_{m=1}^{M}\log_2 p_m \le \frac{1}{N}\log_2 P_{\underline{X}}(\underline{x}) \le -H(X) - \frac{\delta}{M}\sum_{m=1}^{M}\log_2 p_m \\
& 2^{-N\{H(X)+\varepsilon\}} \le P_{\underline{X}}(\underline{x}) \le 2^{-N\{H(X)-\varepsilon\}}
\end{aligned}
\tag{A.5}
$$

where $\varepsilon = -\frac{\delta}{M}\sum_{m=1}^{M}\log_2 p_m > 0$ and is small. Note that in order to make the notation simpler, $\varepsilon = \delta$ can be chosen without any loss of generality. Thus,

$$
2^{-N\{H(X)+\delta\}} \le P_{\underline{X}}(\underline{x}) \le 2^{-N\{H(X)-\delta\}} \tag{A.6}
$$

Proof of part (c):

$$
\left(1-\delta\right)\cdot 2^{N\left(H(X)-\delta\right)} \le \left|T_{N}^{\underline{X}}(\delta)\right| \le 2^{N\left(H(X)+\delta\right)} \tag{A.7}
$$

To prove the right hand side of the above inequality for sufficiently large *N*:

$$
\begin{aligned}
1 &= \sum_{\underline{x}\in\mathcal{X}^N} P_{\underline{X}}(\underline{x}) \\
&\ge \sum_{\underline{x}\in T_{N}^{\underline{X}}(\delta)} P_{\underline{X}}(\underline{x}) \\
&\ge \sum_{\underline{x}\in T_{N}^{\underline{X}}(\delta)} 2^{-N\left(H(X)+\delta\right)} \\
&\ge 2^{-N\left(H(X)+\delta\right)}\left|T_{N}^{\underline{X}}(\delta)\right|
\end{aligned}
\tag{A.8}
$$

where the third inequality in (A.8) uses the result from the second part of Theorem 2.4. Based on (A.8), it follows that $\left|T_{N}^{\underline{X}}(\delta)\right| \le 2^{N\left(H(X)+\delta\right)}$.

In order to prove the left hand side of the inequality in (A.7), note that $\Pr\left[\underline{x}\in T_{N}^{\underline{X}}(\delta)\right] \underset{N\to\infty}{\longrightarrow} 1$ meaning that for large *N*, $\Pr\left[T_{N}^{\underline{X}}(\delta)\right] > \left(1-\delta\right)$ such that:

$$\begin{aligned}(1-\delta) &\le \Pr\left[T_{\underline{N}}^{X}(\delta)\right] \\ &\le \sum_{\underline{x}\in T_{\underline{N}}^{X}(\delta)} 2^{-N(H(X)-\delta)} \\ &= 2^{-N(H(X)-\delta)}\left|T_{\underline{N}}^{X}(\delta)\right|\end{aligned} \tag{A.9}$$

Again, the second inequality in (A.7) uses the inequality from Theorem 2.4. Consequently, we can conclude that $\left|T_{\underline{N}}^{X}(\delta)\right| \ge (1-\delta)\cdot 2^{N(H(X)-\delta)}$, thus completing the proof of the Theorem.

□

A.2 Proof of the SW Achievability Region

Before proceeding with the proof of the SW achievability region, it is imperative to give a few definitions.

Definition A.1: Let X and Y be discrete memoryless sources whose outputs are taken from the sets $\mathcal{X} = \{a_1, a_2, \cdots, a_{M_X}\}$ and $\mathcal{Y} = \{b_1, b_2, \cdots, b_{M_Y}\}$. Two sequences $\underline{x}$ and $\underline{y}$ of length N, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are called jointly typical sequence for a given $\delta > 0$ if

$$\left|\frac{1}{N} N\left[\left(a_{k,} b_m \mid \left(\underline{x}, \underline{y}\right)\right)\right] - p_{km}\right| \le \frac{\delta}{M_X M_Y}. \tag{A.10}$$

for all $k = 1, 2, \cdots, M_X$; $m = 1, 2, \cdots, M_Y$. For the given source $\underline{X}$ and $\underline{Y}$ and a given $\delta > 0$, let $T_{\underline{N}}^{XY}(\delta)$ denote the set of all the jointly-typical sequences of length N.

Theorem A.1: (*AEP: Asymptotic Equipartition Property for Jointly-Typical Sequences)*

For any two discrete memoryless stationary sources $\underline{X}$ and $\underline{Y}$ and for any $\delta > 0$, the following three results are true for the jointly-typical sequences:

a) Let the random sequence $\underline{X}$ and $\underline{Y}$ represent discrete memoryless sources $\underline{X} = X_1, X_2, \cdots, X_N$ and $\underline{Y} = Y_1, Y_2, \cdots, Y_N$ such that X_n's and Y_n's are independent identically distributed (i.i.d.) RV's with alphabet $\mathcal{X} = \{a_1, a_2, \cdots, a_{M_X}\}$ and $\mathcal{Y} = \{b_1, b_2, \cdots, b_{M_Y}\}$ respectively. They for any $\delta > 0$,

$$\Pr\left[\left(\underline{x}, \underline{y}\right) \in T_N^{\underline{X}\underline{Y}}(\delta)\right] \underset{N\to\infty}{\to} 1 \tag{A.11}$$

b) If $\underline{x}$, $\underline{y}$ be jointly typical sequences of a discrete stationary memoryless sources $\underline{x}$ and $\underline{y}$, then

$$2^{-N\{H(X,Y)+\delta\}} \le P_{\underline{X}\underline{Y}}(\underline{x}, \underline{y}) \le 2^{-N\{H(X,Y)-\delta\}} \tag{A.12}$$

where $\delta > 0$ (small). In other words, $P_{\underline{X}\underline{Y}}(\underline{x}, \underline{y}) \approx 2^{-NH(X,Y)}$.

c) $(1-\delta) \cdot 2^{N(H(X,Y)-\delta)} \le \left|T_N^{\underline{X}\underline{Y}}(\delta)\right| \le 2^{N(H(X,Y)+\delta)}$

where $\left|T_N^{\underline{X}\underline{Y}}(\delta)\right|$ is the number of elements in the typical set $T_N^{\underline{X}\underline{Y}}(\delta)$.

Proof of Theorem: Each part of this Theorem is proven separately as follows and is based on the proof of Theorem 2.4:

Proof of part (a): Consider a new discrete memoryless source $Z = (X, Y)$ taking values on discrete alphabet $\mathcal{Z} = \{(a_1, b_1), (a_1, b_2), \cdots, (a_{M_X}, b_{M_Y})\}$ with probabilities $P_Z\left(z = (a_i, b_j)\right) = P_{XY}\left(x = a_i, y = b_j\right)$. Consequently, a random vector can be defined as $\underline{Z} = Z_1, Z_2, \cdots, Z_N$ where Z_n's are independent identically distributed (i.i.d.) RV's with alphabet $\mathcal{Z} = \{(a_1, b_1), (a_1, b_2), \cdots, (a_{M_X}, b_{M_Y})\}$. Thus, using Theorem 2.4 for this length-N source implies that Theorem A.1 is also true.

Proof of part (b) and (c): The proof of part (b) and (c) of this Theorem follows from using Theorem 2.4 for the new source $\underline{Z}$ defined above.

First of all, it will be shown that any rate pair outside of the rate region (2.28) cannot be reliably achieved. Consider the pair of sources (X,Y) as a compound discrete memoryless source. Then using Shannon's noiseless source coding theorem (mentioned in Theorem 2.5), the rate at which this compound source has to be encoded has to satisfy $R_X + R_Y \geq H(X,Y)$ to achieve vanishingly small probability of error. Next, assume that Y is known both at the encoder and decoder. From Shannon's noiseless source coding theorem, the rate in this case for encoding the source X must satisfy $R_X' \geq H(X\,|\,Y)$. Now, if Y is not known at the encoder, then the rate of encoding X has to satisfy $R_X \geq R_X' \geq H(X\,|\,Y)$, resulting in $R_X \geq H(X\,|\,Y)$. Due to the symmetry of the problem, it is also true that $R_Y \geq H(Y\,|\,X)$.

The second part of the proof has been adapted from [Cover 1975] and is based on the random binning method.

Definition A.2: The method of *random binning* is described as follows. Consider length N sequences emitted by the source X. For each such sequence, randomly pick an index from the set $\left\{0,1,2,\cdots,2^{NR_X}-1\right\}$ using uniform probability distribution. The set of sequences that have the same index form a bin. The same procedure can be used for the source Y.

At the decoding end, in order to recover a source sequence from the received bin index, that particular bin is searched for a typical X^N. If there is only one such sequence, than decoding is successful, otherwise if there are more than one such typical sequences then an error is declared. On the other hand, if a non-typical sequence is sent, then in that case the decoder will always make an error but the probability of sending a non-typical sequence is arbitrarily small as shown in Theorem 2.4 (a).

Before proceeding with the proof of the Slepian-Wolf achievability region, the random binning concept explained in Definition A.2 has to be extended to two sources *X* and *Y*. The basic idea is to partition the set $\mathcal{X}^N$ into 2^{NR_X} bins and similarly divide $\mathcal{Y}^N$ into 2^{NR_Y} bins. The next step is to independently assign every $\underline{x} \in \mathcal{X}^N$ to one of 2^{NR_X} bins according to a uniform distribution on $\left\{0,1,2,\cdots,2^{NR_X}-1\right\}$. Likewise, every $\underline{y} \in \mathcal{Y}^N$ has to be assigned to one of 2^{NR_Y} bins according to a uniform distribution on $\left\{0,1,2,\cdots,2^{NR_Y}-1\right\}$. For the encoding, source *X* and *Y* send the index of the bin in which $\underline{x}$ and $\underline{y}$ lie. At the decoder, given the index pair (i_X, i_Y), the decoded sequence pair is declared to be $\left(\hat{\underline{x}}, \hat{\underline{y}}\right) = \left(\underline{x}, \underline{y}\right)$ if there is only one pair of sequences $\left(\underline{x}, \underline{y}\right)$ such that $g_X\left(\underline{x}\right) = i_X$ and $g_Y\left(\underline{y}\right) = i_Y$ and $\left(\underline{x}, \underline{y}\right) \in T_N^{\underline{X}\underline{Y}}(\delta)$. Otherwise, an error is declared.

Let the actual source sequences generated by the sources *X* and *Y* be $\left(\underline{x}', \underline{y}'\right)$. Then there are four error events that can occur at the decoder end. These errors are listed below:

- $E_1 = \left\{\left(\underline{x}', \underline{y}'\right) \notin T_N^{\underline{X}\underline{Y}}(\delta)\right\}$
- $E_2 = \left\{\exists \underline{x}'' \neq \underline{x}' : g_X\left(\underline{x}''\right) = g_X\left(\underline{x}'\right), \left(\underline{x}'', \underline{y}'\right) \in T_N^{\underline{X}\underline{Y}}(\delta)\right\}$
- $E_3 = \left\{\exists \underline{y}'' \neq \underline{y}' : g_Y\left(\underline{y}''\right) = g_Y\left(\underline{y}'\right), \left(\underline{x}', \underline{y}''\right) \in T_N^{\underline{X}\underline{Y}}(\delta)\right\}$
- $E_4 = \left\{\exists\left(\underline{x}'', \underline{y}''\right) : \underline{x}'' \neq \underline{x}', \underline{y}'' \neq \underline{y}', g_X\left(\underline{y}''\right) = g_X\left(\underline{y}'\right), g_Y\left(\underline{y}''\right) = g_Y\left(\underline{y}'\right), \left(\underline{x}'', \underline{y}''\right) \in T_N^{\underline{X}\underline{Y}}(\delta)\right\}$

The total probability of error is as follows:

$$P_e^{(N)} = P\left(E_1 \cup E_2 \cup E_3 \cup E_4\right) \leq P\left(E_1\right) + P\left(E_2\right) + P\left(E_3\right) + P\left(E_4\right) \quad \text{(A.13)}$$

Using Theorem 2.4, it can be deduced that $P(E_1)\rightarrow 0$ as $n\rightarrow\infty$. $P(E_2)$ can be written as follows:

$$\begin{aligned}
P(E_2) &= \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) \bigcup_{(\underline{x}'',\underline{y}')\in T_N^{\underline{X}\,\underline{Y}}(\delta):\underline{x}''\neq\underline{x}'} P\{g_X(\underline{x}'')=g_X(\underline{x}')\} \\
&\le \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) \sum_{(\underline{x}'',\underline{y}')\in T_N^{\underline{X}\,\underline{Y}}(\delta):\underline{x}''\neq\underline{x}'} P\{g_X(\underline{x}'')=g_X(\underline{x}')\} \\
&\le \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) \sum_{(\underline{x}'',\underline{y}')\in T_N^{\underline{X}\,\underline{Y}}(\delta):\underline{x}''\neq\underline{x}'} \sum_{\forall i_X} P\{g_X(\underline{x}'')=g_X(\underline{x}')\mid g_X(\underline{x}')=i_X\} P\{g_X(\underline{x}')=i_X\} \\
&= \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) \sum_{(\underline{x}'',\underline{y}')\in T_N^{\underline{X}\,\underline{Y}}(\delta):\underline{x}''\neq\underline{x}'} 2^{-NR_X} \\
&\le 2^{-NR_X} \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) \left|T_N^{\underline{X}|\underline{y}}(\delta)\right| \\
&\le 2^{-NR_X} 2^{+N(H(X|Y)+\varepsilon)} \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) = 2^{-N(R_X-H(X|Y)-\varepsilon)} \\
&\le 2^{-N(R_X-H(X|Y)-\varepsilon)}
\end{aligned} \tag{A.14}$$

Thus, from (A.14) it can be seen that if $R_X > H(X\,|\,Y)$ with N sufficiently large, then $P(E_2)\rightarrow 0$. By a similar argument, if $R_Y > H(Y\,|\,X)$ then $P(E_3)\rightarrow 0$, with N sufficiently large.

Finally, using a similar argument for $P(E_4)$:

$$\begin{aligned}
P(E_4) &= \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) \bigcup_{\substack{(\underline{x}'',\underline{y}'')\in T_N^{\underline{X}\,\underline{Y}}(\delta): \\ \underline{x}''\neq\underline{x}' \text{ and } \underline{y}''\neq\underline{y}'}} P\{g_X(\underline{x}'')=g_X(\underline{x}'),\, g_Y(\underline{y}'')=g_Y(\underline{y}')\} \\
&\le \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) \sum_{\substack{(\underline{x}'',\underline{y}'')\in T_N^{\underline{X}\,\underline{Y}}(\delta): \\ \underline{x}''\neq\underline{x}' \text{ and } \underline{y}''\neq\underline{y}'}} P\{g_X(\underline{x}'')=g_X(\underline{x}'),\, g_Y(\underline{y}'')=g_Y(\underline{y}')\} \\
&\le 2^{-N(R_X+R_Y)} \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) \left|T_N^{\underline{X}\,\underline{Y}}(\delta)\right| \\
&\le 2^{-N(R_X+R_Y)} 2^{+N(H(X,Y)+\varepsilon)} \sum_{(\underline{x},\underline{y})} P_{\underline{X}\,\underline{Y}}(\underline{x},\underline{y}) = 2^{-N(R_X+R_Y-H(X,Y)-\varepsilon)} \\
&\le 2^{-N(R_X+R_Y-H(X,Y)-\varepsilon)}
\end{aligned} \tag{A.15}$$

Hence, (A.15) shows that for $R_X + R_Y > H(X,Y)$ and for N sufficiently large, $P\left(E_4\right) \to 0$. This completes the proof of the achievability region specified in Theorem 2.6 above.

A.3 Proof of $R_{X|Y}(d)$ for Jointly Gaussian Sources

From probability theory, it is known that for X and Y jointly Gaussian,

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho}}\exp\left(-\frac{1}{2(1-\rho)^2}\left(\frac{x^2}{\sigma_X{}^2} - \frac{2\rho xy}{\sigma_X\sigma_Y} + \frac{y^2}{\sigma_y{}^2}\right)\right)$$

$$\rho = \frac{E[XY]}{\sigma_X\sigma_Y} = \frac{E[X\beta(X+U)]}{\sigma_X\sigma_Y} = \frac{\beta E[X(X+U)]}{\sigma_X\sigma_Y} = \frac{\beta E[X^2 + XU]}{\sigma_X\sigma_Y} = \frac{\beta E[X^2] + \beta E[XU]}{\sigma_X\sigma_Y}$$

$$\rho = \frac{\beta E[X^2] + \beta \cancel{E[X]}\,\cancel{E[U]}}{\sigma_X\sigma_Y} = \frac{\beta\sigma_X{}^{\cancel{2}}}{\cancel{\sigma_X}\sigma_Y} = \frac{\beta\sigma_X}{\sigma_Y}$$

$$\sigma_Y = E[YY] = E[(\beta(X+U))^2] = \beta^2 E[(X+U)^2] = \beta^2 E[X^2 + 2XU + U^2]$$

$$\sigma_Y = \beta^2 E[X^2 + 2XU + U^2] = \beta^2\left(E[X^2] + \cancel{E[2XU]} + E[U^2]\right) = \beta^2\left(E[X^2] + E[U^2]\right)$$

$$\sigma_Y = \beta^2\left(\sigma_X{}^2 + \sigma_U{}^2\right) \tag{A.16}$$

Now, using Bayes rule,

$$f_{X|Y}(x\,|\,y)=\frac{1}{2\pi\sigma_X\sqrt{1-\rho}}\exp\left(-\frac{1}{2\sigma_X{}^2(1-\rho)^2}\left(x-\frac{\rho\sigma_X y}{\sigma_Y}\right)^2\right)$$

i.e. $$E(X\,|\,Y=y)=\frac{\rho\sigma_X y}{\sigma_Y}=\frac{\frac{\beta\sigma_X}{\sigma_Y}\sigma_X y}{\sigma_Y}=\frac{\beta\sigma_X{}^2 y}{\sigma_Y{}^2}$$

$$E(X\,|\,Y=y)=\frac{\beta\sigma_X{}^2 y}{\sigma_Y{}^2}=\frac{\cancel{\beta}\sigma_X{}^2 y}{\beta^{\cancel{2}}\left(\sigma_X{}^2+\sigma_U{}^2\right)}=\frac{\sigma_X{}^2 y}{\beta\left(\sigma_X{}^2+\sigma_U{}^2\right)}$$

and $$Var(X\,|\,Y=y)=\sigma_X{}^2(1-\rho)^2=\sigma_X{}^2\left(1-\left(\frac{\beta\sigma_X}{\sigma_Y}\right)^2\right)=\sigma_X{}^2\left(1-\frac{\beta^2\sigma_X{}^2}{\sigma_Y{}^2}\right)$$

$$Var(X\,|\,Y=y)=\sigma_X{}^2\left(1-\frac{\beta^2\sigma_X{}^2}{\sigma_Y{}^2}\right)=\sigma_X{}^2\left(1-\frac{\cancel{\beta^2}\sigma_X{}^2}{\cancel{\beta^2}\left(\sigma_X{}^2+\sigma_U{}^2\right)}\right)$$

$$Var(X\,|\,Y=y)=\sigma_X{}^2\left(1-\frac{\sigma_X{}^2}{\left(\sigma_X{}^2+\sigma_U{}^2\right)}\right)=\sigma_X{}^2\left(\frac{\sigma_X{}^2+\sigma_U{}^2-\sigma_X{}^2}{\left(\sigma_X{}^2+\sigma_U{}^2\right)}\right)$$

$$Var(X\,|\,Y=y)=\sigma_X{}^2\left(\frac{\sigma_U{}^2}{\left(\sigma_X{}^2+\sigma_U{}^2\right)}\right)=\left(\frac{\sigma_U{}^2\sigma_X{}^2}{\sigma_X{}^2+\sigma_U{}^2}\right) \tag{A.17}$$

In this section the special case will be considered where $\mathcal{X}=\mathcal{Y}=\widehat{\mathcal{X}}=real$, $D(x,\hat{x})=(x-\hat{x})^2$ and X, Y are jointly Gaussian with zero mean and are related by the following (without any loss in generality):

$$Y=\beta(X+U) \tag{A.18}$$

This sub-section will prove that

$$R_{X|Y}(d)=\begin{cases}\frac{1}{2}\log\left(\frac{\sigma_X{}^2\sigma_U{}^2}{\left(\sigma_X{}^2+\sigma_U{}^2\right)d}\right), & \text{for}\quad 0<d<\frac{\sigma_X{}^2\sigma_U{}^2}{\sigma_X{}^2+\sigma_U{}^2}\\[2ex] 0, & \text{for}\quad d\geq\frac{\sigma_X{}^2\sigma_U{}^2}{\sigma_X{}^2+\sigma_U{}^2}\end{cases} \tag{A.19}$$

Essentially, what needs to be shown is that for arbitrary $\widehat{X}\in\mathcal{M}_0(d)$:

$$I(X;\widehat{X}\mid Y)\geq\frac{1}{2}\log\left(\frac{{\sigma_X}^2{\sigma_U}^2}{\left({\sigma_X}^2+{\sigma_U}^2\right)d}\right),\quad \text{for}\quad 0<d<\frac{{\sigma_X}^2{\sigma_U}^2}{{\sigma_X}^2+{\sigma_U}^2} \tag{A.20}$$

$$R_{X|Y}(d)=\min_{\widehat{X}\in\mathcal{M}_0(d):D\leq d} I(X;\widehat{X}\mid Y) \tag{A.21}$$

$$I(X;\widehat{X}\mid Y)=h(X\mid Y)-h(X\mid\widehat{X},Y)=\frac{1}{2}\log\left(2\pi e{\sigma_{X|Y}}^2\right)-h(X\mid\widehat{X},Y)$$

$$I(X;\widehat{X}\mid Y)=\frac{1}{2}\log\left(2\pi e{\sigma_{X|Y}}^2\right)-h(X\mid\widehat{X},Y)=\frac{1}{2}\log\left(2\pi e\frac{{\sigma_U}^2{\sigma_X}^2}{{\sigma_X}^2+{\sigma_U}^2}\right)-h(X\mid\widehat{X},Y) \tag{A.22}$$

$$I(X;\widehat{X}\mid Y)=\frac{1}{2}\log\left(2\pi eb{\sigma_X}^2\right)-h(X\mid\widehat{X},Y)$$

where $b=\dfrac{{\sigma_X}^2}{{\sigma_X}^2+{\sigma_U}^2}$. Letting $\widehat{X}\in\mathcal{M}_0(d)$, and $D(x,\hat{x})=\left(x-\hat{x}\right)^2$

$$h(X\mid\widehat{X},Y)=h(X-\widehat{X}\mid\widehat{X},Y)\leq h(X-\widehat{X})\leq\frac{1}{2}\log_2\left(2\pi ed\right) \tag{A.23}$$

Using (A.23) in (A.22):

$$I(X;\widehat{X}\mid Y)\geq\frac{1}{2}\log\left(2\pi eb{\sigma_X}^2\right)-\frac{1}{2}\log_2\left(2\pi ed\right)=\frac{1}{2}\log\left(\frac{b{\sigma_X}^2}{d}\right) \tag{A.24}$$

with equality if and only if $X\mid\widehat{X}$ is Gaussian with variance d. Consider the following:

$$X=\widehat{X}+Z \tag{A.25}$$

where Z is a zero-mean Gaussian R.V. with $E\left[Z^2\right]=E\left[\left(X-\widehat{X}\right)^2\right]=d$. Then:

$$P_{X|\widehat{X}}\left(x\mid\hat{x}\right)=\frac{1}{\sqrt{2\pi d}}e^{-\frac{1}{2d}\left(x-\hat{x}\right)^2} \tag{A.26}$$

With the distribution in (A.26), the result is

$$h(X\mid\widehat{X},Y)=h(X-\widehat{X})=\frac{1}{2}\log_2\left(2\pi ed\right) \tag{A.27}$$

with the lower bound on $I(X;\widehat{X}\mid Y)$ being achieved. The distortion constraint is also satisfied as $E\left[Z^2\right]=E\left[\left(X-\widehat{X}\right)^2\right]=d$. Thus:

$$R_{X|Y}(d)=\begin{cases}\dfrac{1}{2}\log\left(\dfrac{\sigma_X{}^2\sigma_U{}^2}{\left(\sigma_X{}^2+\sigma_U{}^2\right)d}\right), & \text{for} \quad 0<d<\dfrac{\sigma_X{}^2\sigma_U{}^2}{\sigma_X{}^2+\sigma_U{}^2}\\ 0, & \text{for} \quad d\geq\dfrac{\sigma_X{}^2\sigma_U{}^2}{\sigma_X{}^2+\sigma_U{}^2}\end{cases} \tag{A.28}$$

Another point worth mentioning here is that the limit $0<d<\dfrac{\sigma_X{}^2\sigma_U{}^2}{\sigma_X{}^2+\sigma_U{}^2}$ in (A.28) is the maximum distortion for which there is a finite rate possible. This is essentially $\sigma_{X|Y}{}^2$.

A.4 Formal Statement of *R*(d)*

Definition A.3: *A (N, M, Δ) distributed rate-distortion code* consists of an encoder mapping $g_X:\mathcal{X}^N\rightarrow \mathrm{I}_M=\{0,1,...,M-1\}$ and a decoder mapping $f:\mathcal{Y}^N\times\mathrm{I}_M\rightarrow\widehat{\mathcal{X}}^N$ with $\widehat{X}^N=f\left(Y^N,g_X\left(X^N\right)\right)$ such that

$$E_{X\widehat{X}}\left[D\left(x,\hat{x}\right)\right]=\Delta \tag{A.29}$$

where $P_{\widehat{X}|X,Y}\left(\hat{x}\mid x,y\right)$ is determined by the rate-distortion code and

$$E_{X\widehat{X}}\left[D\left(x,\hat{x}\right)\right]=\begin{cases}\displaystyle\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{\hat{x}\in\widehat{\mathcal{X}}}D\left(x,\hat{x}\right)P_{\widehat{X}|X,Y}\left(\hat{x}\mid x,y\right)P_{X,Y}\left(x,y\right) & \text{for discrete } X,Y\\ \displaystyle\int_x\int_y\sum_{\hat{x}\in\widehat{\mathcal{X}}}D\left(x,\hat{x}\right)P_{\widehat{X}|X,Y}\left(\hat{x}\mid x,y\right)f_{X,Y}\left(x,y\right) & \text{for continous } X,Y\end{cases}.$$

Definition A.4: For the Wyner-Ziv problem from Fig. 2.2, a rate-distortion pair *(R,d)* is achievable if for any $\varepsilon>0$ there exists a code (*N*, *M*, Δ) such that

$$M\leq 2^{N(R+\varepsilon)}, \quad \Delta\leq d+\varepsilon. \tag{A.30}$$

In this setting, $\mathcal{R}$ will denote the set of all achievable rate-distortion pairs *(R,d)*.

Definition A.5: $R^*(d)$ is the *rate distortion function* for the *Wyner-Ziv* case, and is the minimal rate that ensures a distortion not larger than *d*.

$$R^*(d) = \min_{(R,d)\in\mathcal{R}} R \tag{A.31}$$

Appendix B Detection and Estimation Concepts

B.1 Convexity of a Function

Let a function f be the map of a *subset* of n real numbers into a real number, i.e., $f:\Re^n \to \Re$ where $\Re \subseteq \mathbb{R}$. Then f is a convex function if the domain of f, $\Re^n$, is a convex set and if for all $\mathbf{x},\mathbf{y} \in \Re^n$, and α with $0 \le \alpha \le 1$,

$$f\left(\alpha\mathbf{x}+\left(1-\alpha\right)\mathbf{y}\right) \le \alpha f\left(\mathbf{x}\right)+\left(1-\alpha\right)f\left(\mathbf{y}\right). \tag{B.1}$$

That is, if a line is drawn in between the points ($\mathbf{x}$, $f(\mathbf{x})$) and ($\mathbf{y}$, $f(\mathbf{y})$) it always lies above the convex function as illustrated in figure 2.1.

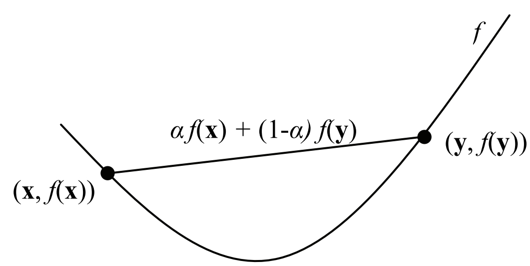


Fig B.1 Example of a convex function; any lines across two points of the function lie above the convex function

B.2 MMSE Estimator

To begin with it is important to find the estimator $\underline{\hat{a}}(\underline{r})$ which minimizes the mean square error. The proof is as follows:

$$E\left\{\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]^T\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]\right\} = \int E\left\{\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]^T\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right] \mid \underline{r}\right\} f_{\underline{R}}(\underline{r})d\underline{r}$$

As $f_{\underline{R}}(\underline{r})$ is not dependent on $\underline{a}$, the minimization of the above expression is equivalent to the minimization of:

$$E\left\{\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]^T\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right] \mid \underline{r}\right\} = \int\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]^T\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right] f_{\underline{A}|\underline{R}}(\underline{a} \mid \underline{r})d\underline{a}$$

To minimize the above expression, it can be differentiated w.r.t. the estimator and the result set to 0 to find the estimator which minimizes the mean square error:

$$\frac{dE\left\{\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]^T\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]\mid \underline{r}\right\}}{d\underline{\hat{a}}}=\frac{dE\left\{\int\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]^T\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]f_{\underline{A}|\underline{R}}(\underline{a}\mid\underline{r})d\underline{a}\right\}}{d\underline{\hat{a}}}=\underline{0}$$

$$\int 2\left[\underline{\hat{a}}(\underline{r})-\underline{a}\right]f_{\underline{A}|\underline{R}}(\underline{a}\mid\underline{r})d\underline{a}=\underline{0}$$

$$\cancel{2}\int\underline{\hat{a}}(\underline{r})f_{\underline{A}|\underline{R}}(\underline{a}\mid\underline{r})d\underline{a}=\cancel{2}\int\underline{a}f_{\underline{A}|\underline{R}}(\underline{a}\mid\underline{r})d\underline{a}$$

$$\underline{\hat{a}}(\underline{r})\int f_{\underline{A}|\underline{R}}(\underline{a}\mid\underline{r})d\underline{a}=\int\underline{a}f_{\underline{A}|\underline{R}}(\underline{a}\mid\underline{r})d\underline{a}$$

$$\underline{\hat{a}}(\underline{r})(1)=E\left[\underline{A}\mid\underline{r}\right]$$

$$\boxed{\underline{\hat{a}}(\underline{r})=E\left[\underline{A}\mid\underline{r}\right]}$$

B.3 Proof of the Conditional Mean in the Wyner-Ziv case

This conditional mean can be simplified as:

$$\begin{aligned}
\Pr\left(X=x\mid X_Q=x_Q,Y=y\right)&\overset{(a)}{=}\frac{\Pr\left(X=x,X_Q=x_Q,Y=y\right)}{\Pr\left(X_Q=x_Q,Y=y\right)}=\frac{\Pr\left(x,x_Q,y\right)}{\Pr\left(x_Q,y\right)}\\
&=\frac{\Pr\left(y\mid x,x_Q\right)\Pr\left(x_Q\mid x\right)\Pr\left(x\right)}{\Pr\left(y\mid x_Q\right)\Pr\left(x_Q\right)}\overset{(b)}{=}\frac{\Pr\left(y\mid x\right)\Pr\left(x_Q\mid x\right)\Pr\left(x\right)}{\Pr\left(y\mid x_Q\right)\Pr\left(x_Q\right)}\\
&=\frac{\left(\Pr\left(y\mid x\right)\Pr\left(x\right)\right)\Pr\left(x_Q\mid x\right)}{\Pr\left(y\mid x_Q\right)\Pr\left(x_Q\right)}\overset{(c)}{=}\frac{\Pr\left(x,y\right)\Pr\left(x_Q\mid x\right)}{\Pr\left(y\mid x_Q\right)\Pr\left(x_Q\right)}=\frac{\Pr\left(x,y\right)\Pr\left(x_Q\mid x\right)}{\Pr\left(y\mid x_Q\right)\Pr\left(x_Q\right)}\\
&\overset{(d)}{=}\frac{\Pr\left(x,y\right)\Pr\left(x_Q\mid x\right)}{\dfrac{\Pr\left(x_Q\mid y\right)\Pr\left(y\right)}{\cancel{\Pr\left(x_Q\right)}}\cancel{\Pr\left(x_Q\right)}}=\frac{\Pr\left(x,y\right)\Pr\left(x_Q\mid x\right)}{\Pr\left(x_Q\mid y\right)\Pr\left(y\right)}=\frac{\dfrac{\Pr\left(x,y\right)}{\Pr\left(y\right)}\Pr\left(x_Q\mid x\right)}{\Pr\left(x_Q\mid y\right)}\\
&\overset{(e)}{=}\frac{\Pr\left(x\mid y\right)\Pr\left(x_Q\mid x\right)}{\Pr\left(x_Q\mid y\right)}
\end{aligned}\tag{B.2}$$

where

$(a) \rightarrow$ Bayes Rule,

$(b) \rightarrow \Pr\left(y \mid x, x_Q\right) = \Pr\left(y \mid x\right)$ From the fact that x_Q is completely specified by x

$(c) \rightarrow \Pr\left(y \mid x\right)\Pr\left(x\right) = \Pr\left(x, y\right)$,

$(d) \rightarrow \Pr\left(y \mid x_Q\right) = \dfrac{\Pr\left(x_Q \mid y\right)\Pr\left(y\right)}{\Pr\left(x_Q\right)}$, and $(e) \rightarrow \dfrac{\Pr\left(x, y\right)}{\Pr\left(y\right)} = \Pr\left(x \mid y\right)$

Thus,

$$\begin{aligned}
E\left(X \mid X_Q = x_Q, Y = y\right) &= \int_{-\infty}^{\infty} x \Pr\left(X = x \mid X_Q = x_Q, Y = y\right) dx \\
&= \int_{-\infty}^{\infty} x \frac{p_{X|Y}\left(x \mid y\right)\Pr\left(x_Q \mid x\right)}{\Pr\left(x_Q \mid y\right)} dx \\
&\overset{(f)}{=} \frac{1}{p_{X_Q|Y}\left(x_Q \mid y\right)} \int_{-\infty}^{\infty} x p_{X|Y}\left(x \mid y\right) p_{X_Q|X}\left(x_Q \mid x\right) dx \qquad \text{(B.3)} \\
&= \frac{1}{p_{X_Q|Y}\left(x_Q \mid y\right)} \int_{-\infty}^{\infty} x p_{X|Y}\left(x \mid y\right) p_{X_Q|X}\left(x_Q \mid x\right) dx \\
&\overset{(g)}{=} \frac{1}{p_{X_Q|Y}\left(x_Q \mid y\right)} \int_{x \in I_{X_Q}} x p_{X|Y}\left(x \mid y\right) dx
\end{aligned}$$

where

$(f) \rightarrow p_{X_Q|Y}\left(x_Q \mid y\right)$ does not depend on x and $(g) \rightarrow p_{X_Q|X}\left(x_Q \mid x\right) = \begin{cases} 1, & \text{if } x \in I_{x_Q} \\ 0, & \text{if } x \notin I_{x_Q} \end{cases}$

At the decoder, $P(X_Q = x_Q \mid Y = y)$ is fed as the side information to aid the decoder in the decoding process. This side information can be expressed as:

$$P(X_Q = x_Q \mid Y = y) = \int_{x \in I_{x_Q}} p_{X|Y}\left(x \mid y\right) dx \qquad \text{(B.4)}$$

Appendix C Coding Tools

C.1 The BCJR Algorithm

Before going into the details of the BCJR algorithm, please note that the source code for the BCJR algorithm used in this thesis was a shareware algorithm used in the TSP lab of Prof. Jan Bajcsy at McGill University.

The numerical implementation of the BCJR algorithm in matlab is in the range of positive/negative $-2^{53}-1$ to $+2^{53}+1$. The smallest possible number representation in matlab is 2^{-1022} or about $2.2251*10^{-308}$. A zero in the BCJR algorithm, for the simulation in this thesis, is handled by setting that zero value close to 10^{-300} which is close to the lowest possible value that matlab can handle.

The basic concept used in decoding involves probabilities i.e. MAP probabilities or maximum a-posteriori probabilities. Thus at the receiver the message is chosen with the highest MAP probability. The system structure can be thought of as $\underline{r} = \underline{\mathrm{m}}+\underline{n}$. The idea is to maximize the probability of $\underline{m}$ in the codebook: $P(\underline{m} \mid \underline{r}) = \dfrac{P(\underline{r} \mid \underline{m})P(\underline{m})}{P(\underline{r})}$.

An optimal decoding algorithm based on this principle and used in many communication systems these days is known as Bahl, Cocke, Jelinek, and Raviv algorithm [6] - this algorithm is used to decode information iteratively and is based on a maximum a-posteriori probability (MAP) notion.

This algorithm is optimal for decoding. It basically maximizes the probability for each bit in the received sequence and outputs the message corresponding to the sequence in which the bit with the highest probability, of occurring at each bit position, is chosen.

The MAP algorithm makes a decision on the information bit b_k based on log-likelihood ratio (LLR), which is the logarithm of the ratio of the a-posteriori probability (APP) of each information bit b_k being 1 against it being 0. Let the state of the encoder at time k be S_k. The LLR of bit b_k is given by,

$$\Lambda(b_k) = \ln \frac{\sum_{S_k=0}^{2^M-1} \sum_{S_k=1}^{2^M-1} \gamma_1(R_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})\beta_k(S_k)}{\sum_{S_k=0}^{2^M-1} \sum_{S_k=1}^{2^M-1} \gamma_0(R_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})\beta_k(S_k)} \tag{C.1}$$

where:

$$\alpha(S_k) = \frac{\sum_{S_{k-1}=0}^{2^M-1} \sum_{i=0}^{1} \gamma_i(R_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})}{\sum_{S_k=0}^{2^M-1} \sum_{S_{k-1}=0}^{2^M-1} \sum_{i=0}^{1} \gamma_i(R_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})}, \qquad \alpha_0(S_0) = \begin{cases} 1 & \text{when } S_0 = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\beta(S_k) = \frac{\sum_{S_{k+1}=0}^{2^M-1} \sum_{i=0}^{1} \gamma_i(R_{k+1}, S_k, S_{k+1})\beta_{k+1}(S_{k+1})}{\sum_{S_k=0}^{2^M-1} \sum_{S_{k+1}=0}^{2^M-1} \sum_{i=0}^{1} \gamma_i(R_{k+1}, S_k, S_{k+1})\alpha_k(S_k)}, \qquad \beta_N(S_N) = \begin{cases} 1 & \text{when } S_N = 0 \\ 0 & \text{otherwise} \end{cases} \tag{C.2}$$

$$\gamma_i(R_k, S_{k-1}, S_k) = P(R_k / d_k = i, S_{k-1}, S_k) * P(d_k = i / S_{k-1}, S_k) * P(S_k / S_{k-1})$$

$\alpha(S_k)$ is the forward recursion probability which sums up the probability of all the possible paths from the start to state S_k.

$\beta(S_k)$ is the backward recursion probability which sums up probabilities of all possible paths from the end state to state S_k.

In the last expression in equation (C.2), $\gamma_i(R_k, S_{k-1}, S_k)$, the third term is the probability of going from state S_{k-1} to S_k i.e. the a-priori probability of the bit b_k. The second probability expression essentially checks if bit i is possible at the arc connecting state S_{k-1} to S_k. Lastly, the first term denotes the probability of receiving the bit R_k from the channel given that bit i was transmitted when the state changed from S_{k-1} to S_k.

Clearly, the BCJR is optimal as it takes into account all the paths passing through e.g. the k^{th} stage for the information bit *1* and then for *0*. In other words, to find the MAP probability for the k^{th} bit to be equal to 1, it takes into account all sequences (sums up the probability for all sequences) which have the k^{th} bit equal to *1*. Similarly, it sums up the probabilities for the k^{th} bit equal to *0* and then the log likelihood ratio is found after which the hard decision is made, whether the k^{th} bit was a *1* or a *0*.

The MAP algorithm is quite complex for implementation in a real system [59]. Thus, to avoid the number of operations as well as number representation issues, all the quantities are computed using the logarithm. The equivalent version of the MAP algorithm with the logarithms is as follows,

$$\Lambda(b_k) = \ln \frac{\sum_{S_k=0}^{2^M-1} \sum_{S_k=1}^{2^M-1} \gamma_1(R_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})\beta_k(S_k)}{\sum_{S_k=0}^{2^M-1} \sum_{S_k=1}^{2^M-1} \gamma_0(R_k, S_{k-1}, S_k)\alpha_{k-1}(S_{k-1})\beta_k(S_k)} \tag{C.3}$$

where:

$$\ln \alpha(S_k) = \ln\left(\sum_{S_{k-1}=0}^{2^M-1} \sum_{i=0}^{1} e^{\ln\left(\gamma_i(R_k, S_{k-1}, S_k)\right)+\ln\left(\alpha_{k-1}(S_{k-1})\right)}\right) - \ln\left(\sum_{S_k=0}^{2^M-1} \sum_{S_{k-1}=0}^{2^M-1} \sum_{i=0}^{1} e^{\ln\left(\gamma_i(R_k, S_{k-1}, S_k)\right)+\ln\left(\alpha_{k-1}(S_{k-1})\right)}\right)$$

$$\alpha_0(S_0) = \begin{cases} 0 & \text{when } S_0 = 0 \\ \alpha_{\min} & \text{otherwise} \end{cases}$$

$$\ln \beta(S_k) = \ln\left(\sum_{S_{k+1}=0}^{2^M-1} \sum_{i=0}^{1} e^{\ln\left(\gamma_i(R_{k+1}, S_k, S_{k+1})\right)+\ln\left(\beta_{k+1}(S_{k+1})\right)}\right) - \ln\left(\sum_{S_k=0}^{2^M-1} \sum_{S_{k+1}=0}^{2^M-1} \sum_{i=0}^{1} e^{\ln\left(\gamma_i(R_{k+1}, S_k, S_{k+1})\right)+\ln\left(\alpha_k(S_k)\right)}\right) \tag{C.4}$$

$$\beta_N(S_N) = \begin{cases} 0 & \text{when } S_N = 0 \\ \beta_{\min} & \text{otherwise} \end{cases}$$

$$\ln \gamma_i(R_k, S_{k-1}, S_k) = \ln\left(P(R_k / d_k = i, S_{k-1}, S_k)\right) + \ln\left(P(d_k = i / S_{k-1}, S_k)\right) + \ln\left(P(S_k / S_{k-1})\right)$$

C.2 Codes Used

Following are some of the matrices that have been used to construct the EXIT charts:

<u>Parallel Compression:</u>

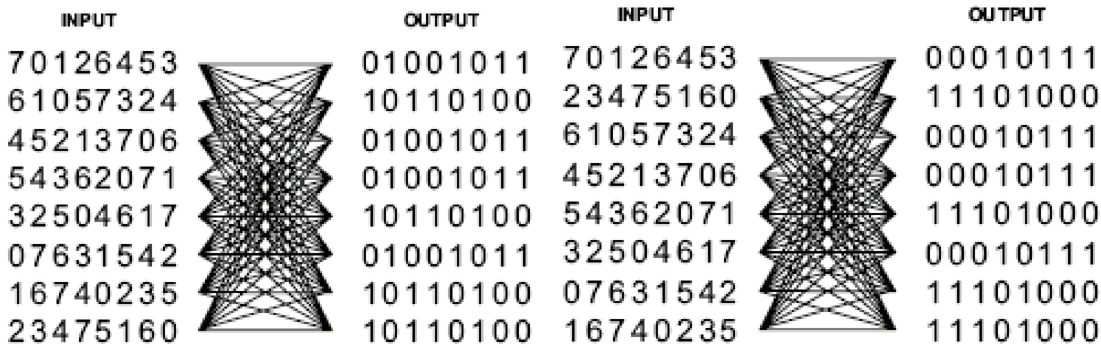
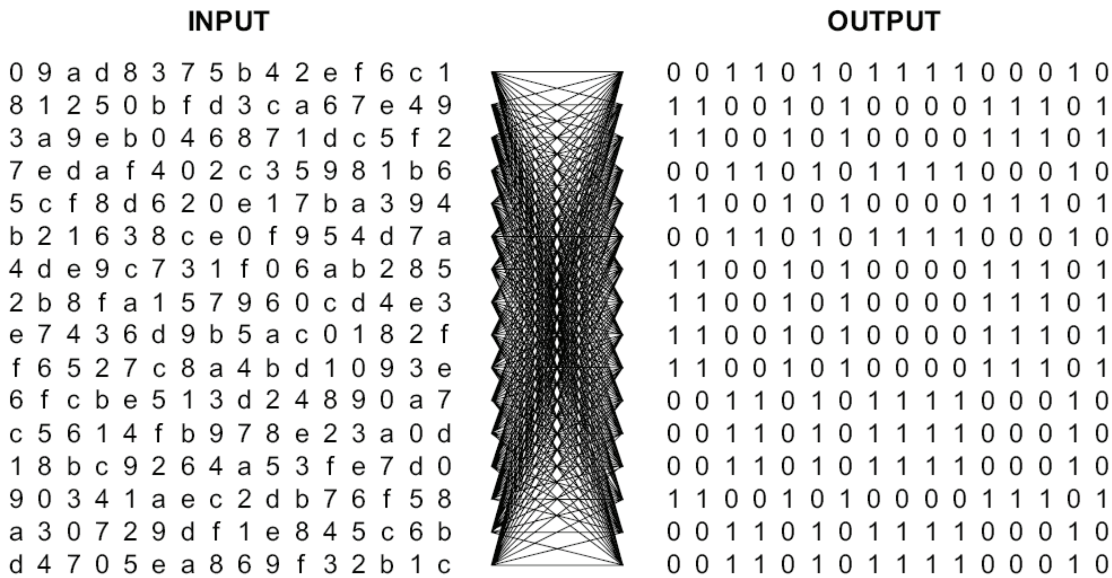


Fig. C.1. Trellises of two rate *3*-to-*1* compressive FSM encoders from ([46], [47]). (Input symbols are shown in base-*8*, while output symbols are binary.)



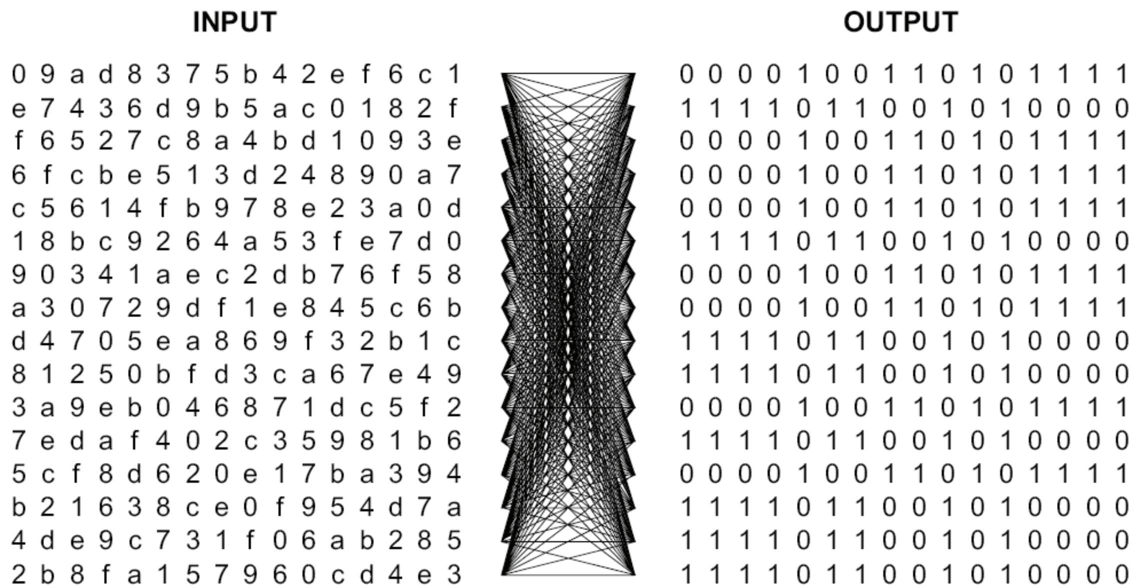


Fig. C.2. Trellises of two rate *4*-to-*1* compressive FSM encoders from ([46], [47]). (Input symbols are shown in base-*16*, while output symbols are binary.)

INPUT	OUTPUT
0dh3nfe6cp92v5rm74jsiko18algbtqu	00010010110011111000110111010100
qnbpdlksm3jo5v1ctu968e2rigfah704	11101101001100000111001000101011
ujft9hgoi7ns1r58pqd2ca6vmkbel340	00010010110011111000110111010100
d0seq23b1k4fi8mra9uhvplc57ot6gnj	00010010110011111000110111010100
hs0i6uvnt8ojeka7ml2d359gpr41qcbf	11101101001100000111001000101011
3ei0kcd5fqa1s6ol47gvhnr2b9mj8upt	11101101001100000111001000101011
nq6k0ophreul8ic1gj4b53fmvt27sad9	11101101001100000111001000101011
f2uco0193m6dgakp8bsjtrne75qv4ilh	11101101001100000111001000101011
e3vdp1082n7chblo9atisqmf64ru5jkg	11101101001100000111001000101011
6bn5h980avf4p3tg12lqkiu7ecjmdrso	00010010110011111000110111010100
c1tfr32a0l5ej9nqb8vguokd46ps7hmi	00010010110011111000110111010100
pk8qemnvl0gr6s2futa5bd1ohjc9i437	00010010110011111000110111010100
94oau67f5g0bmcivedqlrth813sp2kjn	11101101001100000111001000101011
2fj1ldc4erb0t7pk56hugmq3a8ni9vos	11101101001100000111001000101011
vies8ghpj6mt0q49orc3db7unlafk251	00010010110011111000110111010100
58k6iab39sc7q0uj21mpnht4dfgleovr	11101101001100000111001000101011
rmaockltn2ip4u0dsv879f3qjhebg615	11101101001100000111001000101011
mr7l1pogqfvk9jd0hi5a42enus36tbc8	11101101001100000111001000101011
7am4g891bue5o2sh03krljv6fdincqtp	00010010110011111000110111010100
49l7jba28td6r1vi30nomgs5cehkfpuq	11101101001100000111001000101011
ju2g4stlvaqhcm85kn0f17birp63oe9d	00010010110011111000110111010100
shdvbjiqg5lu3p7arof0e84tkm9cn162	11101101001100000111001000101011
iv3h5tskubrgdn94lm1e06ajqo72pf8c	00010010110011111000110111010100
kp5n3rqiodtmbhf2jg7860clsu14v9ea	00010010110011111000110111010100
ol9rfnmuk1hq7t3evsb4ac0pgld8j526	00010010110011111000110111010100
1cg2mef7do83u4qn65itjlp09bkhasrv	00010010110011111000110111010100
85pbv76e4h1andjufcrkqsg902to3lim	11101101001100000111001000101011
a7r9t54c6j38lfhsdepmouib20vq1ngk	00010010110011111000110111010100
lo4m2qrjpcsnage3ih6971dktv05u8fb	00010010110011111000110111010100
gt1j7vums9piflb6nk3c248hoq50rdae	11101101001100000111001000101011
b6q8s45d7i29kegtcfonpvja31ur0mhl	00010010110011111000110111010100
tgcuaijrh4kv2o6bqpe1f95sln8dm073	11101101001100000111001000101011

INPUT	OUTPUT
0dh3nfe6cp92v5rm74jsiko18algbtqu	00000100101100111110001101110101
f2uco0193m6dgakp8bsjtrne75qv4ilh	11111011010011000001110010001010
e3vdp1082n7chblo9atisqmf64ru5jkg	00000100101100111110001101110101
6bn5h980avf4p3tg12lqkiu7ecjmdrso	00000100101100111110001101110101
c1tfr32a0l5ej9nqb8vguokd46ps7hmi	00000100101100111110001101110101
pk8qemnvl0gr6s2futa5bd1ohjc9i437	11111011010011000001110010001010
94oau67f5g0bmcivedqlrth813sp2kjn	11111011010011000001110010001010
2fj1ldc4erb0t7pk56hugmq3a8ni9vos	00000100101100111110001101110101
vies8ghpj6mt0q49orc3db7unlafk251	11111011010011000001110010001010
58k6iab39sc7q0uj21mpnht4dfgleovr	11111011010011000001110010001010
rmaockltn2ip4u0dsv879f3qjhebg615	11111011010011000001110010001010
mr7l1pogqfvk9jd0hi5a42enus36tbc8	00000100101100111110001101110101
7am4g891bue5o2sh03krljv6fdincqtp	11111011010011000001110010001010
49l7jba28td6r1vi30nomgs5cehkfpuq	00000100101100111110001101110101
ju2g4stlvaqhcm85kn0f17birp63oe9d	11111011010011000001110010001010
shdvbjiqg5lu3p7arof0e84tkm9cn162	00000100101100111110001101110101
iv3h5tskubrgdn94lm1e06ajqo72pf8c	00000100101100111110001101110101
kp5n3rqiodtmbhf2jg7860clsu14v9ea	00000100101100111110001101110101
ol9rfnmuk1hq7t3evsb4ac0pgld8j526	00000100101100111110001101110101
1cg2mef7do83u4qn65itjlp09bkhasrv	11111011010011000001110010001010
85pbv76e4h1andjufcrkqsg902to3lim	00000100101100111110001101110101
a7r9t54c6j38lfhsdepmouib20vq1ngk	00000100101100111110001101110101
lo4m2qrjpcsnage3ih6971dktv05u8fb	11111011010011000001110010001010
gt1j7vums9piflb6nk3c248hoq50rdae	00000100101100111110001101110101
b6q8s45d7i29kegtcfonpvja31ur0mhl	11111011010011000001110010001010
tgcuaijrh4kv2o6bqpe1f95sln8dm073	11111011010011000001110010001010
qnbpdlksm3jo5v1ctu968e2rigfah704	00000100101100111110001101110101
ujft9hgoi7ns1r58pqd2ca6vmkbel340	00000100101100111110001101110101
d0seq23b1k4fi8mra9uhvplc57ot6gnj	11111011010011000001110010001010
hs0i6uvnt8ojeka7ml2d359gpr41qcbf	11111011010011000001110010001010
3ei0kcd5fqa1s6ol47gvhnr2b9mj8upt	11111011010011000001110010001010
nq6k0ophreul8ic1gj4b53fmvt27sad9	11111011010011000001110010001010

Fig. C.3. Trellises of two rate 4-to-1 compressive FSM encoders from [47]. (Input symbols are shown in base-32, while output symbols are binary.)

<u>Serial Compression:</u>

An example of serial compression for a code rate *3*-to-*2* is the following: The outer code was a *1*:*2* code (for every input bit, there are *2* output bits) and the inner code was a *3*:*1* code (for every *3* input bits, there is *1* output bit). Thus, the overall code rate is *3*-to-*2*.

Outer Code: [21]

Input trellis (in base *2*) = $\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$

Output trellis (in base *4*) = $\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$

Inner Code: E.g. to get a compression rate of 3-to-2, the inner code could be one of the 3-to-1 compressive codes given above.

IMPORTANT*: These matrices used the symbols 0 and 1 but for the construction of EXIT chart the 0's were replaced by -1's whereas the 1's were left alone.*

References

[1] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compr. Conf.*, pp. 252-261, Snowbird, UT, April 2002.

[2] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, November 2004.

[3] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: A model and two properties," in *Proc. Conf. Information Sciences and Systems*, pp. 742-747, Princeton, NJ, March 20–22, 2002.

[4] A. Ashikhmin, G. Kramer, and S. ten Brink, "Code rate and the area under extrinsic information transfer curves," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2002)*, p. 115, Lausanne, Switzerland, July 2002.

[5] AT&T, "Milestones in AT&T history," available at www.att.com/history/ milestones.html.

[6] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284-287, March 1974.

[7] J. Bajcsy, "Course notes ECSE 686B – Codes and Graphs," Department of Electrical and Computer Engineering, McGill University, Winter 2006.

[8] J. Bajcsy, C. V. Chong, J. A. Hunziker, D. A. Garr, and H. Kobayashi, "Iterative decoding in some existing systems," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 883 – 890, May 2001.

[9] S. Benedetto and G. Montorsi, "Unveiling turbo-codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 409 – 428, March 1996.

[10] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[11] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)," in *Proc. IEEE International Conference on Communication (ICC)*, pp. 1064-1070, Geneva, Switzerland, May 1993.

[12] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261 – 1271, October 1996.

[13] K. Bhattad and K. R. Narayanan, “An MSE Based Transfer Chart to Analyze Iterative Decoding Schemes,” *IEEE Transactions on Information Theory*, vol. 53, no.1, pp. 22-38, January 2007.

[14] S. ten Brink, “Design of Concatenated Coding Schemes based on Iterative Decoding Convergence,” Ph.D. dissertation, University of Stuttgart, Germany, 2001.

[15] S. ten Brink, “Convergence behaviour of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 40, pp. 1727-1737, October 2001.

[16] S. ten Brink, “A rate one-half code for approaching the Shannon limit by 0.1 dB,” *IEEE Electronic Letters*, vol. 36, no. 15, pp. 1293-1294, July 2000.

[17] S. Y. Chung, G. D. Forney, T.J. Richardson, and R. Urbanke, “On the design of low- density-parity-check codes within 0.0045 dB of the Shannon limit,” *IEEE Communication Letters*, vol. 5, no. 2, pp. 58-60, February 2001.

[18] M. H. M. Costa, “Writing on Dirty Paper”, *IEEE Transactions Information Theory*, IT-29, pp. 439-441, May 1983.

[19] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, New York: Wiley-Interscience, 1991.

[20] T. M. Cover, “A proof of data compression theorem of Slepian and Wolf for ergodic sources,” *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 226-228, March 1975.

[21] I. Deslauriers and J. Bajcsy, “Serial Turbo Coding for Data Compression and the Slepian-Wolf Problem”, in *Proc. of the IEEE Information Theory Workshop*, pp. 296-299, Paris, France, March 2003.

[22] I. Deslauriers and J. Bajcsy, “Coding for the Wyner-Ziv problem with Serial concatenated codes”, in *Proc. of 22nd Biennial Symposium on Communications*, Kingston, Ontario, Canada, May 2004.

[23] D. Divsalar, S. Dolinar, F. Pollara, and R. J. McEliece, “Transfer function bounds on the performance of turbo codes,” *TDA Progress Report 42–122*, pp. 44 – 55, August 1995.

[24] N. Dütsch, “Code optimization for lossless turbo source coding,” in *Proc. IST Mobile & Wireless Communications Summit*, paper 197, Dresden, Germany, June 2005.

[25] ETSI EN 301 790, V1.3.1: Digital video broadcasting, “Interaction channel for satellite distribution systems,” March 2003.

[26] G. D. Forney, *Concatenated Codes*, Cambridge, MA: MIT Press, 1966.

[27] R. G. Gallager, “Low density parity check codes,” no. 21 in Research Monograph Series, Cambridge, MA: MIT Press, 1963.

[28] R. G. Gallager, *Information Theory and Reliable Communication*, New York: Wiley, 1968.

[29] V. C. Gaudet and P. G. Gulak, “A 13.3-Mb/s 0.35-um CMOS analog Turbo decoder IC with a configurable interleaver,” *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 2010-2015, November 2003.

[30] J. Haganeuer and M. Winklhofer, “The Analog Decoder,” in *Proc. of IEEE Int. Symp. On Information Theory (ISIT 98)*, p. 145, Boston, USA, August 1998.

[31] J. Haganeuer, A. Barros, and A. Schaefer, “Lossless turbo source coding with decremental redundancy,” in *Proc. Fifth Int. ITG Conference on Source and Channel Codes*, pp. 333-339, Erlangen, Germany, January 2004.

[32] J. Hagenauer, “The EXIT Chart – Introduction to Extrinsic Information Transfer in Iterative Processing,” in *Proc. of 12th European Signal Processing Conference (EUSIPCO 2004)*, pp. 1541-1548, Vienna, Austria, September 2004.

[33] J. Haghighat, W. Hamouda, and M. Soleymani, “Design of lossless turbo source encoders,” *IEEE Signal Proc. Letters*, vol. 13, no. 8, pp. 453-456, August 2006.

[34] M. S. C. Ho, S. S. Pietrobon, and T. C. Giles. “Interleavers for punctured turbo codes,” in *Proc. IEEE Asia-Pacific Conf. on Commun.*, pp. 520 – 524, Singapore, November 1998.

[35] G. Hua and C. W. Chen, "Distributed Source Coding in Wireless Sensor Networks," in *Proc. of Second International Conf. on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE'05)*, p.6, Orlando, August 2005.

[36] W. C. Jakes, *Microwave Mobile Communications*, New York: Wiley, 1974.

[37] Y. Kim, B. J. Jeong, J. Chung, C. Hwang, J. S. Ryu, K. Kim, and Y. K. Kim, “Beyond 3G: Vision, requirements and enabling technologies,” *IEEE Commun. Mag.*, vol. 41, no. 3, pp. 120 – 124, March 2003.

[38] A. N. Kolmogorov and S. V. Fomin, *Elements of the Theory of Functions and Functional Analysis, Volume 2: Measure, The Lebesgue Integral, Hilbert Space*. Baltimore, MD: Graylock, 1961.

[39] L. -N. Lee, A. R. Hammons, F. -W. Sun, and M. Eroz, “Application and Standardization of Turbo Codes in Third Generation High Speed Wireless Data Services,” *IEEE Trans. Vehicular Technology*, vol. 49, no. 6, pp. 2198-2207, November 2000.

[40] S. Lee, *Spread Spectrum CDMA: IS-95 and IS-2000 for RF Communication*, New York: McGraw Hill, 2002.

[41] H. Leib, “Course Notes ECSE 620B – Information Theory and Coding,” Department of Electrical and Computer Engineering, McGill University, Winter 2008.

[42] A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Communication Letters*, vol. 6, no. 10, pp. 440-442, October 2002.

[43] A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Distributed compression of binary sources using convolutional parallel and serial concatenated convolutional codes,” in *Proc. of IEEE Data Compr. Conf.*, pp. 193-202, Snowbird, UT, March 2003.

[44] H. -A. Loelinger, F. Lustenberger, M. Helfenstein, and F. Tarkoy, “Probability propagation and decoding in analog VLSI,” *IEEE Trans. Information Theory*, pp. 837-843, February 2001.

[45] M. L. Miller, G. J. Dorr, and I. J. Cox, “Dirty Paper Trellis Codes for Watermarking,” in *Proc. of International Conference on Image Processing (ICIP 2002)*, vol.2, pp. 129-132, Rochester, NY, September 2002.

[46] P. Mitran and J. Bajcsy, “Turbo source coding: A noise-robust approach to data compression,” in *Proc. of IEEE Data Compr. Conf.*, p.465, Snowbird, UT, April 2002.

[47] P. Mitran, “Design and Applications of Turbo Source Codes,” M. Eng. Thesis, McGill University, Montreal, QC, Canada, 2002.

[48] P. Mitran and J. Bajcsy, “Near Shannon limit coding for the Slepian-Wolf problem,” in *Proc. 21st Queen's Biennial Symposium on Communications*, pp. 95-99, Kingston, ON, June 2002.

[49] P. Mitran and J. Bajcsy, “Coding for the Wyner-Ziv problem with Turbo-like codes,” in *Proc. of IEEE International Symposium on Information Theory (ISIT’02)*, p. 91, Lausanne, Switzerland, July 2002.

[50] G. Montorsi and S. Benedetto, “Design of fixed-point iterative decoders for concatenated codes with interleavers,” *IEEE J. Select. Areas Commun.*, vol. 19, pp. 871–882, May 2001.

[51] M. Mouly and M.-B. Pautet, *The GSM System for Mobile Communications*, Palaiseau, France: Telecom Publishing, 1992.

[52] Y. Oohama and T. S. Han, “Universal coding for the Slepian-Wolf data compression system and the strong converse theorem,” *IEEE Trans. Information Theory*, vol. 40, pp. 1908-1919, November 1994.

[53] J. Proakis, *Digital Communications*, New York: McGraw Hill, 1989.

[54] B. Riaz, "Performance Analysis of Turbo Compressive Codes," B. Eng. Thesis, McGill University, Montreal, QC, Canada, December 2006.

[55] B. Riaz and J. Bajcsy, "EXIT chart analysis for a single source turbo compression scheme," in *Proc. Conference on Information Sciences and Systems*, p. 730, Baltimore, MD, March 2007.

[56] B. Riaz and J. Bajcsy, "Analysis of Analog Iterative Decoders Using Generalized EXIT Charts", *Book of Abstracts of the Analog Decoding Workshop*, page 9, Montreal, QC, May 2007.

[57] B. Riaz and J. Bajcsy, "EXIT chart analysis of Slepian-Wolf Turbo Compression", in *Proc. Canadian Workshop on Information Theory*, pages 61-64, Edmonton, AB, June 2007.

[58] B. Riaz and J. Bajcsy, "Impact of Finite Precision Arithmetics on EXIT Chart Analysis of Turbo Codes", in *Proc. of IEEE Cons. Comm. and Net. Conf. (CCNC)*, pp. 205-206, Las Vegas, NV, January 2008.

[59] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun. (ICC '95)*, pp. 1009-1013, Seattle, WA, June 1995.

[60] C. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, pp. 379-423 and 623-656, July and October 1948.

[61] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Transactions on Information Theory*, pp. 471-480, July 1973.

[62] M. R. Soleymani, Y. Gao, and U. Vilaipornsawai, *Turbo Coding for Satellite and Wireless Communications*, Norwell, MA: Kluwer Academic Publishers, 2002.

[63] Third Generation Partnership Project (3GPP) Technical Specification Group, *Multiplexing and Channel Coding (FDD)*, TS 25.212 v6.5.0, June 2005.

[64] G. Titericz, R. D. Souza, J. Garcia-Frias, and G. I. Shamir, "Comparing Different Transmission Strategies Using Turbo Codes for Nonuniform Memoryless Sources," in *Proc. of IEEE Int. Conf. on Communications*, pp. 2722-2727, Glasgow, Scotland, June 2007.

[65] Z. Tu, J. Li, and R. S. Blum, "Compression of a binary source with side information using parallel concatenated convolutional codes," in *Proc. IEEE Global Commun. Conf.*, vol. 1, pp. 46-50, Dallas, TX, November 2004.

[66] M. Tüchler, "Convergence prediction for iterative decoding of threefold concatenated systems," in *Proc. IEEE Global Commun. Conf.*, pp. 1358 –1362, Taipei, Taiwan, November 2002.

[67] S. Verdú, *Multiuser Detection*, Cambridge, U.K.: Cambridge University Press, 1998.

[68] E. A. Vittoz, “Analog VLSI Signal Processing: Why, Where and How*?,” Journal of VLSI Signal Processing*, pp. 27-44, July 2004.

[69] R. Wooding, "The multivariate distribution of complex normal variables", *Biometrika*, vol. 43, pp. 212-215, June 1956.

[70] Y. Wu and B.D. Woerner, “The influence of quantization and fixed point arithmetic upon the BER performance of turbo codes”, in *Proc. of Vehicular Technology Conference*, vol. 2, pp. 1683-1687, Houston, TX, May 1999.

[71] A. D. Wyner, “On source coding with side information at the decoder,” *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 294–300, May 1975.

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

[73] A. D. Wyner, “The rate-distortion function for source coding with side information at the decoder—II: General sources,” *Information and Control*, vol. 38, issue 1, pp. 60–80, July 1978.

[74] Y. Zhao and J. Garcia-Frias, “Data compression of correlated non-binary sources using punctured turbo codes,” in *Proc. IEEE Data Compr. Conf.*, pp. 242-251, Snowbird, UT, April 2002.

[75] Y. Zhao and J. Garcia-Frias, “Joint estimation and compression of correlated nonbinary sources using punctured turbo codes,” *IEEE Trans. Communications*, vol. 53, no. 3, pp. 385-390, March 2005.

[76] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Trans. on Information Theory*, vol. 23, issue 3, pp. 337-343, May 1977.

[77] G.-C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, “Transmission of Non-Uniform Memoryless Sources via Non-Systematic Turbo Codes,” *IEEE Trans. On Communications*, vol. 52, no. 8, pp. 1344-1354, August 2004.