2021

BINARY LDPC DESIGN FOR SLEPIAN-WOLF CODING OF CORRELATED INFORMATION SOURCES

ELERUJA, SAEED ANIBABA

UNIVERSITY OF
PLYMOUTH

BINARY LDPC DESIGN FOR SLEPIAN-WOLF CODING OF CORRELATED INFORMATION SOURCES

by

SAEED ANIBABA ELERUJA

A thesis submitted to University of Plymouth

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Engineering, Computing and Mathematics

April 2021

Acknowledgements

This thesis is dedicated to my family.

I have been privileged to have Dr Marcel Adrian Ambrose as my director of studies. I am also indebted to Dr Mohammed Ahmed Zaki and Professor Martin Tomlinson for their counsel during my research.

This study was carried out with a grant of study fellowship from Ahmadu Bello University, Zaria and was sponsored by Emerging Markets Telecommunications Services (EMTS) Nigeria Limited (9Mobile Nigeria). My profound gratitude goes to Professor Muhammad Bashir Muazu, Professor Munzali Jibril, Ibrahim Dikko, Abdulrahman Ado, Oyetola Oduyemi, Rose Makinwa, Oluseyi Osunsedo, Babatunde Laniyan as well as the entire ABU and 9Mobile family.

The contributions of Dr Umar-Faruk Abdu-Aguye, Dr Muhammad Bashir Abdurrazaq, Dr Muhammad Dikko Almustapha, Dr Is-Haka Mkwawa and Dr. Ali Al-Nuaimi are well appreciated.

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

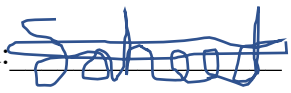
A programme of advanced study was undertaken, which included extensive reading of literature relevant to the research project; development of software for source code compression, analysis and performance simulations in the C++ programming language; writing conference papers; and attendance of international conferences on communications.

The author has presented papers in the following peer-reviewed international conferences:

1. Saeed Anibaba Eleruja, et al, "Design of Binary LDPC Codes for Slepian-Wolf Coding of Correlated Information Sources" *IEEE Global Conference on Signal and Information Processing (GlobalSIP), Hotel Bonaventure Montreal, Canada, November 14th – 16th, 2017.* https://doi.org/10.1109/GlobalSIP.2017.8309135

2. Saeed Anibaba Eleruja, et al, "Correction of a Single Error Bit of the (47, 24, 11) Quadratic Residue Code for Low-delay Distributed Source Coding using a Modified Dorsch Decoder" *The 2019 International Conference on Wireless Networks (ICWN 2019), Las Vegas, USA, 29thJuly– 1stAugust 2019.* https://csce.ucmss.com/cr/books/2019/LFS/CSREA2019/ICW7060.pdf

Word count of thesis: 26,623.

Signed: 

Saeed Anibaba Eleruja
Date: 27th April 2021

Binary LDPC Design for Slepian-Wolf Coding of Correlated Information Sources

Saeed Anibaba Eleruja

Abstract

Communication has shifted from simple point-to-point communication to network communication with many senders and/or receivers. These networks are distributed in nature and require the formation of low-delay and energy-limited communication schemes which make use of correlated sources in which common information from the sources is carefully managed to avoid duplication, thereby attaining the optimal usage of channel resources. Slepian and Wolf first proposed and developed a scheme where multiple information sources can be jointly compressed at a rate greater than the sum of their respective rates if compressed separately. The major challenges in existing schemes are three, namely, the modelling of a correlation channel; consideration for systems with stringent delay restrictions; and the tradeoff between error correction performance and associated complexity.

The aim of this research is to develop efficient means of improving the performance of distributed source coding, particularly, the Slepian-Wolf coding schemes. An Additive White Gaussian Noise (AWGN) equivalent channel is proposed and implemented. Subsequently, to justify the correlation between information sources, a binary symmetric equivalent channel is developed. It is understood from Shannon's theory that block codes of sufficiently large length are asymptotically optimal. This justifies the use of binary channel codes like Low-Density Parity-Check (LDPC) codes for practical Slepian-Wolf coding. Thus, LDPC codes of block lengths $n = 512$, 100 and 1024 were applied on the equivalent channel models and simulation results show improvement over other schemes with similar code lengths. Furthermore, to address the problem of systems with stringent restrictions on delay, new schemes were modelled to utilize very short, low delay codes like the [7, 4, 3] Hamming code, the [15, 7] BCH code as well as the [47, 24, 11] Quadratic Residue (QR) code. Particularly, an improved source coding scheme is proposed to correct a single-bit error of the [47, 24, 11] QR code based on the modification of the Dorsch decoder without the implementation of maximum-likelihood decoding. Finally, a framework is established for attaining maximum likelihood decoding and the associated complexity is presented.

Contents

List of Figures

Figure

List of Algorithms

Glossary

AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BER	Bit Error Rate
BCH	Bose-Chaudhuri-Hocquenghem
BP	Belief Propagation
BPSK	Binary Phase-Shift Keying
BP/SPA	Belief Propagation/Sum-Product Algorithm
BSC	Binary Symmetric Channel
CPEG	Cyclic PEG
CPU	Central Processing Unit
DE	Density Evolution
DSC	Distributed Source Coding
ECC	Error Correction Coding
FER	Frame Error Rate
GB	GigaBytes
GF	Galois Field
GNU	GNU is Not Unix!
IBP/SPA	Improved BP/SPA
IEEE	Institute of Electrical and Electronics Engineers
IPEG	Improved PEG
LDPC	Low-Density Parity-Check
MDS	Maximum-Distance Separable
MLD	Maximum-Likelihood Decoding
PEG	Progressive Edge-Growth
RAM	Random Access Memory
RS	Reed-Solomon
SNR	Signal-to-Noise Ratio

SNR(dB)	Signal-to-Noise Ratio in decibels
SPA	Sum-Product Algorithm
SPC	Single Parity-Check
SW	Slepian-Wolf
QR	Quadratic Residue

Chapter 1

Introduction

The origin of coding theory is in the problem of reliable communication over noisy channels which, in turn led to the Information theory discipline. The information theoretic problem prompted the definition of a mathematical structure called error-correcting code or simply code. Information theory and Coding theory originated with Claude Shannon's famous 1948 paper (Shannon, 1949). The channel coding theorem states roughly that good long codes are guaranteed to exist, without giving a clue how to construct them. Notwithstanding, Shannon's paper encapsulates five major concepts namely, entropy and information content, channel capacity and the noisy-channel coding theorem, formal architecture of communication systems, unification of all information media through digital representation of messages over communication channels, and source coding or data compression.

The information age has been revolutionized by Shannon's fundamental work in 1948. During the past decades, communication has shifted from simple point-to-point communication to network communication with many senders and/or receivers. The emergence of distributed systems has increased the requirements for the efficient and effective use of expensive and limited resources, such as bandwidth and power, hence energy efficiency is a major concern. Distributed Source Coding (DSC) refers to the compression of multiple statistically dependent sources that do not communicate with each other and therefore are encoded in a distributed manner.

Slepian and Wolf, in 1973 (Slepian & Wolf, 1973) laid the foundation of distributed source coding when they proved the counter-intuitive result that separate encoding with joint decoding achieves the same compression rate as joint encoding does. This could be achieved by partitioning all

possible source outcomes into bins indexed by syndromes of some good linear channel code for specific source correlation model (Zixiang, Liveris & Cheng, 2004). The Slepian-Wolf theorem deals with lossless compression of two correlated signals. In another pioneering work, Wyner and Ziv (Wyner & Ziv, 1976) studied the lossy counterpart of this problem which is a special case of distributed lossy source coding named lossy source coding with side information at the decoder.

The classical communication channel system involves a single source from which information is fed to an encoder. However, Slepian-Wolf coding involves two or more sources sending similar information to an encoder simultaneously. Common information from the sources is carefully managed to avoid unnecessary duplication and obtain optimal usage of channel capacity. Practical communication systems make use of correlated sources, for instance smart grid meters. Each smart grid meter for a particular grid conforms to certain protocols and this means that certain information in the header files will be the same for various meters. Common information from the sources is carefully managed to avoid unnecessary duplication and obtain optimal usage of channel capacity. Correlation between information sources can be modelled in various ways. According to Slepian and Wolf (Slepian & Wolf, 1973), multiple sources can be jointly compressed at a rate greater than the sum of their respective rates if compressed separately. Slepian-Wolf coding entails lossless coding of a source with the help of some side information available at the decoder only. Unlike traditional communication channel systems which have a single source from which information is fed to an encoder, Slepian-Wolf coding involves two or more sources sending similar, or correlated information to an encoder simultaneously.

1.1 Linear Codes: Basic Definitions and Notations

1.1.1 Linear Codes

A linear code is an error-correcting code for which any linear combination of codewords is also a codeword. Linear codes could be partitioned into block codes and convolutional codes. As the name implies, a block code maps a block of information bits onto a channel codeword such that there is no dependence on past information bits. In contrast, convolutional codes are highly structured forms of codes designed such that each output block depends not only on the current input block, but also on some of the past inputs as well. Turbo codes can be seen as a hybrid of these two types of codes. Linear codes are the most studied codes from a mathematical point of view because of their algebraic properties. An $[n, k, d]_q$ linear code C is a k-dimensional vector subspace of $\mathbb{F}_q^n$, where $\mathbb{F}_q^n$ is an n-dimensional vector space over a finite field of q elements $\mathbb{F}_q$. The codewords of linear codes are the k- dimensional vector subset of $\mathbb{F}_q^n$ which have length of n symbols. Individual codewords are denoted by $c = [c_0, c_1, \dots, c_k, \dots, c_{n-1}]$ and the maximum number of codewords in C is given by q^k. The minimum distance of the code is d.

1.1.2 Information Rate

Also known as the code rate, the information rate of a code R, is defined as the ratio of the number of information bits, k in each code block to the number of code bits, n.

$$R = \frac{k}{n} \tag{1.1}$$

1.1.3 Hamming Distance

The Hamming distance between two strings x and y of the same length over a finite alphabet, $\mathbb{F}_q^n$, denoted by $\Delta(x, y)$, is defined as the number of positions at which the two strings differ.

$$\Delta(x, y) = |\{i | x_i \neq y_i\}| \tag{1.2}$$

The fractional Hamming distance or relative distance between $x, y \in \mathbb{F}^n$ is given by

$$\delta(x, y) = \frac{\Delta(x,y)}{n} \tag{1.3}$$

1.1.4 Hamming Weight

The hamming weight, $wt(x)$ of a string, x over alphabet $\mathbb{F}_q^n$ is defined as the number of non-zero symbols in the string. Mathematically,

$$wt(x) = |\{i | x_i \neq 0\}| \tag{1.4}$$

This is also the numbers of '1's in a non-zero codeword or the Hamming distance between a codeword and the all-zero codeword.

It is worthy to note that $wt(x - y) = \Delta(x, y)$.

1.1.5 Minimum Distance

The minimum distance, or simply distance, of a code C, denoted $\Delta(C)$, is defined to be the minimum Hamming distance between two distinct codewords of C. Mathematically,

$$\Delta(C) = \min_{\substack{c_1, c_2 \in C \\ c_1 \neq c_2}} \Delta(c_1, c_2) \tag{1.5}$$

For every pair of distinct codewords in C, the Hamming distance between them is at least $\Delta(C)$. The relative distance of C, denoted $\delta(C)$, is the normalized quantity $\frac{\Delta(C)}{n}$, where n is the block length of C. Therefore, any two codewords of C differ in at least a fraction $\delta(C)$ of positions.

1.1.6 Euclidean Distance

Hamming distance is not always suitable for code design. In general, when a soft decision decoder is used at the receiver, Euclidean metric is used as design criteria and coding schemes are chosen that maximize the minimum Euclidean distance. More so, the Euclidean distance dominates the error bound in Gaussian channels with high signal-to-noise ratios.

The Euclidean distance between two strings x and y of the same length over a finite alphabet, $\mathbb{F}_q^n$, denoted by $\Delta_E(x, y)$.

$$\Delta_E(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1.6}$$

1.1.7 Parity-Check Matrix and Parity-Check Equation

The parity-check matrix H of a code C is an $(n - K) \times n$ matrix that contains $n \times k$ linearly independent vectors of $\mathbb{F}_q^n$ so that $cH^T = 0$ for all codewords $c \in C$. In other words, $C$is a null space of H and thus, C is an $[n, k, d]_q$ code if and only if there is a matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ of full row rank such that,

$$C = \{c \in \mathbb{F}_q^n \mid Hc = 0\} \tag{1.7}$$

Every row in the parity-check matrix H represents a parity-check equation. The parity-check matrix may also be expressed in a reduced row echelon form so that the last $(n-k)$ columns of H form an $(n-k) \times (n-k)$ identity matrix denoted I_{n-k}.

1.1.8 Regular and Irregular Codes

Codes with parity-check matrices that have a fixed number of non-zero symbols in each row as well as a fixed number of non-zero symbols in each column are called regular codes. Irregular codes do not have any fixed number of non-zero symbols.

1.1.9 Generator Matrix

Let $C \subseteq \mathbb{F}_q^n$ be a linear code of dimension k. A matrix $G \in \mathbb{F}_q^{n \times k}$ is said to be a generator matrix of C if its k columns span C. The generator matrix G provides a way to encode a message $x \in \mathbb{F}_q^n$ as the codeword $Gx \in C \subseteq \mathbb{F}_q^n$. As in the case of the parity-check matrix, H, the generator matrix, G may be expressed in a reduced row echelon form by elementary row operations. In that case, the first k coordinates of would be the identity matrix, I_k. A linear code has an encoding map $E: \mathbb{F}_q^k \to \mathbb{F}_q^n$ which is a linear transformation $x \to Gx$.

1.1.10 Syndrome

Suppose vector $y \in \mathbb{F}_q^n$ is some arbitrary vector, the syndrome of y is a vector $s \in \mathbb{F}_q^{n-k}$

defined by,

$$s = yH^T \tag{1.8}$$

In the case where y is actually a valid codeword, i.e., $y \in C$, then $s = 0$. Otherwise, at least one coordinate of s would have a non-zero value and $s \neq 0$.

1.2 Background to Distributed Source Coding

Slepian and Wolf (Slepian & Wolf, 1973) in the 1973 pioneer paper presented sixteen cases depending upon the information available to the encoders and decoders through varying configurations of four switches. The first, case-1111 in which all four switches are closed contains nothing new and is obtained by regarding the pair (X, Y) as a new discreet random variable. The most interesting and novel of case of Slepian-Wolf's paper, however, is the case-0011 where each encoder sees only its source. This setup is called asymmetric Slepian-Wolf (SW) coding. The results from these cases were presented as an admissible rate region in the R_X-R_Y plane and they generalize a similar and well-known result for a single information sequence, namely $R_X \geq H(X)$ for faithful reproduction. This paper also raised a lot of research questions. These include:

- How the foregoing extends to N correlated sources instead of two.
- How the foregoing extends to a rate-distortion theory of correlated sources.
- How to design block length codes given n to have small error probability for correlated sources.
- The theory of variable-length encodings for correlated sources.
- And how the theory extends for correlated sources that are not independent

drawings of pairs of correlated variables.

It is well known that the infimum of achievable rates is given by H(X/Y). This is the conditional entropy of X knowing Y and several practical coding schemes have been proposed (Coleman, Medard & Effros, 2005), (Zixiang, Liveris & Cheng, 2004). A good number of these works is based on channel codes (Coleman *et al.*, 2004), (Stankovic *et al.*, 2006), and particularly Low-Density Parity-Check (LDPC) codes (Cui *et al.*, 2011), (Liveris, Xiong & Georghiades, 2002), (Matsuta, Uyematsu & Matsumoto, 2010). Several researchers have worked on channel coding for the design of good LDPC codes. Specifically, (Richardson, Shokrollahi & Urbanke, 2001), (Richardson & Urbanke, 2001) show that the performance of a code depends on its degree distributions. The degree distribution optimization can be obtained from density evolution which constitutes a good starting point for practical code design, but the issue of constructing proper coding matrix at finite length remain (Dupraz, Savin & Kieffer, 2015).

Ancheta (Ancheta, 1976) proposed the syndrome-source-coding method of using error-correcting codes to obtain data compression in which the source sequence is treated as an error pattern whose syndrome forms the compressed data. He formulated a "universal" generalization of syndrome-source-coding which provided robustly effective distortion-less coding of source ensembles. The fundamental principle is that the source output is treated as a channel error pattern **e**, rather than as the received channel codeword. The syndrome **s**, which is the pattern of parity-check failures, is taken as the compressed data. At the user end, an "error-pattern estimator" for the code, which produces a likely error pattern **ê** consistent with **s**, is used as the source decoder. Ancheta showed that, for a

memoryless binary source and for arbitrarily small distortion, the required number of transmitted digits per source letter can be made arbitrarily close to the source entropy.

Theoretically,

$r = x + e$

where **r** is received word,

x is transmitted word,

e is the source output which is assumed to be statistically independent of **x** and where the addition is component-by-component in Galois Field (GF) (2).

The encoder (or compressor, syndrome-former) computes **s** as,

$s = rH^T$

where **H** is the parity-check matrix of the code, so that:

$s = rH^T = (x + e)H^T = eH^T$

This is because any valid codeword multiplied by the transpose of the H-matrix gives a zero vector.

In other words, $xH^T = 0$.

The decoder (de-compressor, error-pattern estimator) takes **s** as input and outputs the estimate **ê** of **e**. The corresponding estimate **X** of **x** is given by $X = r - \hat{e}$.

Although, syndrome coding techniques could be developed using a variety of codes, K. Zhang et al (Zhang *et al.*, 2014) showed that the best performance could be achieved with codes that are specifically designed for syndrome coding. The authors concluded, among

other things, that these codes have the highest value of the equivocation rate, which is an information secrecy metric for a given code length and code rate.

In their 2002 paper, Angelos D. Liveris et al (Liveris, Xiong & Georghiades, 2002) showed how LDPC codes can be used as an application of the Slepian-Wolf theorem for correlated binary sources. Their approach was based on viewing the correlation as a channel and applying the syndrome concept is focused on the asymmetric case of compression with side information. They simulated a regular and three irregular rate-half codes of length 10000 and 100000 for 100 iterations in the decoder. The simulated results show that the practical lossless compression achieved by LDPC codes is higher than that of the turbo schemes available at the time for binary cases.

LDPC codes are forward error-correction codes first proposed by Gallager (Gallager, 1962) in his 1962 PhD thesis at MIT. As their name suggests, LDPC codes are block codes with parity check matrices that contain only a very small number of non-zero entries. It is the sparseness of the parity check matrices which guarantees both a decoding complexity which increases only linearly with code length and a minimum distance which also increases linearly with the code length.

A good LDPC code for channel coding is not necessarily good for SW coding as pointed out in (Bhattar, Ramakrishnan & Dasgupta, 2010) and (Chen, He & Jagmohan, 2009b). This is because the source distribution in SW coding is not necessarily uniform and the correlation channel is not necessarily symmetric (Cheng *et al.*, 2009), (Toto-Zarasoa, Roumy & Guillemot, 2010). The channel coding scheme and the Slepian-Wolf coding scheme thus require codes of different rate and code degree distributions. Notwithstanding, it has been showed that for a given SW source distribution and a given correlation channel,

it is possible to identify an equivalent channel which requires the same coding rate and which leads to the same LDPC decoder performance (Chen, He & Jagmohan, 2009a).

Cover (Cover, 1975) derived a simpler proof for the data compression theorem of Slepian and Wolf and also established that the Slepian-Wolf theorem is true without change for the arbitrary ergodic processes and countably infinite alphabets.

Any decoder which accepts soft inputs and delivers soft outputs can be used for iterative decoding of two-dimensional systematic convolutional codes using log-likelihood algebra (Hagenauer, Offer & Papke, 1996). The authors concluded that iterative decoding is possible for convolutional codes in systematic feedback form, for any systematic block code or for combinations thereof. They also observed that the key to achieving nearly optimal performance is the proper transferring of extrinsic information from one iteration to the next. Optimal and suboptimal decoders with reduced complexity are presented. They concluded that very simple component codes such as the 4-state convolutional code and the Hamming code are sufficient to achieve surprisingly good results.

An analysis under the iterative decoding of coset LDPC codes over GF(q) designed for use over arbitrary discrete-memoryless channels was presented in (Bennatan & Burshtein, 2006). They showed that coset GF(q) LDPC codes are a natural extension of binary LDPC codes to nonbinary channels by generalization of the analysis that had been developed from binary LDPC codes to coset GF(q) LDPC codes. The have also generalized the all-zero codeword assumption, the symmetry property as well as channel equivalence due to the fact that random coset analysis helps overcome the absence of output symmetry. Although they focused on the decoding problem, a by-product of their generalization of the all-zero codeword assumption is that no encoder needs to be implemented. Finally, using

quantization coupled with the generalization of Extrinsic Information Transfer (EXIT) charts, they obtained simulation results for the Additive White Gaussian Noise (AWGN) channel within 0.56 dB of the Shannon limit at a spectral efficiency of 6 bits/s/Hz. However, the authors could not realize density evolution, instead they resorted to permutation invariance for the analysis of stability property as well as approximation. Density evolution was implemented in (Dupraz, Savin & Kieffer, 2015) and others.

The use of LDPC codes for non-uniform sources under distributed source coding paradigm was investigated in (Bhattar, Ramakrishnan & Dasgupta, 2010) and it was showed that several capacity approaching LDPC codes do approach the Slepian-Wolf bound for non-uniform sources as well. Their Monte Carlo simulation results show that highly biased sources can be compressed to 0.049 bits/sample away from Slepian-Wolf bound for moderate block lengths.

Yong Fang (Fang, 2009) showed that the redundancies in LDPC syndromes can be used to estimate the crossover probability between two correlated binary sequences and suggested that their proposed algorithm also applies to irregular LDPC codes by means of the mean of intrinsic Log-Likelihood Ratios (LLRs)

In their paper, F. Daneshgaran et al (Daneshgaran, Laddomada & Mondin, 2009) proposed a novel iterative joint decoding algorithm based on LDPC codes for compression of correlated sources at rates approaching the Slepian-Wolf bound. They also demonstrated that significant compression can be obtained, depending on the extent of the actual source correlation estimated through an iterative paradigm, relative to the case the decoder does not use the implicit knowledge of the existence correlation. They concluded that LDPC decoding does not converge when the decoder does not iterate for estimating the actual

value of the cross-over probability but uses instead its mean value which is assumed to be implicitly known.

J.J. Micallef, et al (Micallef, Farrugia & Debono, 2011) considered the construction of rate-adaptive LDPC codes in which the edges of the variable nodes receiving unreliable information are evenly distributed among all the check nodes. They attributed sub-optimal performance of error correcting codes used for DSC to failure of the assumption of a random distribution of errors. This occurs because prediction of the error distribution is possible in certain DSC application. The performance these codes so constructed is similar to that of the traditionally constructed codes when error prediction fails. However, the gap to the SW bounds are reduced by up to 56% with improvements in accurate error predictions.

In contrast to the situation treated by Slepian and Wolf, where knowledge of the side information at the encoder does not allow a reduction of the transmission rate, A.D. Wyner and J. Ziv in (Wyner & Ziv, 1976), showed that in fact, the knowledge of the side information at the encoder permits transmissions at a given distortion level using a smaller transmission rate. The authors determined the infimum of rates such that communication is possible at an average distortion level. It is interesting to note that, at zero distortion, the Wyner-Ziv problem simply transforms to the Slepian-Wolf problem.

M.J. Wainwright and E. Martinian (Wainwright & Martinian, 2009) described and analyzed the source and channel coding properties of a class of sparse graphical codes based on compounding a Low-Density Generator Matrix (LDGM) with a LDPC code. They established that there exist codes from this ensemble, with all degrees remaining bounded independently of block length, that are optimal for both channel and source coding

simultaneously with binary data. They also showed that finite-degree constructions can achieve any pair on the rate-distortion curve of the binary symmetric channel in the context of lossy compression. For channel coding on the other hand, they proved that the same finite-degree codes can achieve any pair on the capacity-noise curve of the binary symmetric channel. Finally, the authors showed that the compound construction has a nested structure that can be exploited to achieve the Wyner-Ziv bound for Source Coding with Side Information (SCSI) as well as the Gelfand-Pinsker bound for Channel Coding with Side Information (CCSI).

S. S. Pradhan et al (Pradhan, Chou & Ramchandran, 2003) explored the information-theoretic duality between source coding with side information at the decoder and channel coding with side information at the encoder. They carried out a mathematical characterization of the functional duality between the classical source and channel coding and formulated the precise conditions under which the optimal encoder for one problem is functionally identical to the optimal decoder for the other problem. They further generalized the result of Wyner and Ziv relating to no rate loss for source coding with side information from Gaussian to more arbitrary distributions. Several examples corresponding to both discrete and continuous-valued cases were considered to illustrate the formation. Their geometric treatment inspires the construction and dual use of practical coset codes for a large class of applications for coding with side information such as watermarking, information-hiding communication systems and distributed sensor networks.

In a recent publication, E. Dupraz et al (Dupraz, Savin & Kieffer, 2015) investigated the problem of designing good non-binary LDPC codes for Slepian-Wolf coding based on

density evolution that gives the asymptotic error probability of the decoder for given code degree distribution. Since the correlation channel in Slepian-Wolf coding is not necessarily symmetric and source distribution has to be taken into consideration, the assumption of symmetry upon which density evolution is developed does not necessarily hold. The authors optimized the code degree distribution by means of differential evolution. Asymptotic analysis and finite-length simulations illustrated the performance gain at considering optimized degree distributions.

The performance of the normalized BP-based and the offset BP-based algorithms was analyzed by means of density evolution (Chen, Fossorier & Ieee, 2002). They showed that the performances of these two improved BP-based algorithms are within limits of the BP-based algorithm with a properly chosen parameter. The simulations were carried out on codes with moderately long code lengths and the results obtained validates the proposed optimization.

J. Chen et al (Chen, He & Jagmohan, 2009a), (Chen, He & Jagmohan, 2009b) also studied Slepian-Wolf coding with a mismatched decoding metric and established two different dualities between Slepian-Wolf coding and channel coding under mismatched decoding. These dualities namely, the type-level duality and the linear codebook-level duality, provide a systematic framework for comparing linear Slepian-Wolf codes, non-linear Slepian-Wolf codes, and variable-rate Slepian-Wolf codes. They concluded that the minimum rate achievable with non-linear Slepian-Wolf codes under mismatched decoding can be strictly lower than that achievable with linear Slepian-Wolf codes. Precisely, they established that each Slepian-Wolf coding problem is equivalent to a channel coding problem for binary-input output-symmetric channel under density evolution.

S. Cheng et al (Cheng *et al.*, 2009) proposed an adaptive Slepian-Wolf decoder using particle filtering based belief propagation. They showed that their algorithm can simultaneously reconstruct a compressed source and estimate the joint correlation between the sources. Their approach can also achieve higher compression under varying correlation compared to the conventional Slepian-Wolf coder based on standard belief propagation. Consequently, they have been able to resolve a major difficulty affecting the practical use of Slepian-Wolf coding which is that the precise correlation among sources needed to be known a priori. Key to their solution is the fact that the error probability of a Binary Symmetric Channel (BSC) is updated for each variable node step by step by introducing the particle filtering algorithm in the left-hand side of the factor graph in the standard belief propagation algorithm.

In a subsequent paper (Cui *et al.*, 2011), same authors proposed the first adaptive asymmetric and non-asymmetric SW coding schemes that can perform online estimation of the correlation among sources while decoding. This is essential for practical implementation of SW coding since encoders cannot communicate to each other and thus cannot perform correlation estimation.

J. Chou et al (Chou, Pradhan & Ramchandran, 2003) studied the problem of rate-distortion efficient constructions for the problem of SCSI. Their work was aimed to reduce the gap between theory and practice regarding the Wyner-Ziv theorem from information theory that prescribed rate-distortion performance bounds for the SCSI problem. The proposed two different frameworks based on a trellis construction and a turbo-based construction and obtained impressive results.

S. Chung et al (Chung, Richardson & Urbanke, 2001) used a Gaussian approximation for message densities under density evolution to simplify the analysis of decoding algorithms for memoryless binary-input continuous-output AWGN channels and sum-product decoders. They achieved this by converting the infinite-dimensional problem of iteratively calculating message densities that was needed to find the exact threshold to a one-dimensional problem of updating means of Gaussian densities. This made it easier to design good irregular LDPC codes for AWGN channels and also allows the quick calculation of thresholds and understand the behaviour of the decoder better. They also demonstrated how the optimization of degree distributions can be visualized graphically.

Three new innovations for compression using LDPC codes for the Slepian-Wolf problem were introduced in (Coleman *et al.*, 2004). This was inspired by improvements of analogous results in multiple access channel coding literature. These three are: a general iterative Slepian-Wolf decoding incorporating a graphical structure of all the encoders and operates in a turbo-like fashion, a source-splitting to enable low-complexity pipelined implementations of Slepian-Wolf decoding at rates besides corner points of the Slepian-Wolf region, and a linear programming relaxation to maximum-likelihood sequence decoding that exhibits the maximum-likelihood certificate property. This is another demonstration of the duality between source and channel coding, taking ideas developed for channel coding and transforming them appropriately to construct new source coding techniques.

Subsequently in (Coleman, Medard & Effros, 2005) the authors again showed that there exist linear codes for which minimum-entropy decoders achieve the same error exponent as maximum-likelihood decoders using method of types. They introduced practical

approximation algorithms for minimum entropy decoding by exploiting two key observations; that the number of distinct types grows exponentially in n and that recent results in the optimization literature have illustrated polytope projection algorithms with complexity that is a function of the number of vertices of the projected polytope. They explicitly demonstrated linear code constructions that admit provable good performance in the binary case.

Authors in (Davey & MacKay, 1998) showed that LDPC codes defined over finite fields GF(q) of order $q > 2$ show significantly improved performance compared to their analogous binary counterparts which had been shown to near Shannon limit performance when decoded using probabilistic decoding algorithm. They also presented the results of Monte Carlo simulations of the decoding of infinite LDPC codes which can be used to obtain good constructions for finite codes.

A practical coding scheme for universal source coding with side information at the decoder was developed in (Dupraz, Roumy & Kieffer, 2013). The scheme encompasses the determination of the coding rate as well as the design of the encoding process, both contributions resulting from the information-theoretical compression bounds of universal lossless source coding with side information. They also proposed a novel decoder in which the available information regarding the class is considered. Finally, their proposed scheme avoids the use of a feedback channel or the transmission of a learning sequence, which would result in an increase in rate at a finite length.

X. Hu et al (Hu, Eleftheriou & Arnold, 2005) proposed a general method for the construction Tanner graphs with a large girth by establishing edges between symbol and check nodes in an edge-by-edge manner. The novelty in their approach compared to

existing constructions is in its simplicity and flexibility. Its complexity is such that it can easily be used for constructing codes of very large block lengths and good girth guaranteed by the lower bound. Also, it successfully generates good codes for any given block length and any rate when using density-evolution-optimized degree sequence. It can also be used to generate linear-time-encodable LDPC codes with a slight modification.

G. Lechner et al (Lechner, Weidmann & Ieee, 2008) studied the optimization of binary LDPC codes for the q-ary symmetric channel with moderate alphabet sizes q. They derived a factor graph representation of the front-end and showed that it can be processed with a complexity linear in the number of bits per symbol, m. They also used extrinsic information transfer analysis to optimize binary LDPC codes and showed that these codes perform close to the capacity of the q-SC over a wide range.

T. Matsuta et al (Matsuta, Uyematsu & Matsumoto, 2010) established the existence of a universal Slepian-Wolf source code using LDPC matrices in the case where the source is stationary memoryless. At the time, the existing LDPC matrices for Slepian-Wolf source coding were based on maximum likelihood decoding and thus not universal. Their work allowed for arbitrarily decreasing the error probability for all sources whose achievable rate region contains the rate pair encoders even if the probability distribution of the sources is unknown.

T. J. Richardson et al (Richardson, Shokrollahi & Urbanke, 2001) designed very good LDPC codes from highly irregular bipartite graphs with carefully chosen degree patterns on both sides. This work is based on (Richardson & Urbanke, 2001) in which a general method for determining the capacity of LDPC codes under message-passing decoding when used over any binary-input memoryless channel with discrete or continuous output

alphabets was presented. The authors have established three vital proofs; One, that the probability densities at the message nodes of the graph possess a certain symmetry, assuming that the underlying communication channel is symmetric. Secondly, that assuming no cycles, the message densities always converge as the number of iterations tend to infinite. And thirdly, a stability condition which implies an upper bound on the fraction of errors that a belief-propagation decoder can correct when applied to a code induced from bipartite graph with a given degree distribution. Simulation results show that the performance of the codes is very close to the asymptotic theoretical bounds.

Pradhan and Ramchandran (Pradhan & Ramchandran, 2005) introduced a constructive approach for distributed binning of two codebooks with applications to many multi-terminal source representation problems. They employed generalized coset codes constructed in a group-theoretic setting for this approach and analyzed the performance in terms of distance properties and decoding algorithms.

Inspired by (Pradhan & Ramchandran, 2005), V. Stankovic et al (Stankovic *et al.*, 2006) addressed the problem of practical code design for general multi-terminal lossless networks where multiple memoryless correlated binary sources are separately compressed and sent while each decoder receives a set of compressed sources and attempts to jointly reconstruct them. The novelty in this design is the possibility to approach the theoretical limits with a single channel code for ant rate allocation among the encoders. They authors also provided a detailed solution for both asymmetric and symmetric Slepian-Wolf coding based on partitioning a single channel code (Stankovic *et al.*, 2004). The devised a powerful scheme that is capable of approaching any point on the Slepian-Wolf bound by using systematic

IRA and turbo codes. They obtained results which are 0.04 bits away from the theoretical limit in both symmetric and asymmetric Slepian-Wolf cases.

V. Toto-Zarasoa et al (Toto-Zarasoa, Roumy & Guillemot, 2010) showed that the problem of non-asymmetric Slepian-Wolf coding of two correlated non-uniform Bernoulli sources is not symmetric in both sources unlike the case of uniform sources due to the asymmetry induced by two underlying channel models namely additive and predictive binary symmetric channels. They also developed a joint non-asymmetric decoder of the two sources based on LDPC codes and message passing decoding. Finally, they suggested a necessary and sufficient condition for the recovery of the two sources, that imposes a triangular structure of a sub-part of the equivalent matrix representation of the code.

Z. Wang et al (Wang, Li & Xu, 2009) proposed an improved decoding algorithm based on joint bit-plane decoding for distributed video coding. This coding is based on Slepian-Wolf and Wyner-Ziv theorems proposed in 1970s. Their algorithm uses the decoded bit-plane results of quantization coefficients as a prior knowledge and decodes the other bit-plane to avoid correlation weakening between sources when decoding bit-planes independently. They also made full use of the correlation between Wyner-Ziv frames and key frames thus enhancing rate-distortion performance of distributed video coding without increasing decoding complexity.

Mina Sartipi and Faramarz Fekri (Sartipi & Fekri, 2005) proposed a scheme for distributed source coding that achieves any arbitrary rate on the Slepian-Wolf rate region using a single systematic LDPC code. Their method is based on sending fractions of the information bits along with a fraction of parity bits generated by the LDPC code. They also proposed to use non-uniform LDPC codes for this application and also generalized their approach to any

arbitrary rate on the Slepian-Wolf rate region. Furthermore, they illustrated that the design procedure for LDPC codes simplifies to the design of rate adaptive LDPC codes that need unequal error protection. They also showed that the performance of distributed source coding at any arbitrary rate is almost the same as that of asymmetric rates. Their approach does not suffer from the problems of heavily damaged or propagation of the errors because each of the sources is in the decoding algorithm is decoded independently. Finally, they did simulations with code block length of 1000, which is similar to that being used in this study. This is significant for a fair comparison.

Serener et al (Serener, Natarajan & Gruenbacher, 2008) introduced highly optimized short-block-length LDPC codes along with Walsh-Hadamard spreading to Orthogonal Frequency-Division Multiplexing (OFDM) systems. The authors showed that the use of spreading along with code optimization and girth conditioning improves performance and lowers the error floor for short-length LDPC codes.

Jin et al (Jin *et al.*, 2018) developed a novel LDPC decoding algorithm based on Markov Chain Monte Carlo (MCMC) method. They also introduced two improved versions, MCMC-S (this version introduces a sort function, $s = sort(d_i, a)$ as an alternative to using an unreasonably small noise variance for computations) and MCMC-L (L stands for the number of preserved optimal paths) which achieve better results. These algorithms outperform the traditional BP decoding method. The authors also gave a Very Large-Scale Integration (VLSI) architecture of the proposed methods.

Guo (Guo, 2018) proposed a design of LDPC code via the masking technology and progressive optimization. The quantity of short loops in check matrix is reduced by masking the elements in the matrix. This also improves the error correcting ability of the

code. The authors also showed that the performance of new codes can exceed the capacity of LDPC codes generated by randomized construction methods.

Subsequently, the authors proposed another scheme (Sartipi & Fekri, 2008) which improves the performance of distributed source coding of two sources considerably. A study was also made of distributed source coding of three sources that are pairwise correlated with the same correlation probability.

Amongst the most common approach to the distributed source coding of continuous-valued sources, like audio-visual data, is to first convert them to discrete-valued sources and then apply lossless Slepian-Wolf coding (Pradhan & Ramchandran, 2003; Zixiang, Liveris & Cheng, 2004). The problem of this approach is the introduction of quantization and binning losses by the source encoders. It is understood from Shannon that block codes of sufficiently large length are asymptotically optimal. This justifies the use of binary channel codes like LDPC and Turbo Codes for practical Slepian-Wolf coding as highlighted above. However, the problem arises when low delay is imposed on the system and as such, long length LDPC and Turbo Codes are not very useful. On the other hand, analog mapping has been used (Akyol *et al.*, 2014; Chen & Tuncel, 2011) in attaining zero-delay source-channel coding. Although, they have lower complexities, analog mapping techniques do not benefit from the advantages of digital communications.

Several attempts (Vaezi, 2014; Vaezi, Combernoux & Labeau, 2013; Vaezi & Labeau, 2012a; Vaezi & Labeau, 2012b; Vaezi & Labeau, 2013; Vaezi & Labeau, 2014) have been made in finding a balance to these two extremes, that is, sufficiently large length block codes and short-length, zero-delay codes. These attempts were based on the original works by Marshall (Marshall, 1984) and Wolf (Wolf, 1983) who first proposed the Discrete

Fourier Transform (DFT) to tackle the problem of error correction in the real field using real-number codes. Subsequently, Marshall also introduced the BCH-DFT codes as an important subclass of the DFT codes. Interestingly, Wolf, in collaboration with Slepian laid the very foundation for distributed source coding a decade earlier.

Y. H. Chen and T. K. Truong (Chen & Truong, 2011) presented a general algorithm for decoding the binary systematic QR codes using lookup tables. Although, the proposed model is used in decoding either reducible or irreducible generator polynomials, the authors suggested that the number of elements in the Galois field would be less than the sum of all correctable error patterns provided that the generator polynomial of the QR codes is reducible. Simulation results for the (31, 16, 7) QR code and the (73, 37, 13) QR code shows reduction in memory requirements up to 92 percent. The problem with lookup tables however is the memory requirement increases exponentially with increase in code length.

Subsequently, Lin et al (Lin *et al.*, 2010) modified the algebraic decoding of the eight possible errors of the (89, 45, 17) binary QR code using an efficient determination algorithm of the primary unknown syndromes. Their soft decision algorithm modification is said to be four times faster in computation time compared to the previous hard decision decoding algorithm while maintaining the same error patterns. However, there is a high computational complexity associated with simulating this soft decision decoding with the hard decoding algorithm of Chase-II (Chase, 1972) in other to obtain performance measures for comparison.

It follows that Yong Li et al (Li *et al.*, 2018) came up with what is considered to be a natural generalization of the cyclic weight (CW) algorithm for the (47, 24, 11) QR code developed by Lin et al (Lin *et al.*, 2010). The authors developed an algorithm for faster decoding of

the binary systematic QR codes based on the Difference of Syndromes (DS) which combines the advantages of the syndrome-weight algorithm and properties of the cyclic codes. The DS algorithm improves the decoding efficiency and memory usage while using the (47, 24, 11), the (71, 36, 11), the (73, 37, 13), and the (89, 45, 17) QR codes as examples. The best among all QR codes of lengths less than 100 is the (89, 45, 17) QR code and the proposed algorithm improved the decoding speed by 26 times and saved up to 76.6 percent memory compared to the best pre-existing decoding algorithm.

Pengwei et al (Zhang *et al.*, 2015) presented a hard decision scheme to facilitate a faster decoding of the (47, 24, 11) QR code that directly determines the coefficients of the error-locator polynomial by eliminating unknown syndromes in Newton identities and simplifying the condition that indicates the occurrence of four errors. A reliability-based shift-search algorithm is used to decode weight-5 error patterns while a previous scheme is used in decoding up to three errors. This proposed scheme reduces the decoding complexity and saves memory while maintaining the same error-rate performance. The authors also used the same scheme in decoding the (71, 36, 11) QR code (Li *et al.*, 2015).

Tsung-Ching Lin et al (Lin *et al.*, 2016) developed a new speed-up approach for decoding a binary systematic (71, 36, 11) QR code by simplifying the step of calculating the condition and avoiding the calculation of the unknown syndromes. This approach also uses the channel measurement information proposed by Chase (Chase, 1972) to sequentially invert the bits of the received word until one of the error is cancelled for the five-error case.

Xuemin Chen et al (Chen, Reed & Truong, 1994) presented a scheme for half rate binary quadratic residue (QR) codes using Binary Phase-Shift Keyed (BPSK) modulation and hard decoding. Performance measures obtained theoretically and by means of simulations

are compared with commonly used half rate convolutional codes with constraint lengths from 3 to 7. The authors showed that the binary QR codes of different lengths are equivalent in error-correction performance to some half rate convolutional codes which have a constraint length that corresponds to the error-control rate and minimum distance of the QR codes.

Gregory et al (Dubney *et al.*, 2009) also developed an algorithm that was based on the an idea first developed by Reed (Reed, 1959) in a 1959 MIT Lincoln Laboratory Report to facilitate faster decoding of the (47, 24, 11) QR code. This model uses real channel data to estimate the individual bit-error probabilities in a received word and then sequentially inverts the bits with the highest probability of error until one of the errors is canceled. Thereafter, the remaining errors are corrected by algebraic decoding techniques. It is also an appropriate modification to the algorithm developed by Chase (Chase, 1972).

1.3 Thesis Aim and Objectives

The aim of this thesis is to optimize the performance of short-to-medium block length LDPC codes on the Binary Symmetric Chanel as well as the Additive White Gaussian Noise for using Progressive Edge Growth (PEG) algorithm-based constructions. Consideration would also be given to both lossless and lossy compression of correlated information sources. The lossy scheme would imply the implementation of the concept of rate distortion. Furthermore, because capacity-achieving channel codes require unbounded complex encoder/decoder with infinite number of signaling degrees of freedom or block-length, the performance of DSC based on LDPC codes is highly affected in practical real-

time applications where delay and complexity limitations are stringent. Thus, this work is set to investigate the problem of distributed source-channel coding based on QR codes with a view of designing optimal codes with short code-lengths.

1.4 Thesis Organization

The remaining part of this thesis is organized as follows.

Chapter 2 introduces distributed source coding and discusses different schemes under DSC systems, including point-to-point source coding, source coding with side information, as well as distributed lossless and lossy source coding . It also discusses the different kinds of codes and channels implemented in this research such as the Hamming codes and BCH codes.

Chapter 3 discusses the message passing (iterative) decoding of LDPC codes for DSC, belief propagation algorithm decoding implementation, as well as the modifications of BP/SPA algorithms for the Slepian-Wolf AWGN and binary symmetric equivalent channels.

Chapter 4 highlights the challenges of improving the performance of short-length codes for real-time applications, quadratic residue codes, the Dorsch decoder and its modifications for DSC, simulations for performance measure, maximum likelihood decoding, and the complexity associated with its implementation.

Chapter 5 summarizes the entire thesis, highlights the contributions to the body of knowledge as well as the recommendations for future works.

Chapter 2

Distributed Source Coding, Hamming codes and Bose-Chaudhuri-Hocquenghem (BCH) codes

2.1 Introduction

This chapter discusses the basics of distributed source coding and identifies the major schemes under DSC systems. It also highlights the step-by-step development of a straightforward coding model as well as a Slepian-Wolf coding model of two correlated binary sources using very simply short length codes, viz-a-viz the [7, 4, 3] Hamming code and the [15, 7] BCH codes.

2.2 Distributed Source Coding

Consider a communication system with two separate correlated signals X and Y coming from two sources that cannot communicate with each other, as shown in Figure 2.1. This arrangement is known as DSC because encoding is done independently or in a distributed manner. However, the receiver can perform joint decoding since it can see both encoded signals. An example of such a system is a smart metering network composed of spatially separated smart meters, sending correlated observations to a common fusion center. The problem is to find the minimum required encoding rate such that both signals can be faithfully recovered without any loss.

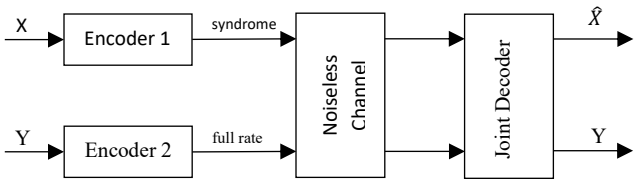


Figure 2.1: Block diagram representing Slepian-Wolf coding.

Slepian and Wolf (Slepian & Wolf, 1973) laid the foundation of this problem namely separate lossless compression of two correlated sources. They proved the counter-intuitive results that separate encoding with joint decoding achieves the same compression rate as joint encoding does. Lossless source coding with side information at the decoder is a special case of Slepian-Wolf coding where one signal, known as the side information, is available at the decoder. Wyner and Ziv (Wyner & Ziv, 1976) extended this special case to a more general one, namely lossy source coding with side information at the decoder. Intriguingly, when the source and side information are jointly Gaussian and the distortion measure is the MSE, Wyner-Ziv coding does not suffer a rate loss compared to the case where the side information is also available at the encoder. In other words, separate encoding is as efficient as joint encoding in lossy source coding. However, in general, Wyner-Ziv coding incurs some loss in rate when compared to lossy source coding with side information available at both the encoder and decoder.

Perfect representation of continuous-valued random variable requires an infinite number of bits. This means that any description of such a variable with finite number of bits is imperfect and incurs some distortion. Thus, a basic problem in rate-distortion theory is to find the minimum expected distortion for a particular rate, given a source distribution and

a distortion measure (Cover, 2006). The rate-distortion function $R(D)$, is defined by the infimum of rates R such that (R, D) is achievable (El Gamal & Kim, 2011), for a given distortion value D.

2.2.1 Point-to-Point Source Coding

Theorem 2.1. Shannon's Lossy Source Coding Theorem (Cover, 2006)

The rate-distortion function for a Discrete Memoryless Source (DMS) X with a distribution $p(x)$ and a distortion measure $d(x, \hat{x})$ is equal to the associated information rate-distortion function:

$$R(D) = \min_{p(\hat{x}|x):\mathbb{E}(d(X,\hat{X}))\leq D} I(X;\hat{X}), \tag{2.1}$$

for $D \geq D_{min} \triangleq min_{\hat{x}}\mathbb{E}(d(X,\hat{x}))$, where $I(X;\hat{X})$ is the mutual information of X and $\hat{X}$.

This means that a rate R is achievable with distortion D for $R > R(D)$, but it is not achievable for $R < R(D)$. When $D = 0$, $\hat{x} = x$; and thus $R(0) = I(X,\hat{X}) = H(X)$, which is the optimal rate for lossless source coding. This shows that lossless source coding theorem is a special case of lossy source coding theorem.

2.2.2 Source Coding with Side Information

Literature has showed that several extensions of Theorem 2.1 have been studied for cases when casual or non-casual side information (SI) is available at the encoder or decoder (El Gamal & Kim, 2011). The non-casual side information is when the entire side information sequence is available. When side information Y is available both at the encoder and decoder, the rate-distortion function becomes:

$$R_{SI-ED}(D) = \min_{p(\hat{x}|x,y):\mathbb{E}(d(X,\hat{X}))\leq D} I(X;\hat{X}|Y), \tag{2.2}$$

This is known as the conditional source coding problem, usually represented by the conditional rate-distortion function $R_{X|Y}(D)$. It potentially decreases the required encoding rate to achieve the same distortion (Dragotti & Gastpar, 2009), when compared with the case where side information is not available at the encoder or decoder.

Theorem 2.2. Wyner-Ziv's Theorem (Wyner & Ziv, 1976)

If (X, Y) are two discrete memoryless sources, the rate-distortion function of X with distortion measure $d(x, \hat{x})$ when side information Y is available only at the decoder is equal to:

$$R_{SI-D}(D) = \min_{p(u|x), \hat{x}(u,y):\mathbb{E}(d(X,\hat{X}))\leq D} I(X;U|Y), \tag{2.3}$$

where U is an auxiliary random variable and $U \to X \to Y$ forms a Markov chain.

This rate-distortion function is mostly represented by $R_{WZ}(D)$ or $R_{X|Y}^{WZ}(D)$ in literature and it can be verified that the difference between $R_{SI-D}(D)$ and $R_{SI-ED}(D)$ is understood to be in the sense that the minimum is taken over $p(u|x)$ and $p(u|x, y)$, respectively. This means that $R_{SI-D}(D) \geq R_{SI-ED}(D)$ and indicates that $R_{SI-D}(D) - R_{SI-ED}(D) \geq 0$ is incurred when the encoder does not know side information. Wyner and Ziv, however proved the intriguing result that $R_{SI-D}(D) = R_{SI-ED}(D)$ for Gaussian memoryless sources and mean-squared error distortion. Particularly, without loss of generality, for $X \sim \mathcal{N}(0, \sigma_X^2)$ and side information $Y = X + U$ with $U \sim \mathcal{N}(0, \sigma_U^2)$ independent of X:

$$R_{WZ}(D) = R_{X|Y}(D) = \begin{cases} \frac{1}{2}\log_2\left(\frac{\sigma_{X|Y}^2}{D}\right) & \text{if } 0 \le D \le \sigma_{X|Y}^2 \\ 0 & \text{if } D > \sigma_{X|Y}^2 \end{cases} \quad (2.4)$$

where $\sigma_{X|Y}^2 = \frac{\sigma_X^2\sigma_U^2}{\sigma_X^2+\sigma_U^2}$. Although in the quadratic case there is no loss if the encoder does not have side information, the underlying coding scheme is very different from the case that side information is available to both encoder and decoder. Furthermore, when the side information suffers from rate loss, this result does not hold, and the exact solution is unknown.

2.2.3 Distributed Lossless Source Coding

Slepian and Wolf (Slepian & Wolf, 1973) laid the foundation of DSC, where statistically dependent signals are encoded in a distributed manner but decoded jointly. They proved the counter-intuitive result that separate encoding can be as effective as joint encoding. From Shannon's source coding theorem (Cover, 2006), for probability of decoding error to approach zero, the minimum sum rate is simply the joint entropy *H(X, Y)*. Surprisingly, the same combined rate is sufficient even if the signals are encoded separately as described in the following theorem.

Theorem 2.3. Slepian-Wolf's Theorem (Slepian & Wolf, 1973)

The optimal rate region for distributed coding of two DMS sources $(X, Y) \sim p(x, y)$ is the set of rate pairs (R_X, R_Y) that:

$$R_X \ge H(X|Y)$$
$$R_Y \ge H(Y|X)$$
$$R_X + R_Y \ge H(X, Y) \quad (2.5)$$

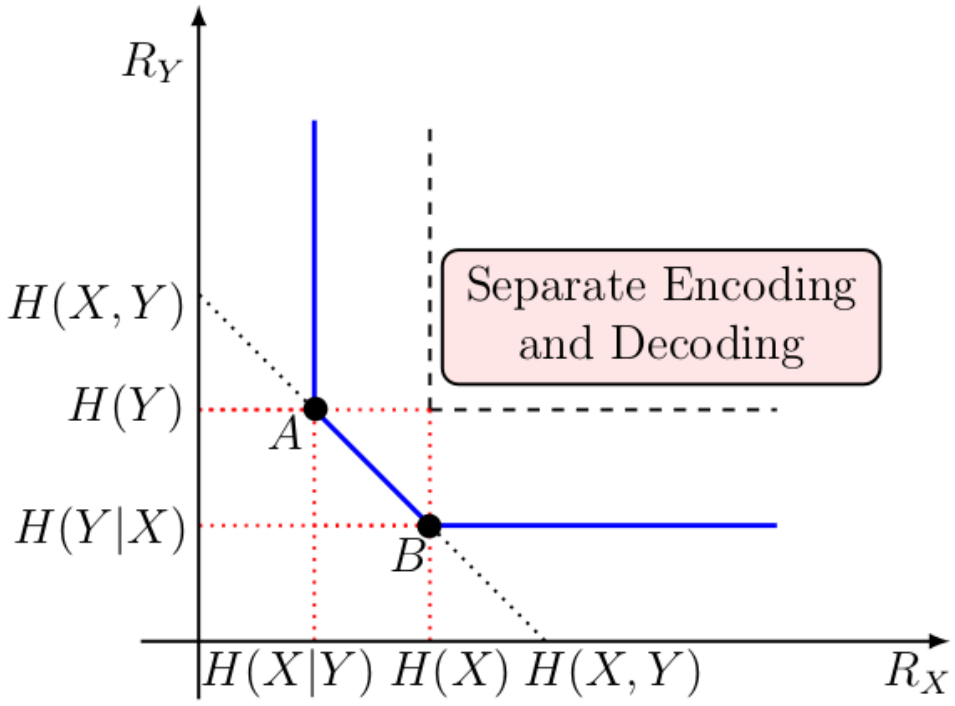


Figure 2.2: Achievable rate regions for the Slepian-Wolf coding (solid lines) and separate encoding with separate decoding (dashed lines).

Figure 2.2 illustrates the Slepian-Wolf rate region compared to the conventional entropy coding. This scheme reduces the rate required for lossless transmission of correlated sources. With conventional separate entropy encoding and separate decoding, one can only achieve $R_X \geq H(X)$ and $R_Y \geq H(Y)$; thus $R_X + R_Y = H(X) + H(Y)$ which is greater than $H(X,Y)$ for correlated X and Y. Slepian and Wolf showed that the corner point A is achievable even when the first sender does not know Y. Thus, the sum rate $R_X + R_Y = H(X,Y)$ is achievable even though the sources are separately encoded. The limit for lossless DSC can be smaller than that of separate coding but the compression is no more error free. Notwithstanding, the probability of error can be negligible for long sequences.

2.2.4 Distributed Lossy Source Coding

This is an extension of Slepian and Wolf problem in which reconstruction is no longer perfect. Two sources X and Y are separately encoded, and the descriptions are sent over noiseless communication channels to a common decoder, like the lossless DSC. However, in this case the compression is lossy, and the decoder wishes to reconstruct the two sources with distortions D_X and D_Y, respectively.

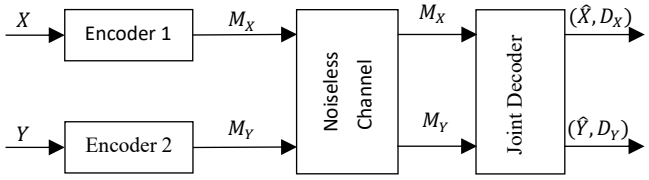


Figure 2.3: Distributed lossy source coding.

The problem is to find the minimum required description pairs (R_X, R_Y) that achieve distortion pair (D_X, D_Y). This problem is more involved, and the solution is not known, except for the quadratic Gaussian case.

2.3 The Hamming Codes

The shortest Hamming code, that is, the [7, 4, 3] code is an example of the quadratic residue codes (these are discussed in more details in Chapter 4) where $l = 2$ and $p = 7$ and has a generator polynomial $(x + \alpha)(x + \alpha^2)(x + \alpha^4) = x^3 + x + 1$ where $\alpha \in GF(2^3)$. The [7, 4, 3] code has $K = 4$ information bits, $N = 7$ coded bits and $(N - K) = 3$ parity bits.

2.3.1. Trivial Coding of the [7, 4, 3] Hamming Code

In the pioneering paper of Slepian and Wolf (Slepian & Wolf, 1973), sixteen cases were considered depending upon the information available to the encoders and decoders through varying configurations of four switches. This is illustrated in Figure 2.4. The first, case-1111 in which all four switches are closed contains nothing new and is obtained by regarding the pair (X, Y) as a new discreet random variable. This is implemented in this section. The most interesting and novel case of Slepian-Wolf's paper, however, is the case-0011 where each encoder sees only its source. Case-0011 is discussed in Section 2.3.2 in detail. What follows here is an explanation of implementation of the straightforward coding of two correlated sources.

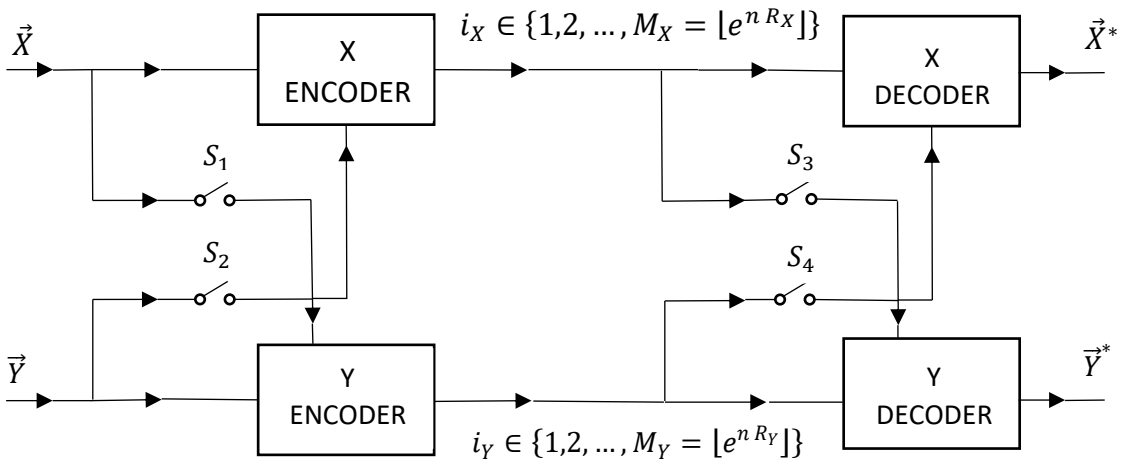


Figure 2.4: Block diagram representing the sixteen cases of correlated source coding.

A text file containing the code (H-matrix) was created systematically in such a way that, only positions in the array with a '1' are represented while ignoring positions with a '0'. The first line of the file shows the dimension of the matrix. In this case, N=3, M=4 indicates

that the matrix has three rows and four columns. For ease of decoding, the H-matrix was arranged in ascending orders of the column converted to decimals.

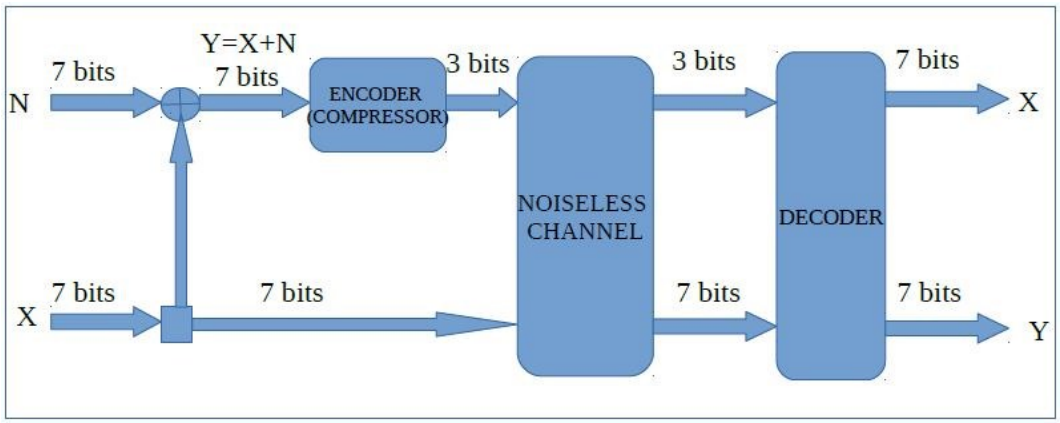


Figure 2.5: Block diagram representing straightforward coding of two correlated sources.

The program generates strings of 7-bit random numbers with the *rand* () function. Additive White Gaussian Noise (AGWN) is then added to obtain a noisy version of the data. The purpose of this is to add errors to the data. Because we are dealing with the [7, 4] Hamming code, only one bit of error can be corrected each time. Hence, the noise vector to be added should have been such that it comprises of an all zeros but one, seven-digit random numbers (that is, 1000000, 0100000, 0010000, 0001000, 0000100, 0000010, and 0000001). The eighth noise vector is the all zeros seven-digit number. This implies that there can be only eight different patterns of noise vector that should added to the original data in other to obtain a noisy version that is correctable.

Subsequently, to obtain the syndromes, the transpose of the difference between the original data and noisy version (in other words, the noise vectors) is then multiplied with the H-matrix. The corresponding syndromes obtained, being just 3 bits are 000, 001, 010, 011, 100, 101, 110 and 111. Hence, compression is achieved by sending just 10 bits (original data [7 bits] plus syndrome [3 bits]) instead of 14 bits that would have been required to

send both sources without any compression.

Finally, at the receiver, since each of the 8 syndromes corresponds to a distinct error vector, it would have been sufficient to implement a look-up table in other to achieve a simple decoder. However, the disadvantage of such approach becomes apparent when the size of the data increases. Particularly, we obtained eight syndromes because each is just three bits long. The number of syndromes grows exponentially with the length of the syndromes. Avoiding a look-up table, the H-matrix is arranged systematically in ascending orders of the columns in decimal. As such, each syndrome, apart from the 000 corresponding to the 0000000 noise vector, is matched to a particular column position in the H-matrix array, ranging from column 0 to column 6. In other words, there is a one-to-one mapping from each received syndrome to a noise (or error) vector and finally to the most probable codeword that would have been sent. This holds as far as no errors or at most, 1-digit error occur between that sent and received messages, in which case, would be detected and corrected. This is because the [7, 4] Hamming code can correct at most one error bit.

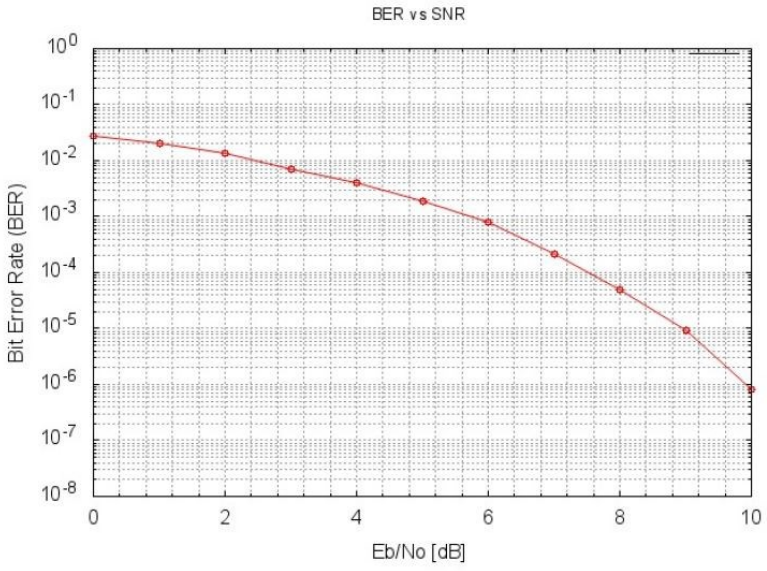


Figure 2.6: Plot of BER against E_b/N_o for the Hamming code.

In the less likely cases where more than one error occurs, the message would not be correctly decoded. In fact, it could result in creating more errors in the process of trying to correct one. To avoid such situations, this program is designed such that, when computing the error vector, once the first error is encountered, it breaks out and ignore subsequent error(s) that might be present in any other bits of the 7 bits number. As a result, even when there are more than one errors, one of it is corrected and no situation occurs when additional errors are created in an attempt to correct one.

Functions are created to generate an array of E_b/N_o from 0 to 10 and the corresponding BERs are also computed. The relationship between E_b/N_o and the probability of error is given by equation 2.6.

$$BER = \frac{1}{2} erfc(\sqrt{SNR}) \tag{2.6}$$

The relationship between E_b/N_o and SNR is established in equation 2.7.

$$SNR = 2 \times R \times E_b/N_o \tag{2.7}$$

R is the rate of the code. It follows that for a half-rate code, SNR = E_b/N_o

A plot of BER vs E_b/N_o as shown in Figure 2.6. This defines an operating point at 10^{-3} gives E_b/N_o of about 5 dB. This implies that, at a random signal-to-noise ratio of, say 5 dB, only one bit was recorded in error out of 1000 bits that was transmitted. This is quite promising at this stage of development of a model.

2.3.2 Slepian-Wolf Coding of the [7, 4, 3] Hamming Code

Again, according to Slepian-Wolf (Slepian & Wolf, 1973), even when each source is constrained to operate without knowledge of the other source, and the decoder has access to both encoded binary message streams, the model can still be coded and transmitted at same rate as if the sources communicate, such that they can be faithfully reconstructed by the decoder.

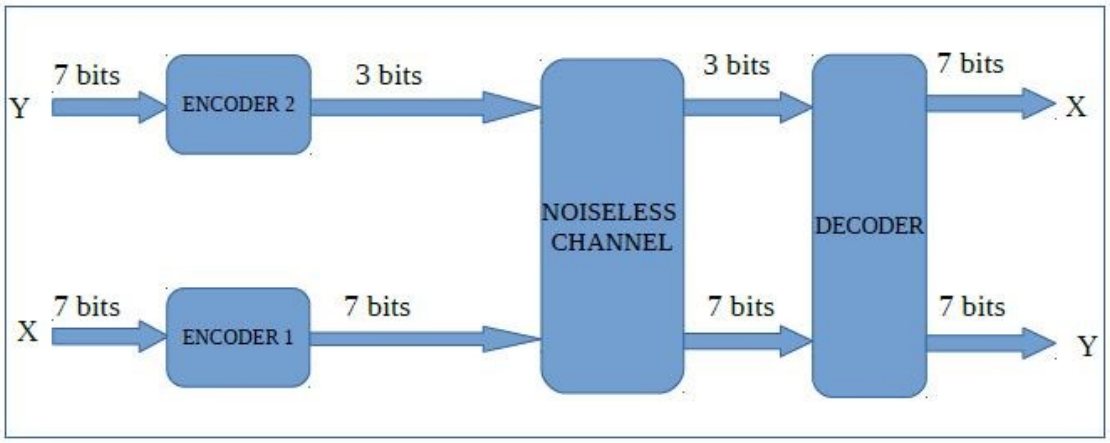


Figure 2.7: Block diagram representing Slepian-Wolf coding of two correlated sources.

The main difference between this implementation and the straight-forward coding is that both sources do not communicate with each other. Hence, in other to obtain a syndrome, which is just three bits, the H-matrix is multiplied by the noisy data at source *Y*. For implementation, this is equivalent to identifying the positions in the H-matrix with non-zero entries, then adding up the entries of the corresponding positions of the noisy data. The random data at source *X* is transmitted wholly without any compression. As in the previous case, a total of 10 bits of data is sent over the noiseless channel. However, at the decoder, the syndrome of sources *X* is also obtained by multiplication with same H-matrix.

Subsequently, both syndromes are *XOR-ed* at the decoder and it becomes interesting to note that the result of this *XOR* operation is same as the syndrome obtained in the previous case where the difference between the sources was multiplied by the H-matrix. And as in the previous case, the H-matrix is arranged systematically in ascending orders of the columns in decimal. As such, each syndrome, apart from the 000 corresponding to the 0000000 noise vector, is matched to a particular column position in the H-matrix array, ranging from column 0 to column 6. In other words, there is a one-to-one mapping from each received syndrome to a noise (or error) vector and finally to the most probable codeword that would have been sent.

The [7, 4, 3] Hamming code is an example of the binary QR code that is further discussed in Chapter 4 and Chapter 5 of this thesis.

2.3.3 Implementation on the [7, 4, 3] Hamming Code

An overview of the major steps taken in the implementation of coding and decoding the [7, 4, 3] Hamming code over the Slepian-Wolf channel is presented in Algorithm 2.1. All algorithms are coded in C/C++ language.

For the purpose of this implementation, $sigma = \sqrt{(pow(10, -0.1 \times E_b/N_o))}$

Algorithm 2.1: Algorithm for Coding and Decoding the [7, 4, 3] Code over the Slepian-Wolf Channel

Total-errors = 0

for *blocks transmitted = 1* **to** *maximum number of codewords transmitted in simulation* **do**

begin

- generate a length *n* random message using *rand* to determine the values of its individual bits for source1.

- convert the binary message from source1 to signal voltages, that is, bit 0 = -1.0V and bit 1 = +1.0V, to represent BPSK modulated signal transmission s_j for $0 \leq j \leq n-1$.
- generate random noise signal of length n to affect converted signal in each bit position using a typical uniformly distributed random number generator from C++ library.

  $$sigma \times \sqrt{(-2 \times \log\left(\frac{(double)rand()}{rand\_max}\right))} \times \cos\left(2 \times m\_pi \times \frac{(double)rand()}{rand\_max}\right)$$
- add converted vector $\boldsymbol{s}$ to noise vector $\boldsymbol{N}$ to obtain the noisy version of the converted signal now, $\boldsymbol{X_j} = \boldsymbol{s_j} + \boldsymbol{N_j}$ for $0 \leq j \leq n-1$.
- quantize the noise signals of the noisy version by converting positive voltages to 1 and negative voltages to 0, we call these source2.
- compute the syndrome of the random bits of source2 by multiplication by the parity check matrix, these become the side information at the decoder.
- decompress the syndrome by performing an exclusive OR operation of the source2 and the syndrome. This is the same as the syndrome of the differences in the sources in straight forward coding.
- do a one-to-one mapping of all possible error vectors to the decompressed data. This mapping is used to decode the correct message sent.

- Repeat these procedures over SNRs from 0 to 10.

end

final error probability, P_e= *Total-errors / maximum number of codewords transmitted in simulation.*

2.4 The Bose, Chaudhuri and Hocquenghem (BCH) Codes

The Bose, Chaudhuri and Hocquenghem (BCH) codes are a large category of powerful random error-correcting cyclic codes and are as well, a remarkable generalization of the Hamming codes for multiple error correction. Binary BCH codes are the most important subclass from both theoretical and implementation standpoints. They were first discovered by Hocquenghem (Hocquenghem, 1959) in 1959 and then, independently rediscovered by Bose and Chaudhuri (Bose & Ray-Chaudhuri, 1960) in 1960. The simplest form of the binary BCH codes which can correct up to two bits of error at any instance is the [15,7]

BCH code. In chapter four, a class of cyclic BCH codes known as QR codes would be discussed.

2.4.1 Slepian-Wolf Coding of the [15, 7] BCH Code

Because Hamming codes are only capable of correcting just one bit of error, there is the need to implement SW coding to correct up-to two errors using other codes, for instance, the BCH code. The BCH codes form a large class of powerful random error-correcting cyclic codes. They are a class of codes that is a remarkable generalization of the Hamming codes for multiple-error correction.

For any positive integers $m(m \geq 3)$and $t(t < 2^{m-1})$, there exists a binary BCH code with the following parameters:

Block length:	$n = 2^m - 1$	(2.8)
Number of parity-check digits:	$n - k \leq mt$	(2.9)
Minimum distance:	$d_{min} \geq 2t + 1$	(2.10)

This code is capable of correction any combination of *t* or fewer errors in a block of $n = 2^m - 1$ digits. This is called a *t*-error-correcting BCH code. In general, entries of the parity-check matrix H of BCH codes are elements in $GF\ (2^m)$. Thus, the binary parity-check matrix for a code is obtained by replacing each entry in H with its corresponding m-tuple over GF (2). For $(m = 4)$ and $(t = 2)$, we obtain a (15, 7) code. This is a code of length n = 15, the message bits, $k = 7$ while the parity-check bits, $n - k = 8$.

A text file containing the code (H-matrix) was created systematically in such a way that, only positions in the array with a '1' are represented while ignoring positions with a '0'. The first line of the file shows the dimension of the matrix. In this case, N=8, M=12 indicates that the matrix has eight rows and a maximum of twelve non-zero entries column-wise. For ease of decoding, the columns of the H-matrix are converted to decimals and the H-matrix is re-arranged in ascending orders of the columns. Also, for ease of coding, a -1 entry indicates the end of entries to any row.

The program is like the previous case of one error-correcting Hamming code (see section 2.3.3) with the following improvements:

a) A function to generate a Look-Up Table, GenLUT. This function first initializes an array of length LenLUT containing integers to zero. Loops are created to search through the H-matrix for non -1 entries and convert to decimals. Subsequent loops are created to compare all possible combinations of individual columns, taking the modulo two additions of each combination (equivalent to XOR operations) and converting same to decimals. The results in decimals are stored in a systematic form comprising a quotient and its remainder (i+iLen*j). A total number of $2^{n-k} = 256$ possible syndromes were considered. Finally, an equation cleverly inverts the addresses in the LUT and the entries in each address for decoding.

b) The decompression function has the LUT added to the other augments namely, Data1, Data3 and the Syndrome. Furthermore, equations are defined to correct one error and two errors accordingly.

c) The main program subsequently calls the GenLUT function and continues with the execution of other functions as described in the case of the Hamming code (see section 2.3.3).

2.4.2 Implementation on the [15, 7] BCH Code

An overview of the major steps taken in the implementation of coding and decoding the [15, 7] BCH code over the Slepian-Wolf channel is presented in Algorithm 2.2. All algorithms are coded in C/C++ language. The algorithm is similar to that in section 2.3.3 save for the introduction of a look-up table.

Algorithm 2.2: Algorithm for Coding and Decoding the [15, 7] BCH Code over the Slepian-Wolf Channel

Total-errors = 0

for *blocks transmitted = 1* **to** *maximum number of codewords transmitted in simulation* **do**

begin

- generate a length n random message using *rand* to determine the values of its individual bits for source1.
- convert the binary message from source1 to signal voltages, that is, bit 0 = -1.0V and bit 1 = +1.0V, to represent BPSK modulated signal transmission $\boldsymbol{s_j}$ for $0 \leq j \leq n-1$.
- generate random noise signal of length n to affect converted signal in each bit position using a typical uniformly distributed random number generator from C++ library.

$$sigma \times \sqrt{(-2 \times \log\left(\frac{(double)rand()}{rand\_max}\right)} \times \cos\left(2 \times m\_pi \times \frac{(double)rand()}{rand\_max}\right)$$

- add converted vector $\boldsymbol{s}$ to noise vector $\boldsymbol{N}$ to obtain the noisy version of the converted signal now, $\boldsymbol{X_j} = \boldsymbol{s_j} + \boldsymbol{N_j}$ for $0 \leq j \leq n-1$.
- quantize the noise signals of the noisy version by converting positive voltages to 1 and negative voltages to 0, we call these source2.
- compute the syndrome of the random bits of source2 by multiplication by the parity check matrix, these become the side information at the decoder.

- decompress the syndrome by performing an exclusive OR operation of the source2 and the syndrome. This is the same as the syndrome of the differences in the sources in straight forward coding.
- generate a **look-up table** with same dimension with the parity-check matrix of the [15, 7] code.
- do a one-to-one mapping of all the possible **256 error patterns** to the decompressed data. $\boldsymbol{LUT}[\boldsymbol{Tmp}[\boldsymbol{i}]] = \boldsymbol{i}$; ***Tmp*** becomes the address of ***LUT*** while its address ***i*** becomes the content of ***LUT.*** This mapping is used to decode the correct message sent.

- Repeat these procedures over SNRs from 0 to 10.

end

final error probability, P_e= *Total-errors / maximum number of codewords transmitted in simulation*

2.5 Summary

An overview of DSC and its major schemes has been discussed in this chapter with emphasis on Slepian-Wolf coding of two correlated binary sources. An efficient algorithm for correcting the single digit error of the [7, 4, 3] Hamming code was developed. Finally, the chapter concluded by extending the algorithm to correcting to bits of error of the [15, 7] BCH code. The performance achieved by the [15, 7] BCH is similar to that of the [7, 4, 3] Hamming code, (see 2.3.1, Figure 2.6). Hence, no need of repetition of the plot.

Chapter 3

Message-Passing (Iterative) Decoding using LDPC codes for DSC.

3.1 Introduction

This chapter introduces the iterative decoding of LDPC codes for DSC. If further discusses the step-by-step implementation of the belief propagation decoding algorithm and its modifications for the Slepian-Wolf channel. Two new channels were modelled: the AWGN and binary symmetric equivalent channels. Comparison is also made with the performance of other scheme with similar block length and code rate.

3.2 Low-Density Parity-Check Codes

LDPC codes are linear block codes defined by parity check matrices which contain mostly 0's and a very small number of 1's. It is because of this attribute of the parity check matrices, i.e. the relatively very small number of 1's compared to 0's, that they are described as *sparse* or *low-density*. These codes are most commonly decoded using the low-complexity probabilistic decoding method known as Belief Propagation (BP) or Sum-Product (SP) Algorithm. The main advantages of this decoding scheme lies in its simplicity as the computation per digit per iteration is independent of the code's block length. In addition, compared to other decoding schemes, it has a low computational time, very small memory requirements and generally lower implementation costs.

3.3 Message-Passing (Iterative) Decoding

Perhaps, the most significant difference between LDPC codes and classical block codes is the way they are decoded. Classical block codes are generally decoded with Maximum Likelihood (ML) like decoding algorithms and are therefore, usually short and designed algebraically to make the task less complex. Conversely, LDPC codes are decoded iteratively using a graphical representation of their parity-check matrix and thus, are designed with the properties of the parity-check matrix as a focus. Also known as message-passing decoding, iterative decoding operations basically involve the passing of messages along the edges of a Tanner graph. In other words, the messages are passed back and forward between the bit and check nodes iteratively until either a result is achieved, or the process is halted.

Different message-passing algorithms are named with respect to the type of messages passed or the type of operation performed at the nodes. For instance, in bit-flipping decoding, these messages are binary. This research is however, focused on belief propagation decoding, where the messages passed are probabilities that represent the levels of belief about the value of the codeword bits. Furthermore, it is often convenient to represent probability values as log likelihood ratios and decoding algorithms implemented thus are referred to as sum-product decoding algorithms.

3.4 Belief Propagation or Sum-Product Algorithm (BP/SPA) decoding implementation.

A Belief Propagation decoding is referred to as Sum-Product Algorithm (SPA) decoding because the use of log likelihood ratios allow for the calculations at the bit and check nodes to be computed using sum and product operations. One such algorithms presented in (Abdu-Aguye, Ambroze & Tomlinson, 2016b) and (Abdu-Aguye, Ambroze & Tomlinson, 2016a) which is implemented using the GNU/Linux based C programming language, is adopted and modified to take side information into consideration (see 3.4.1). The BP/SPA decoder reads the contents of a text file which contains an LDPC matrix. The first line of the configuration file contains important information about the LDPC matrix which is used for reading the parity check matrix correctly. These include the code length, i.e., the number of symbol bits in each block of the encoded message, the parity length, i.e., the number of parity-check bits in each block of the encoded message, the number of rows of parity-check equations in the configuration file, the signal-to-noise ratio, in decibels, at the receiver and the maximum number of decoding iterations to be executed in the Sum-Product Algorithm decoder.

The subsequent rows in the configuration file contain parity-check equations; from parity-check equation 0 to parity-check equation Z - 1 in sequence, where Z is the number of rows of parity-check equations in the configuration file. The serial numbers of all the symbol nodes involved in each parity-check equation, from parity-check equation 0 to parity-check equation Z - 1 in sequence, are listed in the corresponding row (from the second row of the configuration file which represents parity-check equation 0 to the Z-th row of the configuration file of the LDPC matrix configuration file. The symbol nodes in each parity-

check equation are also listed with one or two spaces placed between consecutive symbol node numbers. The numbers in each row indicate the position of the '1' bits in the corresponding row of the parity check matrix. A '-1' is used to terminate each row of parity check equations after all the symbol nodes in it have been completely listed.

An all-zero parity-check matrix with number of rows equal to *rows* and number of columns equal to the code length, *code length* as specified in the first line of the configuration file is created order to read the parity check matrix in the configuration file into memory. Subsequently, the positions of the '1s' in each line of the LDPC parity-check matrix are determined from the contents of the corresponding line of the parity-check equations in the configuration file and the changes from 0 to 1 are made in the corresponding positions in the all-zero parity-check matrix created in memory.

Going forward each time a '-1' is encountered it is interpreted as the end of that line of the LDPC matrix configuration file and the program proceeds to read the contents of the next line in the configuration file until it reaches the final row's '-1', when the number of rows of parity-check equations, *rows*, specified in the first line of the configuration file has been exhausted. At this point, it is determined that the LDPC matrix configuration file has been fully read into memory.

Subsequently, the program checks to ensure that the parity-check matrix (or *H* matrix) does not have any redundant rows or columns. A redundant row is a row with less than two symbol nodes. On the other hand, a redundant column is an all-zero column or a column without a single '1' bit; that is, the corresponding symbol node is not involved in any of the parity-check equations. If any these is found in the configuration file, a corresponding

error message is displayed, the redundant row or column number is specified, and the program terminates.

If no errors are found in the configuration file, the program proceeds to obtain the codeword generator matrix corresponding to the parity-check matrix read from the configuration file by performing the Gaussian elimination procedure on the parity-check matrix to convert it into its row echelon form. The row echelon parity-check matrix is further converted, using additional Gaussian elimination, to the reduced row echelon parity-check matrix in which the parity-check bits are expressed as explicit sums of information bits.

The presence of one or more all-zero rows in this reduced row echelon form of the parity-check matrix is an indication of the presence of linearly dependent parity-check equations. The configuration file must be manually edited to conform with the *parity length* specified in the first line of the parity-check matrix. If the parity length is less than the number of rows in the parity-check matrix, then the code rate is slightly higher than initially intended in the code design.

If the correct parity length of the code has been specified, at this stage the program proceeds to apply some elementary matrix manipulation techniques on the parity-check matrix in its Gaussian reduced row echelon form to arrive at the codeword generator matrix. The Gaussian H matrix is compared with $H = [A \mid I_{n-k}]$, where n is the length of the code and k is the information/message length, so that A is an $(n-k) \times k$ binary matrix and I_{n-k} is an identity matrix of order $n-k$, that is it is an $(n-k) \times (n-k)$ matrix.

The A part of the matrix is extracted from the reduced row echelon parity-check matrix and a matrix transpose operation is carried out on it to produce another matrix, A^T, which is a k

$\times$ *(n-k)* matrix. Next, an identity matrix, I_k, of order k is generated. These two matrices are merged to form a codeword generator matrix given by $G = [I_k \mid A^T]$. If column swapping operations were carried out in the process of obtaining the row echelon form of the parity-check matrix, the swapped columns would have to be unswapped in the reverse order in which they were originally swapped in order for the generator matrix to correspond to the original parity-check matrix read from the configuration file. It can be observed that the row space of the generator, G, is orthogonal to the original parity-check matrix, H. For any generator matrix, G, for a code with parity-check matrix H, the following relationship holds true; $GH^T = 0$.

The BP/SPA decoder program is designed to decode signals received over the Binary Input Additive White Gaussian Noise (BIAWGN) channel using BPSK modulation. Each transmitted 0 bit is assigned a voltage of magnitude -1.0V while each 1 bit is assigned +1.0V. For random variables, the SNR can be defined as $X = s + N$, where s is a constant signal and N is a random variable having an expected value of zero. The $SNR = s^2/\sigma_N^2$, where s^2 is the signal power's mean squared value, and σ_N^2 is the variance of N. σ_N^2 represents the noise power and is equal to its variance since the noise has zero mean. The signal-to-noise ratio, SNR, at which the iterative decoder carries out the code performance simulation, is given in decibels and is easily converted to linear SNR from the relationship, $SNR\ (dB) = 10log_{10}\ SNR$. Thus, $SNR = 10^{(SNR\ (dB)\ /\ 10)}$.

Consequently, with $s = \pm 1.0$V, $s^2 = 1$, which also implies that $SNR = 1 / \sigma_N^2$. Solving for σ_N, we obtain $\sigma_N = 1 / \sqrt{SNR}$. The standard deviation of the noise, σ_N, is used to determine the amplitude of the AWGN influence on the received signal in commensurate measure to

the SNR of signal reception. All transmitted messages/code words have the same messages energies.

Two functions obtained from (Seiler & Seiler, 1989) are utilized to properly characterize the effect of noise at any given signal-to-noise ratio on the sequence of bits transmitted through the BIAWGN channel using BPSK modulation. These are the *ran2* and the *gasdev* functions. The *ran2* function is a random number generator which returns a uniform random deviate between 0.0 and 1.0 every time the function is called. Within the limits of its floating-point precision, *ran2* provides perfect random numbers. The *gasdev* function returns a normally distributed deviate with zero mean and unit variance, using the *ran2* function as a source of uniform deviates for its operation. The *gasdev* function perfectly simulates the influence of additive white noise with a Gaussian distribution which is used to affect the BPSK modulated signals. Algorithm 2.1 summarized the algorithm of the core part of the software designed for Monte Carlos simulations for performance evaluation of PEG LDPC codes constructed in this study. It entails the repetitive aspects of the program execution including the BP/SPA decoding algorithm and excludes the aspects of the program which are executed only once and have already been explained in the preceding text. An illustration of this procedure is also presented in Figure 3.1.

3.4.1 Implementation of BP/SPA using LDPC Codes.

An overview of the major steps taken in the implementation of BP/SPA using LDPC codes over the conventional channel is presented in Algorithm 3.1. LDPC codes of block lengths, $n = 512$ and $n = 1024$ are used in this section. All algorithms are coded in C/C++ language.

Algorithm 3.1: Algorithm for LDPC Code Performance Simulations using BPSK Modulation over the AWGN Channel and BP/SPA Iterative Decoding.

Total-errors = 0

for *blocks transmitted = 1* **to** *maximum number of codewords transmitted in simulation* **do**

begin

- generate a length k random message using *ran2* to determine the values of its individual bits, where message length (k) = code length (n) – parity length (p)
- multiply the random message with the codeword generator matrix to obtain the binary codeword corresponding to the message.
- convert the binary codeword to the transmitted codeword signal, that is, bit 0 = -1.0V and bit 1 = +1.0V, to represent BPSK modulated signal transmission $\boldsymbol{s_j}$ for $0 \leq j \leq n-1$.
- generate Additive White Gaussian Noise signal of length n to affect transmitted signal in each bit position using; $\boldsymbol{N_j = \sigma_N \times gasdev}$, for $0 \leq j \leq n-1$ and a new function call is made to *gasdev* for all j.
- add transmitted vector $\boldsymbol{s}$ to noise vector $\boldsymbol{N}$ to obtain the received signal, $\boldsymbol{X_j = s_j + N_j}$ for $0 \leq j \leq n-1$.
- calculate the received probabilities, $P(1)_j$, for all the bits in the received signal to obtain a received probability matrix, say $R_{P(1)j}$, for $0 \leq j \leq n-1$.
- create and initialize a probability matrix, H_P, from the original parity-check matrix, H, in the configuration file such that bit $0 \rightarrow 0.00$ and bit $1 \rightarrow 0.50$.

for *iterations = 1* to *maximum number of iterations* **do**

begin

execute the sum-product algorithm decoding on the received probabilities using the probability matrix, H_P as a workspace.

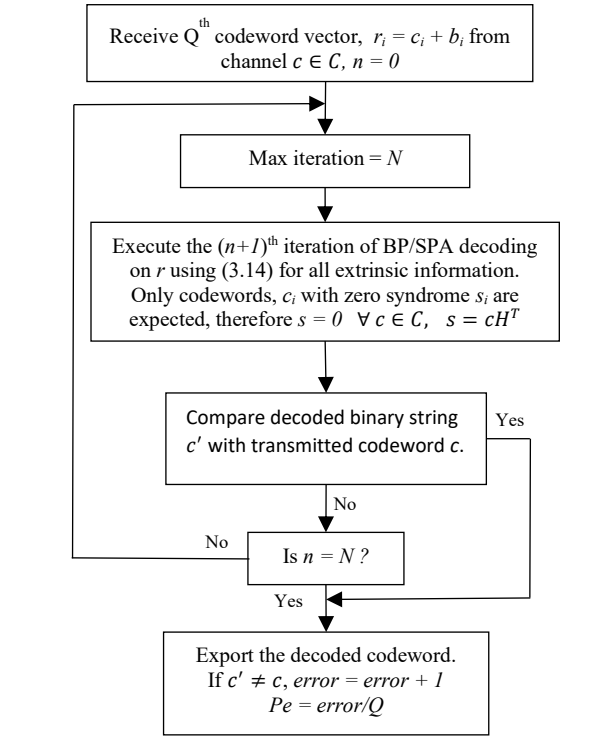
end

- convert the final bit value probabilities, $P(1)_j$, into bits, using **if** $P(1)_j \geq 0.5$ **then** $s_j = 1$, **else** $s_j = 0$, $\forall\ 0 \leq j \leq n-1$.
- compare the bit values of the decoded output to the original message codeword transmitted, if they are the same then *error* = 0, and if they are different *error* = 1.
- if *error* = 1, increment the *Total-errors* count.
- calculate and display the block error rate, P_e, at regular intervals. This is the

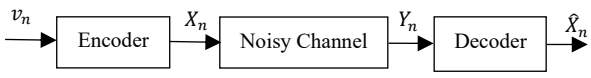
ratio of the present number of unsuccessfully decoded codewords to the present total number of codewords processed by the sum-product algorithm decoder.

end

final error probability, P_e= *Total-errors* / *maximum number of codewords transmitted in simulation*



(a)



(b)

Figure 3.1: The conventional decoder (a) workflow (b) block diagram.

3.5 Modifications for Slepian-Wolf Decoding

SW coding involves two or more sources sending similar information to an encoder simultaneously. Common information from the sources is carefully managed to avoid unnecessary duplication and obtain optimal usage of channel capacity. One method of using error-correcting codes to obtain data compression is syndrome source coding (Ancheta, 1976). The source sequence is treated as an error pattern whose syndrome forms the compressed data.

3.5.1 The Slepian-Wolf AWGN Equivalent Channel

The Classical Communication Channel code which has been described above is modified to implement a Slepian-Wolf AWGN channel. Here, the focus is on the major modifications made, as the original channel has been described in section 3.4. From the onset, the entire block length consists of randomly generated binary digits as against the former case, where, for a rate half code, half of the block length consist of message bits while the other half is parity bits. In other words, there are no codewords per se, rather there exist, random bits of length n.

The random bits so generated is denoted as *source1*. One way of defining the correlation between the sources is by making the second source (*source2*) a noisy version of the *source1*. To achieve this, additive white Gaussian noise is added to the random bits from the first source to form the second source. Then, the syndrome of *source1* is obtained by

multiplication with the H-matrix, this being just half in length of the original source. Finally, the syndrome of *source1* and the noisy version (*source2*) are transmitted separately over a noiseless channel to the decoder. It could be observed that compression is achieved by sending the syndrome of one source instead of the original source which is twice as long as its syndrome. And according to Slepian and Wolf, these sources can be faithfully reconstructed at the decoder.

Another major modification to the classical channel is with regards to the syndromes. In the classical channel, the syndrome necessarily has to be zero to ensure that a codeword was sent and received. But, as earlier highlighted, technically there are no codewords in the case of the Slepian-Wolf channel. Hence, there exist syndromes which are a combination of zeros and ones. These syndromes are used as side information at the decoder to obtain the messages that have been transmitted at any particular instant. Simplification of Gallager's equations leads to instances where the calculation of the probability of a message bit received to be a one. For the purpose of illustration, we consider the $k = 2, n = 3$ even parity check code. However, the discussion is similar for codes of any length. Assume the values, r_i were received and the probabilities of the received vectors are computed from (3.1).

$$p_i^c = P(c_i = c|r_i) = ae^{-\frac{(r_i - v^c)^2}{2\sigma^2}} \tag{3.1}$$

where a is a constant, $v^0 = -1.0$, $v^1 = +1.0$ and SNR $= 1/\sigma^2$. i is the bit position and c is the bit value, $i = 0, 1, 2$. The probability of each codeword is computed as shown in Table 1.

Table 1: MAP decoding of the SPC by codeword enumeration.

$c_0c_1c_2$	$P(c \mid r)$
000	$p_0^0 p_1^0 p_2^0$
011	$p_0^0 p_1^1 p_2^1$
101	$p_0^1 p_1^0 p_2^1$
110	$p_0^1 p_1^1 p_2^0$

To determine the probability of output bit c_i being c, we sum the probabilities of all the codewords for which $c_i = c$. For example, the output probability of c_1 being 1 is:

$$p_1^{1o} = p_0^0 p_1^1 p_2^1 + p_0^1 p_1^1 p_2^0 = p_1^1 (p_0^0 p_2^1 + p_0^1 p_2^0) \tag{3.2}$$

The quantity p_1^1 is known as intrinsic information and was known before decoding the Single Parity Check (SPC) code. The quantity:

$$p_1^{1e} = p_0^0 p_2^1 + p_0^1 p_2^0 \tag{3.3}$$

is known as extrinsic information and was produced by decoding the SPC code. It can be observed that the extrinsic information for bit i to be 0, p_i^{0e} is the sum of all products of the other bit probabilities for which the sum of the bits is even. Also, the extrinsic information for bit i to be 1, p_i^{1e} is the sum of all products of the other bit probabilities for which the bit is odd.

Gallager observed that if we calculate, for example, for bit $i = 1$, we obtain the following products:

$$(p_0^0 + p_0^1)(p_2^0 + p_2^1) = 1 \ = p_0^0 p_2^0 + p_0^0 p_2^1 + p_0^1 p_2^0 + p_0^1 p_2^1 \tag{3.4}$$

$$(p_0^0 - p_0^1)(p_2^0 - p_2^1) = (1 - 2p_0^1)(1 - 2p_2^1) \ = p_0^0 p_2^0 - p_0^0 p_2^1 - p_0^1 p_2^0 + p_0^1 p_2^1 \tag{3.5}$$

Adding (3.5) to (3.4), we obtain twice the sum of all even terms, that is $2p_1^{0e}$. However, subtracting (3.5) from (3.4) gives the sum of all odd terms, that is $2p_1^{1e}$. In equation form:

$$p_1^{0e} = \frac{1 + (1 - 2p_0^1)(1 - 2p_2^1)}{2} = p_0^0 p_2^0 + p_0^1 p_2^1 \tag{3.6}$$

$$p_1^{1e} = \frac{1 - (1 - 2p_0^1)(1 - 2p_2^1)}{2} = p_0^1 p_2^0 + p_0^1 p_2^0 \tag{3.7}$$

In another example,

$$p_2^{1e} = \frac{1 - (1 - 2p_0^1)(1 - 2p_1^1)}{2} \tag{3.8}$$

And in general,

$$p_i^{0e} = \frac{1 + \prod_{j \neq i}(1 - 2p_j^1)}{2} \tag{3.9}$$

$$p_i^{1e} = \frac{1 - \prod_{j \neq i}(1 - 2p_j^1)}{2} \tag{3.10}$$

In the classical channel, the syndrome should always be zero, thus the conversion of received signals to probabilities is done using (3.10). However, we introduce an additional computation for the cases where syndromes in the Slepian-Wolf channel are a combination of odd and even syndromes. Hence, we implement a second formula (3.9) for cases where the syndrome is a one (or odd).

3.5.2 Implementation of BP/SPA for SW AWGN-equivalent Channel.

An overview of the major steps taken in the implementation of BP/SPA using LDPC codes over the Slepian-Wolf AWGN-equivalent channel is presented in Algorithm 3.2. LDPC codes of block lengths, $n = 512$ and $n = 1024$ are used in this section. This summarizes the core part of the software designed for Monte Carlo simulations for performance evaluation of PEG LDPC codes constructed for the Slepian-Wolf AWGN channel. An illustration of this procedure is also presented in Figure 3.2.

Algorithm 3.2: Algorithm for LDPC Code Performance Simulations using BPSK Modulation over the Slepian-Wolf AWGN Equivalent Channel and BP/SPA Iterative Decoding

Total-errors = 0.

for *blocks transmitted = 1* **to** *maximum number of codewords transmitted in simulation* **do**

begin

- generate a length *n* random message using *ran2* to determine the values of its individual bits for source1.
- convert the binary message from source1 to signal voltages, that is, bit 0 = -1.0V and bit 1 = +1.0V, to represent BPSK modulated signal transmission $\boldsymbol{s_j}$ for $0 \leq j \leq n - 1$.
- generate Additive White Gaussian Noise signal of length *n* to affect converted signal in each bit position using; $\boldsymbol{N_j} = \boldsymbol{\sigma_N} \times \boldsymbol{gasdev}$, for $0 \leq j \leq n - 1$ and a new function call is made to *gasdev* for all *j*.
- add converted vector $\boldsymbol{s}$ to noise vector $\boldsymbol{N}$ to obtain the noisy version of the converted signal now called source2, $\boldsymbol{X_j} = \boldsymbol{s_j} + \boldsymbol{N_j}$ for $0 \leq j \leq n - 1$.
- compute the syndrome of the random bits of sources1 by multiplication by the parity check matrix, these become the side information at the decoder.

- calculate the received probabilities, $P(1)_j$, for all the signal received from source2, according to the corresponding syndrome (either odd or even). A received probability matrix, say $R_{P(1)j}$, for $0 \leq j \leq n-1$ is hereby obtained.
- create and initialize a probability matrix, H_P, from the original parity-check matrix, H, in the configuration file such that bit $0 \rightarrow 0.00$ and bit $1 \rightarrow 0.50$.

for *iterations = 1* to *maximum number of iterations* **do**

begin

execute the sum-product algorithm decoding on the received probabilities using the probability matrix, H_P as a workspace.

end

- convert the final bit value probabilities, $P(1)_j$, into bits, using **if** $P(1)_j \geq 0.5$ **then** $s_j = 1$, **else** $s_j = 0$, $\forall\ 0 \leq j \leq n-1$.
- compare the bit values of the decoded output to the original message of source1, if they are the same then *error* = 0, and if they are different *error* = 1.
- if *error* = 1, increment the *Total-errors* count.
- calculate and display the block error rate, P_e, at regular intervals. The block error rate, P_e, is the ratio of the present number of unsuccessfully decoded codewords to the present total number of codewords processed by the sum-product algorithm decoder.

end

final error probability, P_e= *Total-errors* / *maximum number of codewords transmitted in simulation.*

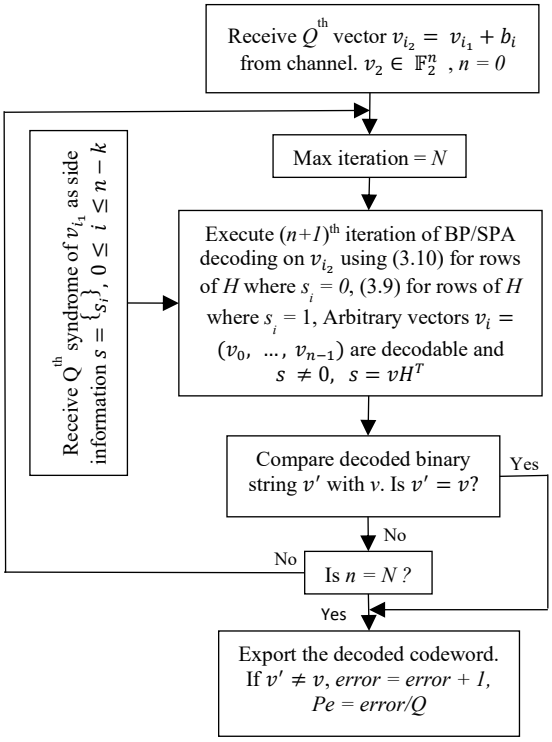


Figure 3.2: The proposed SW decoder (a) workflow (b) block diagram.

3.5.3 The Slepian-Wolf Binary Symmetric Equivalent Channel

The BSC is a binary channel in which the input symbols are complemented with probability p. It is the simplest model of a channel with errors and captures most of the complexity of the general problem. When an error occurs, a 0 is received as a 1 and vice versa.

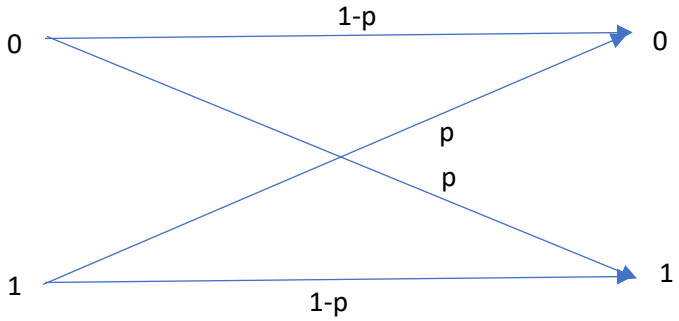


Figure 3.3: The Binary Symmetric Channel.

At the initial stage, the decoder is designed with the correlation channel being an AWGN channel. This implied that one source consists of bit strings while the other consist of real numbers representing the voltages transmitted over the channel after addition of noise. These voltage values are converted into probabilities in the decoder. However, the set back of this arrangement is the deficiency in defining the correlation between the sources. An ideal justification for the correlation between the source would require both sources to be in the same form. Therefore, we proceed by implementing the correlation channel as an equivalent BSC. Consequently, the received vector probabilities are replaced with the crossover probability, p of the BSC.

$$p = Q(\sqrt{SNR}) \tag{3.11}$$

3.5.4 Implementation of BP/SPA for SW BSC-equivalent channel.

An overview of the major steps taken in the implementation of BP/SPA using LDPC codes over the Slepian-Wolf BSC-equivalent channel is presented in Algorithm 3.3. LDPC codes of block lengths, $n = 512$ and $n = 1024$ are used in this section. It summarises the core

part of the software designed for Monte Carlo simulations for performance evaluation of PEG LDPC codes constructed for the Slepian-Wolf BSC equivalent channel. A major difference between the convention and proposed SW decoder is the availability of side information at the decoder in the later. Moreover, syndromes are calculated for each random bit sequence generated in the proposed model as opposed to the conventional case where the syndrome is only computed in the event of failure in decoding a codeword correctly. This leads a more complex computational overhead in the proposed model. Furthermore, although AWGN is added to obtain the second source in the proposed model, both sources are transmitted noiselessly to the joint decoder while the conventional channel is usually noisy, hence the need for error correction by forming codewords. Finally, there are no codewords present in the proposed model but a sequence of randomly generated bit strings of data.

Algorithm 3.3: Algorithm for LDPC Code Performance Simulations using BPSK Modulation over the Slepian-Wolf Binary Symmetric Equivalent Channel and BP/SPA Iterative Decoding

Total-errors = 0.

for *blocks transmitted = 1* **to** *maximum number of codewords transmitted in simulation* **do**

begin

- generate a length n random message using *ran2* to determine the values of its individual bits for source1.
- convert the binary message from source1 to signal voltages, that is, bit 0 = -1.0V and bit 1 = +1.0V, to represent BPSK modulated signal transmission $\boldsymbol{s_j}$ for $0 \leq j \leq n-1$.
- generate Additive White Gaussian Noise signal of length n to affect converted signal in each bit position using; $\boldsymbol{N_j = \sigma_N \times gasdev}$, for $0 \leq j \leq n-1$ and a new function call is made to *gasdev* for all j.
- add converted vector $\boldsymbol{s}$ to noise vector $\boldsymbol{N}$ to obtain the noisy version of the converted signal now, $\boldsymbol{X_j = s_j + N_j}$ for $0 \leq j \leq n-1$.

- threshold the voltage signals of the noisy version by converting positive voltages to 1 and negative voltages to 0, we call these source2.
- compute the syndrome of the random bits of sources1 by multiplication by the parity check matrix, these become the side information at the decoder.
- using equation 2.3, calculate the received probabilities, $P(1)_j$, for the signal received from source2, according to the corresponding syndrome (either odd or even). A received probability matrix, say $R_{P(1)j}$, for $0 \leq j \leq n-1$ is hereby obtained.
- create and initialize a probability matrix, H_P, from the original parity-check matrix, H, in the configuration file such that bit $0 \rightarrow 0.00$ and bit $1 \rightarrow 0.50$.

for *iterations = 1* to *maximum number of iterations* **do**
begin
execute the sum-product algorithm decoding on the received probabilities using the probability matrix, H_P as a workspace.
end

- convert the final bit value probabilities, $P(1)_j$, into bits, using **if** $P(1)_j \geq 0.5$ **then** $s_j = 1$, **else** $s_j = 0$, $\forall\ 0 \leq j \leq n-1$.
- compare the bit values of the decoded output to the original message of source1, if they are the same then *error* = 0, and if they are different *error* = 1.
- if *error* = 1, increment the *Total-errors* count.
- calculate and display the block error rate, P_e, at regular intervals. The block error rate, P_e, is the ratio of the present number of unsuccessfully decoded codewords to the present total number of codewords processed by the sum-product algorithm decoder.

end

final error probability, P_e= *Total-errors* / *maximum number of codewords transmitted in simulation*

3.6 Performance of LDPC Codes

LDPC codes are most commonly decoded using the low-complexity probabilistic decoding method known as Belief Propagation (BP) or Sum-Product (SP) Algorithm. The main advantages of this decoding scheme lies in its simplicity as the computation per digit per iteration is independent of the code's block length. It also has a low computational time, very small memory requirements and generally lower implementation costs compared to

other decoding schemes, for instance, building a look-up table (see section 2.4.1). The disadvantage of the BP/SPA decoding is that it is not optimal in the artificial sense of minimising the probability of decoding error. However, for most practical implementations, its simplicity adequately compensates for this perceived sub-optimality. As a result of their capacity-approaching performance using low-complexity iterative decoding, error correction code design based on graph theory, mainly LDPC codes, have continued to attract a lot of research effort.

What follows are plots of results obtained from simulation of LDPC codes on the conventional AWGN channel, the Slepian-Wolf AWGN equivalent channel as well as the Slepian-Wolf binary symmetric equivalent channel. These simulations are run for block lengths of 1024 and 512. Also, comparisons are made between the three named models. The following observations were made:

a. The conventional AWGN model in which all syndromes are zero is a sub-set of the Slepian-Wolf AWGN equivalent model which contains both odd and even syndromes. Hence, the very close performance of these two models.
b. As expected, the performance of the Slepian-Wolf binary symmetric equivalent model lags behind that of its AWGN counterpart as these codes were originally optimized for the AWGN channel.
c. Even with the sub-optimal performance of the Slepian-Wolf binary symmetric equivalent channel, the model still competes very closely with models obtained in (Sartipi & Fekri, 2005), which are of similar block length and code rate.

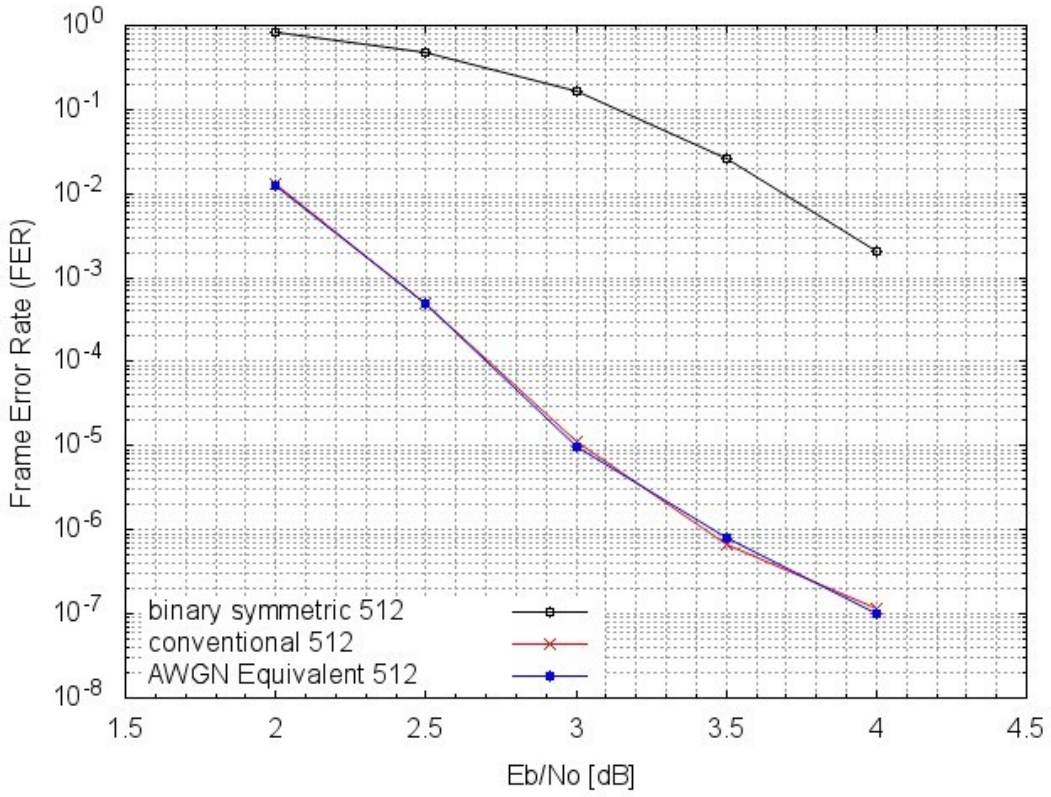


Figure 3.4: Plot of FER against E_b/N_o for conventional and Slepian-Wolf channels (n = 512).

Figure 3.4 is a plot of the frame error rate (FER) against the E_b/N_o ratio of the conventional and the Slepian-Wolf AWGN equivalent channels for LDPC codes of block length, $n = 512$. The Slepian-Wolf binary symmetric equivalent channel is only super-imposed to show the relative performance of the models. For the conventional (red plot) and AWGN equivalent (blue plot) schemes, it defines an operating point at E_b/N_o equals 4 dB of 1×10^{-7} frame error rate. This implies that, at 4 dB, only one frame was recorded in error out of 10,000,000 frames that was transmitted at the particular E_b/N_o. In contrast, the binary symmetric equivalent channel (black plot) recorded a frame in error after only 1,000 frames were transmitted. In general, the lower the curve, the better the performance at any particular E_b/N_o.

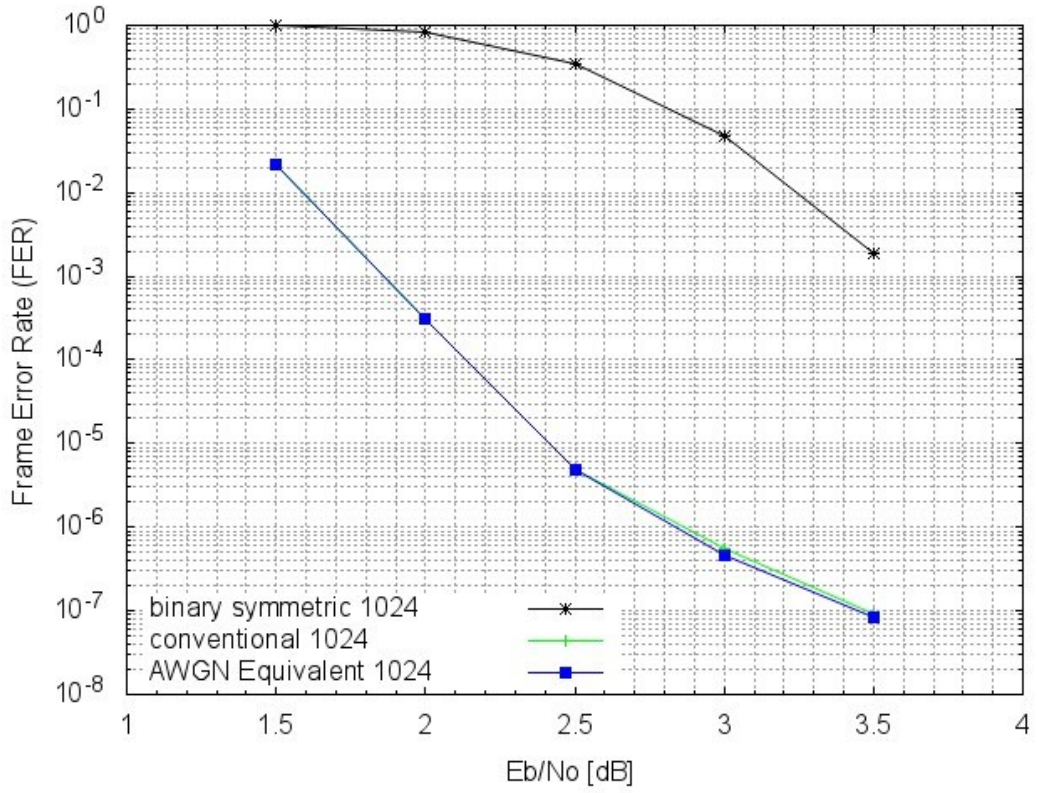


Figure 3.5: Plot of FER against E_b/N_o for conventional and Slepian-Wolf channels (n = 1024).

It is understood from Shannon's theory that the performance of codes generally improves with code length. Thus, the error probability tends to zero as the block length tends to infinity. For this reason, simulation are repeated with LDPC codes of twice the block length ($n = 1024$) and the results are presented in Figure 3.5. For the conventional channel (green plot) and the AWGN equivalent channel (blue plot), it can be observed that, at a lower E_b/N_o of 3.5 dB, only one frame was received in error out of 10,000,000 frames that were transmitted. This same result was achieved at E_b/N_o of 4 dB when the block length was 512 (see Figure 3.4). There is also a corresponding improvement of the binary symmetric equivalent channel in Figure 3.5 accordingly.

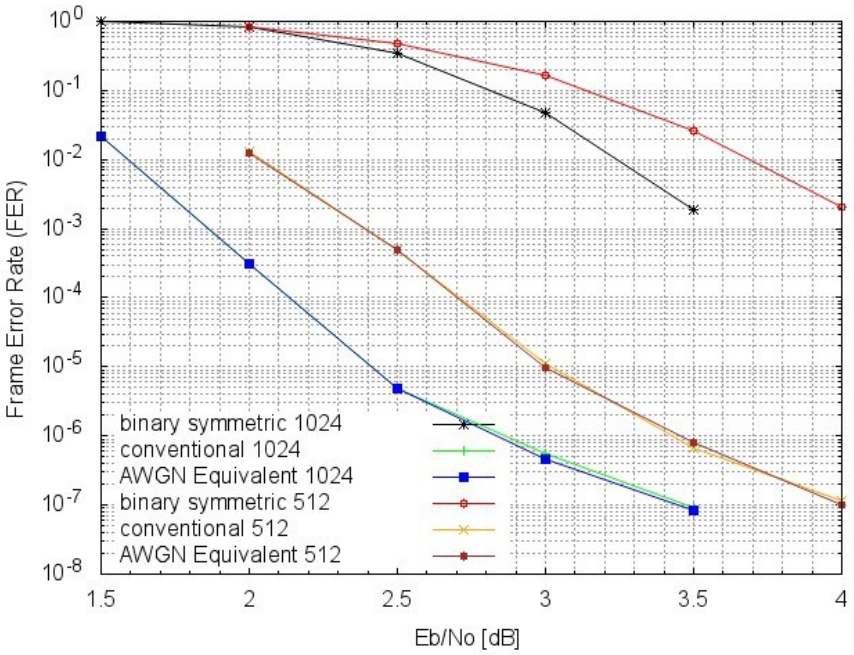


Figure 3.6: Plot of FER against E_b/N_o for Conventional and Slepian-Wolf Channels (n = 512, 1024)

Figure 3.6 is a superimposition of Figure 3.4 and Figure 3.5 in order to have an overview of all three models over block lengths, n = 512 and n = 1024. Generally, the lower curves show a better error performance than the higher ones.

However, in literature, the standard for showing the performance of the Slepian-Wolf binary symmetric equivalent models is to plot the BER against the joint entropy of the sources as presented in Figure 3.7. Even though it has been shown that the AWGN equivalent channels performs better than the binary symmetric equivalent channels, the correlation of the former is not an ideal correlation because one source is binary digits while the other consist of voltages in decimals. In a bid to properly justify the correlation between the sources, the existing AWGN equivalent channel is replaced with the binary symmetric equivalent channel, effectively using the crossover probabilities for the formation of the received probabilities matrices.

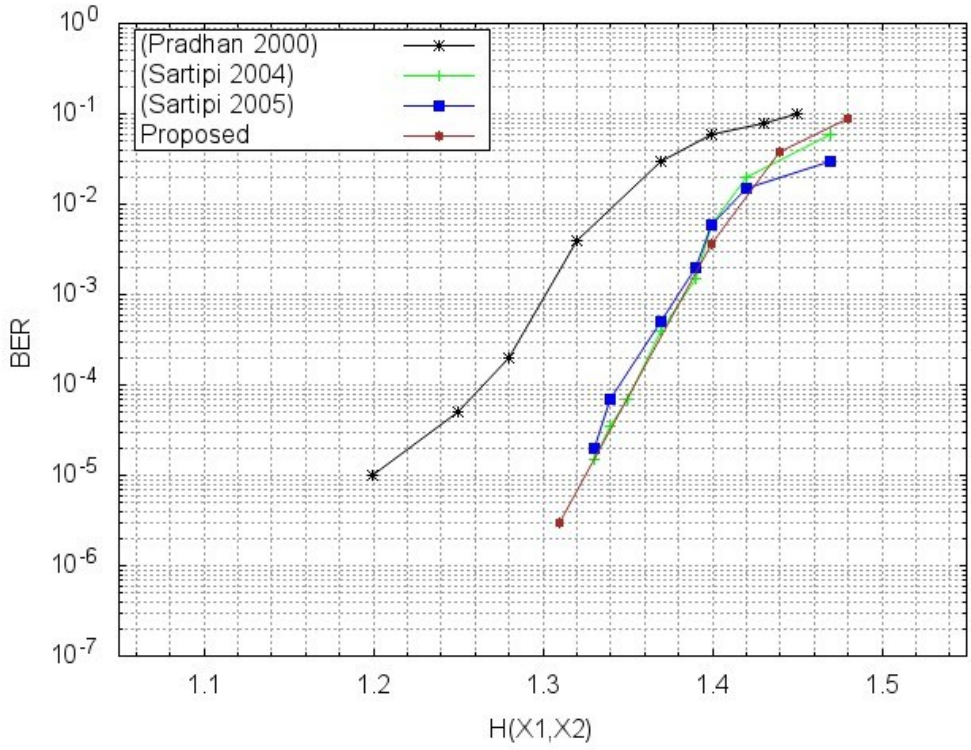


Figure 3.7: Plot of BER against Joint Entropy (n = 1000)

Figure 3.7 shows the results of the bit error rate averaged of the proposed model over the two sources *X1* and *X2* as a function of the joint entropy for asymmetric rates alongside with asymmetric and symmetric rate models in (Sartipi & Fekri, 2005), (Pradhan & Ramchandran, 2000) and (Sartipi & Fekri, 2004). All four models have a block length, n = 1000 and code rate of half. The Slepian-Wolf theoretical limit for this scheme is 1.5 bits for ideal channels. Thus, the curves closest to the H(X1, X2) = 1.5 line are the best in terms of performance. It can be observed that for the most part, the proposed model (brown plot) trails that of (Sartipi & Fekri, 2005) {blue plot} and (Sartipi & Fekri, 2004) {green plot} until around H(X1, X2) = 1.42, at which point, there no bits received in error out of 100 bits transmitted. (Pradhan & Ramchandran, 2000) however lags behind compared to others and thus has the least performance.

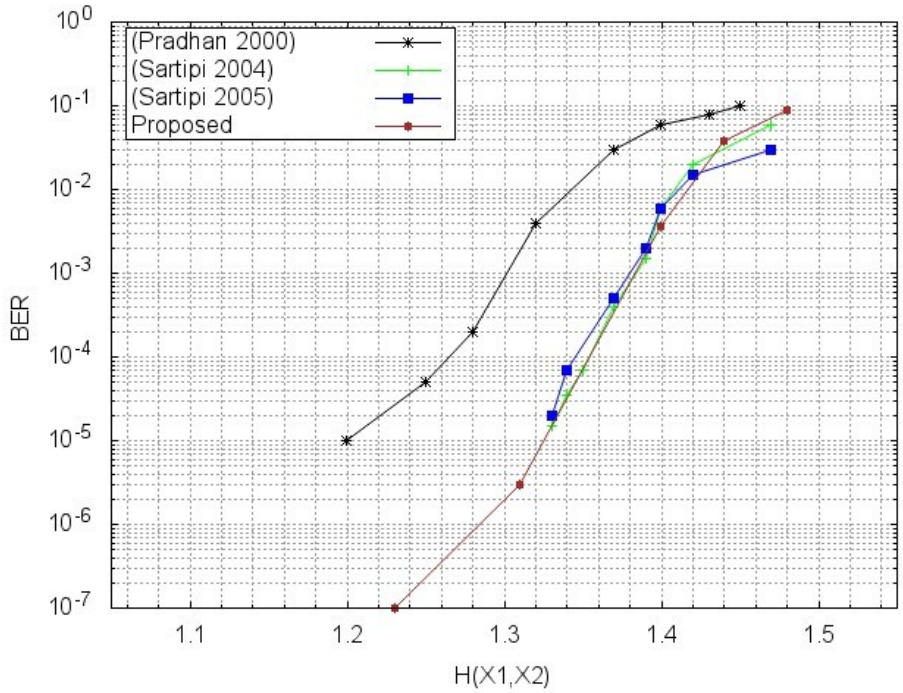


Figure 3.8: Distributed source coding using LDPC codes of block length, n = 1000 at symmetric and asymmetric rates.

The key objective of digital audiovisual coding technologies is to compress the original audiovisual information into a much smaller number of bits without adversely affecting the decoded signal quality and following a set of requirements depending on the target application. Consequently, and in contrast to previous works, the proposed model has been extended (up to H(X1, X2) = 1.25) to reflect the error floor performance as presented in Figure 3.8. This would be necessary in applications requiring such level of operation, for instance, in distributed video coding, watermarking and dirty-paper applications.

3.7 Summary

A scheme for distributed source coding to correct multiple errors, using LDPC codes to compress close to the Slepian-Wolf limit for correlated binary sources has been developed. Two novel equivalent channels were modelled, namely the additive white Gaussian noise equivalent channel and the binary symmetric equivalent channel. The performance achieved is seen to be better than previously published schemes for similar block length and code rate.

As an initial step to modifying the conventional decoder to the proposed SW decoder, the syndrome concept is incorporated to serve as side information available at the decoder only. One source is transmitted at full rate without any compression whatsoever and the other source is transmitted at half rate (for the half-rate code); the syndrome is half the block length of the code. Although, the correlation between the sources at this stage is not ideal, a very good compression was achieved with no apparent loss in performance compared to the conventional decoder. The same set of codes were run of both models and Figure 3.4, Figure 3.5 and Figure 3.6 show that this model has more or less the same performance as that of the conventional channel.

A framework for the complexity associated with the proposed binary symmetric equivalent channel is established. It is shown that although the proposed scheme seems to have additional computational overhead due to the formation of syndromes each time random bit sequences are generated; this overhead is offset by the absence of encoding operation all together in the proposed scheme.

Chapter 4

Challenges to improving the performance of short-length codes for real-time applications.

4.1 Introduction

This chapter highlights the challenges of improving the performance of short-length codes for real-time applications. It further discusses QR codes, the Dorsch decoder and its modifications for DSC. It concludes with the complexity associated with it the implementation of the various schemes.

It is understood from Shannon that block codes of sufficiently large length are asymptotically optimal. This justifies the use of binary channel codes like LDPC and Turbo Codes for practical Slepian-Wolf coding as highlighted in chapter three. However, for practical systems and in real-time applications, where delay and complexity limitations are stringent, long length LDPC and Turbo Codes are not very useful. Consequently, the performance of DSC systems based on long block length codes is highly impaired. This motivates our choice of a short-length, powerful, low-delay and very efficient QR codes.

4.2 Quadratic Residue (QR) codes

The quadratic-residue (QR) codes are cyclic codes of prime block length, p over a field GF(l), where l is another prime which is a quadratic residue modulo p (MacWilliams & Sloane, 1977). Examples of quadratic-residue codes are the binary [7, 4, 3] Hamming code (treated in chapter 2), the binary [23, 12, 7] and the ternary [11, 6, 5] perfect Golay codes. In the important case of binary quadratic residue codes ($l = 2$), the block length, p has to

be a prime of the form $(8m \pm 1)$. The binary QR codes introduced by Prange (Prange, 1985) are a family of cyclic BCH codes with code rates greater than or equal to one-half and generally tend to have large minimum distances, at least if the block length is not too large (MacWilliams & Sloane, 1977).

Because we are particularly interested in block codes of short code lengths, there are eleven binary codes with code length less than 100. These are codes with block lengths; 7, 17, 23, 31, 41, 47, 71, 73, 79, 89 and 97. Although, our algorithm is applicable to any of these, this work focuses on the [47, 24, 11] code with generator polynomial $x^{23} + x^{19} + x^{18} + x^{14} + x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^6 + x^5 + x^3 + x^2 + x + 1$.

There has been a variety of decoding methods (Chen *et al.*, 1994; Dubney *et al.*, 2009; Reed *et al.*, 1992) developed to decode QR codes which involve algebraic decoding techniques that require many complicated computations in a finite field. These also lead to a time delay and becomes unrealistic for real-time applications.

4.3 The Dorsch Decoder

This decoder was initially introduced by Dorsch (Dorsch, 1974) in 1974. It was designed for linear binary block codes using soft decisions quantised to J levels, although applicable to any linear block code. Particularly, it does not rely upon any features of the code such as being a concatenated code or having a sparse parity-check matrix. The decoder also features hard decisions being derived from the soft decisions using standard bit by bit detection, choosing the binary state closest to the received coordinate.

More so, codes with relatively high minimum distances generally tend to improve the overall performance of DSC systems. On the other hand, LDPC codes are largely decoded iteratively, which is not well suited for codes with high minimum distances. Consequently, attempts have been made (Tomlinson, Tjhai & Ambroze, 2007; Tomlinson *et al.*, 2017) to address this constraint by modifying the Dorsch decoder such that the most likely codewords were derived from a partial correlation function of low information weight codewords and which leads to an efficient fast decoding. Specifically, works have also been published on the number of erasures that are correctable by a linear code (Tomlinson *et al.*, 2007) and how to decode serial concatenated codes using erasure patterns (Tomlinson & Ambroze, 2010).

4.4 Implementation of Dorsch decoding

Aside the difference in the parity-check matrices used, the encoding parts of this implementation is largely same as done for the Hamming code (section 2.3.3), BCH code (section 2.4.2) and LDPC codes (section 3.4.1). This section discusses the entire encoding and decoding operations and highlights the modifications made on the decoding part. The Dorsch decoder entails producing a new parity-check matrix by re-ordering the columns of the original parity-check matrix according to likelihood of each candidate codeword. An illustration of the original parity-check matrix of the [47, 24, 11] QR code is shown in Figure 4.1. This parity-check matrix is initially saved as a text file from where it is read from within the C++ program. Also, along with the parity-check matrix, the code length, number of parity-check bits, number of rows, SNR and number of iterations are also read from the text file. These can be changed according to the parity-check equation in question.

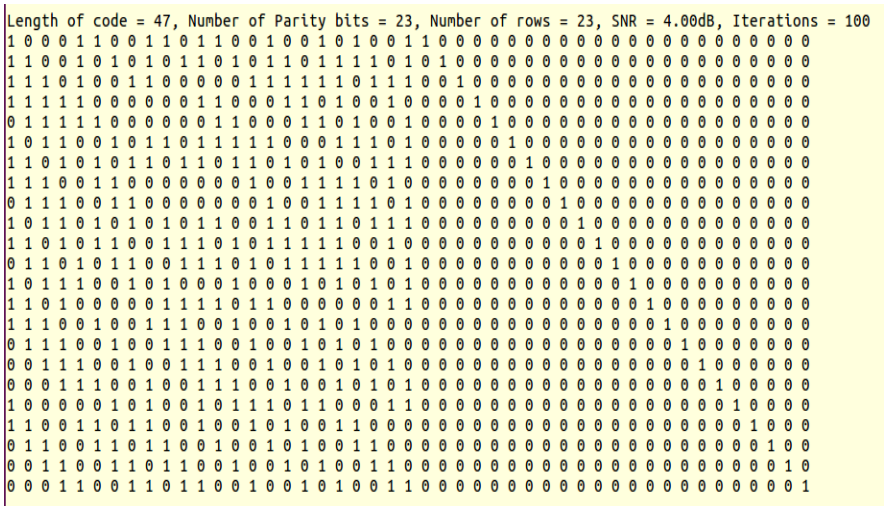


Figure 4.1: An illustration of the parity-check matrix of the [47, 24, 11] QR code.

The resulting parity-check matrix is then reduced to echelon canonical form by elementary row operations followed by the evaluation of several candidate codewords and the codeword with the minimum soft decision metric is output from the decoder.

The algorithm is based on the premise that most codes, on average can correct almost $(n - k)$ erasures. Meanwhile, the guaranteed number of correctable erasures is $(d - 1)$ and the guaranteed number of correctable hard decision errors is $(d - 1)/2$. However, correction of any combination of $n - k$ erasures is only possible for Maximum Distance Separable (MDS) codes. This led to the development of the Modified Dorsch Decoder (Tomlinson, Tjhai & Ambroze, 2007) that solves $(n - k)$ erasures for a non-MDS code. This modified decoder uses alternative columns of the parity-check matrix without the need for column permutations and was unnecessary to keep calculating each candidate codeword and its associated soft decision metric in order to find the most likely codeword.

Furthermore, an incremental correlation approach is adopted which features low information weight codewords and a correlation function involving only a small number

of coordinates of the received vector. Maximum likelihood is realised provided all codewords are evaluated up to the bounded information weight.

The decoder lends itself to a low complexity, parallel implementation involving a concatenation of hard and soft decision decoding. Near maximum likelihood is achieved for codes as long as 1000 bits provided the code rate is high enough.

The correlation between two sources is achieved by adding AWGN to the randomly generated bits of one source, *X* to form the other source, *Y,* with unity noise variance, σ^2. Subsequently, the syndrome of source, *X* is obtained by multiplication with the parity-check matrix of the code. Finally, the syndrome of source, *X* and the correlated source, *Y* are transmitted separately over a noiseless channel to the decoder. Because the block-length of any information source is larger than the length of its syndrome, sending the syndrome in place of its original ensures compression.

To implement this, random bits of length, $(n = 47)$ are generated using the *ran2* function. This is the first source. The random bits are converted to voltages and additive white Gaussian noise with variance, σ^2 is added to obtain the second source.

The amount of white noise added is a function of the signal-to-noise ratio. Because the guaranteed number of correctable hard decision errors is $(d - 1)/2$, thus the [47, 24, 11] QR code can only correct up to 5 bits of error. Several random bits were generated, and varying amount of white noise added.

Subsequently, Figure 4.2 illustrates the procedure as follows. The original string of random bits is coined *Codeword*. This is converted into voltages of type double coined *Encodedword* by converting zeros to -1.000000 and ones to 1.000000. Additive white

Gaussian *Noise* of type double is generated and added to the *Encodedword* to obtain the *Distorted* also of type double. Two types of data are obtained from *Distorted*, namely *AbsDistorted* and *Received*. The *AbsDistorted* of type double is simply the absolute values of the *Distorted* data while the *Received* is the result of the quantization of *Distorted*. Consequently, the two sources are correlated in that the second source is a noisy version of the first. In other words, *Codeword* and *Received* are correlated in that they differ in only a few bit positions. These are illustrated in Figure 4.2. It could be observed that at a particular instance, the first entry *codeword* was a '1', converting to voltage gives 1.000000, a random noise of 0.324900 was generated and added to get a *distorted* signal of 1.324900. The absolute *distorted* value in this case remains the same and this is quantised to give a binary digit of '1'.

Gaussian Noise sigma = 0.630957

FER = 1 BER = 0.638298 (Error Messages = 1, 30 / Messages Transmitted = 1, 47) Noise seed = 33

Codeword	Encodedword	Noise	Distorted	AbsDistorted	Received
1	1.000000	0.324900	1.324900	1.324900	1
1	1.000000	0.064607	1.064607	1.064607	1
1	1.000000	-1.035365	-0.035365	0.035365	0
1	1.000000	0.410807	1.410807	1.410807	1
0	-1.000000	-0.095226	-1.095226	1.095226	0
0	-1.000000	1.107614	0.107614	0.107614	1
1	1.000000	0.045918	1.045918	1.045918	1
1	1.000000	-0.232139	0.767861	0.767861	1
0	-1.000000	0.533173	-0.466827	0.466827	0
1	1.000000	0.248896	1.248896	1.248896	1
1	1.000000	0.856027	1.856027	1.856027	1
0	-1.000000	-0.413264	-1.413264	1.413264	0
0	-1.000000	0.655292	-0.344708	0.344708	0
0	-1.000000	-0.777424	-1.777424	1.777424	0
1	1.000000	1.232040	2.232040	2.232040	1
0	-1.000000	-0.249822	-1.249822	1.249822	0
1	1.000000	0.246318	1.246318	1.246318	1
1	1.000000	0.533263	1.533263	1.533263	1
1	1.000000	0.346709	1.346709	1.346709	1
1	1.000000	-0.508268	0.491732	0.491732	1
0	-1.000000	1.035975	0.035975	0.035975	1
0	-1.000000	0.269071	-0.730929	0.730929	0
0	-1.000000	0.660958	-0.339042	0.339042	0
1	1.000000	0.111621	1.111621	1.111621	1
1	1.000000	1.126280	2.126280	2.126280	1
1	1.000000	-0.575045	0.424955	0.424955	1
1	1.000000	0.613778	1.613778	1.613778	1
0	-1.000000	0.353347	-0.646653	0.646653	0
1	1.000000	-0.941100	0.058900	0.058900	1
1	1.000000	1.418794	2.418794	2.418794	1
1	1.000000	0.045871	1.045871	1.045871	1
1	1.000000	-0.555140	0.444860	0.444860	1
1	1.000000	-0.101795	0.898205	0.898205	1
0	-1.000000	-0.999964	-1.999964	1.999964	0
1	1.000000	1.468008	2.468008	2.468008	1
1	1.000000	0.521683	1.521683	1.521683	1
1	1.000000	0.063519	1.063519	1.063519	1
0	-1.000000	0.486723	-0.513277	0.513277	0
0	-1.000000	-0.222212	-1.222212	1.222212	0
1	1.000000	0.388890	1.388890	1.388890	1
0	-1.000000	0.024369	-0.975631	0.975631	0
0	-1.000000	-0.091177	-1.091177	1.091177	0
1	1.000000	-0.721273	0.278727	0.278727	1
1	1.000000	0.581443	1.581443	1.581443	1
1	1.000000	-0.050047	0.949953	0.949953	1
0	-1.000000	-0.377424	-1.377424	1.377424	0
1	1.000000	-1.155863	-0.155863	0.155863	0

Figure 4.2: Evaluation of the most reliable bits of the received word.

Basically, the *k* most reliable bits that are received from the encoder are initially taken as correct and the *(n – k)* least reliable bits are taken as erasures. The parity-check matrix is used to solve for the erased bits and a codeword is obtained which is either equal to the transmitted codeword or needs only small changes.

Depending on the code, the *(n – k)* least reliable bits usually cannot all be considered to be erasures. This is dependent on two key factors: the positions of the erased coordinates as well as the power of the code. As a matter of fact, maximum distance separable (MDS) codes are the only codes that can solve *(n – k)* erasures irrespective of positions of the erasures in the received codeword. Unfortunately, there are no binary MDS codes apart from trivial examples.

Notwithstanding, a set of *(n – k)* erasures can always be solved from the *(n – k + s)* least reliable bit positions, where *s* is usually a small integer depending on the code. In order to obtain the best performance, the very least reliable bit positions are solved first. For any received coordinate, the priori log-likelihood ratio of the bit being correct is proportional to the magnitude of the absolute value of the received bit |*r*|. The received vector with coordinates ranked in order of reliability is defined as follows:

$$\left(r_{\mu_0}, r_{\mu_1}, r_{\mu_2}, \ldots, r_{\mu_{n-1}}\right) \tag{4.1}$$

where

$$\left|r_{\mu_0}\right| > \left|r_{\mu_1}\right| > \left|r_{\mu_2}\right| > \ \ldots > \left|r_{\mu_{n-1}}\right| \tag{4.2}$$

Figure 4.2 shows the steps of the initial generation of random bits of length 47, denoted by column named '*Codeword*'. The second column shows the random bits converted to voltages before the addition of additive white Gaussian noise to form the '*Distorted*'

column. As indicated in equation (4.2), the most reliable bits are those that have the highest magnitude of the absolute value of the received word. A plot of the absolute values of the received word is showed in Figure 4.3. It can be observed that at E_b/N_o = 4 dB, amongst the 47 randomly generated bits, the most reliable bit had an absolute magnitude of 2.4 while the least reliable bit had an absolute magnitude of about 0.2. In any case, the 23 least reliable bits are considered as erased and then recalculated.

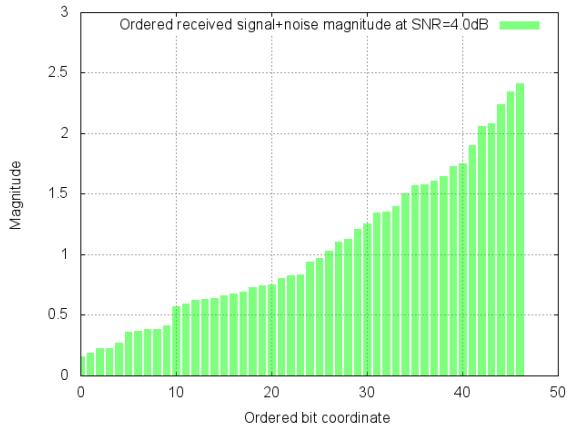


Figure 4.3: Illustration of received coordinate magnitude in their solved order for the (47, 24, 11) code.

This summarises the entire encoding operation. The noisy vector along with the associated syndromes are sent over a noiseless channel to the decoder.

4.5 Modification for Distributed Source Coding (DSC)

The conventional method of ensuring that any vector received at the decoder is indeed a codeword sent from an encoder over a transmission channel is by computing its syndrome. If an all-zero syndrome is obtained, it shows that that the received vector is at least, a codeword, but not necessarily the exact codeword sent. The minimum distance plays a vital role in telling how many errors a received vector may contain and still be decoded correctly.

The higher the minimum distance, the more errors it may contain and still be decoded correctly.

However, in contrast to convention, this scheme, does not deal with codewords in the technical context. Instead, we have strings of randomly generated bits. Thus, for all intent and purposes, the formed syndromes of the received vectors are a combination of zeros and ones. Consequently, these syndromes are used as the side information at the decoder along with the messages transmitted at any instant.

Again, for maximum distance separable (MDS) codes, this is quite straightforward as codewords are formed directly from the received vector, *r*.

$$\left(x_{\mu_0}, x_{\mu_1}, x_{\mu_2}, \dots, x_{\mu_{n-1}}\right) \tag{4.3}$$

The (n – k) coordinates are considered erased and derived from the most reliable coordinates using the parity-check matrix, **H**. A maximum distance separable code is classified as one of the pivotal class in error correcting codes that meet the singleton bound, ($d = n - k + 1$). But for non-MDS codes as in this scheme, (n – k) coordinates cannot be solved from the parity-check matrix, **H** because it is not a Cauchy or Van der-Monde matrix. An order that is slightly different from equation (4.3) is defined.

$$\left(x_{\eta_0}, x_{\eta_1}, x_{\eta_2}, \dots, x_{\eta_{n-1}}\right) \tag{4.4}$$

η_{n-1} is set equal to μ_{n-1} and $x_{\eta_{n-1}}$ is solved first by flagging the first parity-check equation that contains $x_{\eta_{n-1}}$ and then subtracting this equation from all other parity-check equations containing $x_{\eta_{n-1}}$. $x_{\eta_{n-1}}$ is then only contained in one equation, the first flagged equation.

The label of the next coordinate η_{n-2} is set equal to μ_{n-2} and an attempt is made to solve $x_{\eta_{n-2}}$ by finding an unflagged parity-check equation containing $x_{\eta_{n-2}}$. $x_{\eta_{n-2}}$ is set equal to μ_{n-3} if there is not an unflagged equation containing $x_{\eta_{n-2}}$. The procedure is repeated until an unflagged equation contains $x_{\eta_{n-2}}$. This is subtracted from all other parity-check equations containing $x_{\eta_{n-2}}$.

This procedure continues until all of the (n – k) codeword coordinates $(x_{\eta_{n-1}}, x_{\eta_{n-2}}, x_{\eta_{n-3}}, \dots, x_{\eta_k})$ have been solved and all (n – k) equations have been flagged. The remaining k ranked received coordinated are set equal to $(r_{\eta_0}, r_{\eta_1}, r_{\eta_2}, \dots, r_{\eta_{k-1}})$ in the most reliable order. $(x_{\eta_0}, x_{\eta_1}, x_{\eta_2}, \dots, x_{\eta_{k-1}})$ are then determined using the bit decision rule; $[x_{\eta_i} = 1 \ if \ r_{\eta_i} < 0, \ else \ x_{\eta_i} = 0]$.

Figure 4.4 highlights all 23 flagged equations of the [47, 24, 11] QR code parity-check matrix. It can be observed that every other entry aside the highlighted '1's (circled in red) in each column is a zero. These 23 columns which just a single '1' and all other entries '0' are used to recalculate the erased bits according to equation 4.8 to equation 4.11.

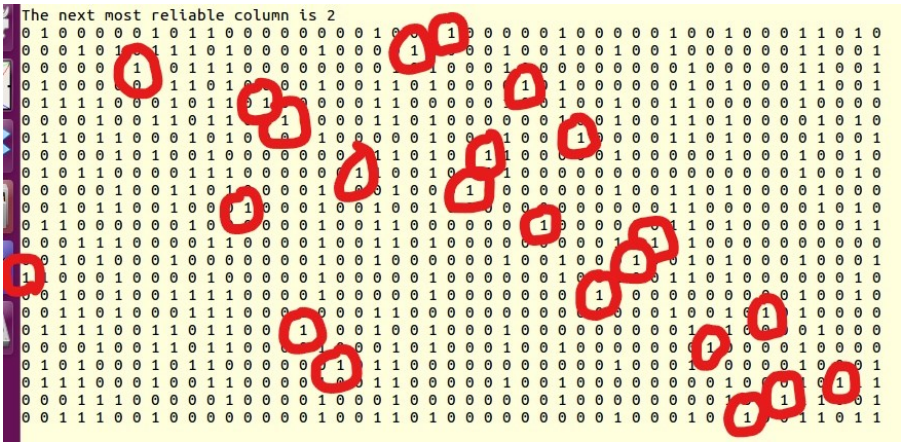


Figure 4.4: Illustration of the 23 Flagged Equations in the [47, 24, 11] Parity-Check Matrix

4.5.1 Evaluation of the Erased Bits

Going forward, once all the parity-check equations have been flagged, what follows is the evaluation of the erasures. Now the flagged parity-check equations are in upper triangular form and thus, would be solved in reverse order starting with the last flagged equation. The evaluating of the last flagged equation gives the solution to x_{η_k}. This is followed by evaluation $x_{\eta_{k+1}}$ and back substituting the solution and so on with coordinate $x_{\eta_{n-1}}$ solved at the end. We denote the set of e erasures as a list of erased bit positions defined as f_i where:

$$0 < i < e \quad f_i \in 0 \ldots n-1 \tag{4.5}$$

Again, in contrast to convention where an all zero syndrome suggests that the received vector is a codeword, in this case, a codeword $\boldsymbol{x} = x_0, x_1, \ldots, x_{n-1}$ satisfies the parity-check equations of the parity-check matrix **H:**

$$Hx^T = s \tag{4.6}$$

Furthermore, a codeword with e erasures is defined as

$$x = (x_{u_0}, x_{u_1}, \ldots, x_{u_{n-1-e}} | x_{f_0}, x_{f_1}, \ldots, x_{f_{e-1}}) \tag{4.7}$$

where x_{u_j} are the un-erased coordinates of the codeword, and the set of e erased coordinates is defined as f_e. There are a total of $(n - k)$ parity-check equations and the erased bits may be solved using a parity-check equation, provided that the erased bit positions correspond to independent columns of **H**.

Mathematically, unique parity-check equations were obtained for each of the erased bits. The parity-check equations may be used to solve each erasure as shown in equations (4.8) to (4.11):

$$x_{f_0} = h_{0,0}x_{u_0} + h_{0,1}x_{u_1} + \ h_{0,n-e-1}x_{u_{n-e-1}} + s_0 \tag{4.8}$$

$$x_{f_1} = h_{1,0}x_{u_0} + h_{1,1}x_{u_1} + \ h_{1,n-e-1}x_{u_{n-e-1}} + s_1 \tag{4.9}$$

$$x_{f_2} = h_{2,0}x_{u_0} + h_{2,1}x_{u_1} + \ h_{2,n-e-1}x_{u_{n-e-1}} + s_2 \tag{4.10}$$

... ...

$$x_{f_{e-1}} = h_{e-1,0}x_{u_0} + h_{e-1,1}x_{u_1} + h_{e-1,2}x_{u_2} + \ h_{e-1,n-e-1}x_{u_{n-e-1}} + s_{e-1} \tag{4.11}$$

where $h_{i,j}$ is the coefficient of row *i* and column *j* of **H** and s_i is the value of the syndrome at bit position *i*.

The decoded codeword is denoted as $\hat{x}$ and the mapped version is denoted as $\hat{c}$. The codeword most likely to be transmitted, $\check{x}$ has the smallest squared Euclidean distance, D ($\check{x}$) between the mapped codeword and the received vector. This is the case for soft-decision decoding. As for hard decision decoding, the Hamming distance is used in place of the Euclidean distance.

$$D(\check{x}) = \sum_{j=0}^{n-1}(r_j - \check{c}_j)^2 \tag{4.12}$$

$D(\check{x}) < D(x)$ for all other codewords, *x*.

4.5.2 Implementation of a Modified Dorsch Decoder for SW Channel.

An overview of the major steps taken in the modification of the Dorsch decoder using QR codes over the Slepian-Wolf channel is presented in Algorithm 4.1. QR codes of block lengths, $n = 47$ are used in this section. All algorithms are coded in C/C++ language.

Algorithm 4.1: Algorithm for QR Code Performance Simulations using BPSK Modulation Over the Slepian-Wolf Channel for a Modified Dorsch Decoder

Total-errors = 0

for *blocks transmitted = 1* **to** *maximum number of codewords transmitted in simulation* **do**

begin

- generate a length *n* random message using *ran2* to determine the values of its individual bits for source1.
- convert the binary message from source1 to signal voltages, that is, bit 0 = -1.0V and bit 1 = +1.0V, to represent BPSK modulated signal transmission $\boldsymbol{s_j}$ for $0 \leq j \leq n - 1$.
- generate Additive White Gaussian Noise signal of length *n* to affect converted signal in each bit position using; $\boldsymbol{N_j = \sigma_N \times gasdev}$, for $0 \leq j \leq n - 1$ and a new function call is made to *gasdev* for all *j*.
- add converted vector $\boldsymbol{s}$ to noise vector $\boldsymbol{N}$ to obtain the noisy version of the converted signal now, $\boldsymbol{X_j = s_j + N_j}$ for $0 \leq j \leq n - 1$.
- threshold the voltage signals of the noisy version by converting positive voltages to 1 and negative voltages to 0, we call these source2.
- compute the syndrome of the random bits of sources1 by multiplication by the parity check matrix, these become the side information at the decoder.
- obtain the absolute values of the noisy version and rearrange same in descending order to determine their reliability, vector named order [xj] for $0 \leq j \leq n - 1$ is created.
- each of the least reliable bit positions is used to flag a unique parity-check equation (or column) in the H-matrix by elementary row operations and then considered as erased.
- the most reliable bit positions are quantized and compared with the corresponding entries in the original random message.
-

for *iterations = 1* to *maximum number of iterations* **do**
begin
flip all the different bit patterns in the received vector until there are no errors in the most reliable bits.
end

- the most reliable bits are used to calculate each erased bit as per the flagged equations.

end

final error probability, P_e= *Total-errors* / *maximum number of codewords transmitted in simulation*

4.5.3 Simulations for performance evaluation

To ensure the reliability of results, a million blocks per noise seed across ten seeds of codewords was transmitted for a range of signal-to-noise ratios Eb/No. The algorithm developed was implemented in C language and all codewords were transmitted over the AWGN channel and binary phase shift keying BPSK modulation is applied.

Meanwhile, the theoretical bound of bits in error is first computed for each signal-to-noise ratio from a Q-function table. As an illustration, we know the error probability is obtained from equation (4.13)

$$P_e = Q\left(\sqrt{SNR}\right) \tag{4.13}$$

The signal-to-noise ratio (SNR) is obtained from equation (4.14)

$$SNR = 10^{0.1 \times SNR(dB)} \tag{4.14}$$

For 0.5dB,

$$SNR = 10^{0.05} = 1.122018454$$

Hence,

$$P^e = Q\left(\sqrt{1.22}\right) = 1.4686 \times 10^{-1}$$

This translates to approximately 15 errors in a hundred or 7 errors in 47. A plot of the theoretical upper bound of bits received in error is shown in Figure 4.5. A plot of the actual average number of errors obtained before the correction of any of the errors is also presented in Figure 4.6. It can be observed that at E_b/N_o = 5 dB, the theoretical bound foe average number of errors in the entire block (green bar) is about 1.75 in Figure 4.5 while the actual average number of errors before any error correction at the same E_b/N_o = 5 dB is about 0.1 in Figure 4.6. This is quite impressive considering the fact that error correction had not taken place.

On the other hand, at E_b/N_o = 0.5 dB, the theoretical bound for the number of bits in error from the most reliable bits is 6.9 while the actual average recorded for the most reliable bits before error correction is about 0.8.

The actual average number of bits in error is very relevant in estimating the number of times searches need to be made to correct an error (more on this in section 4.5.7).

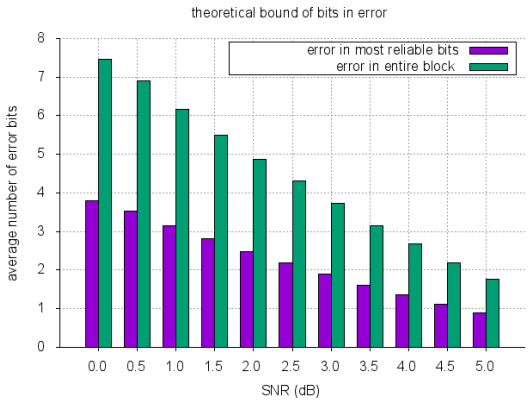


Figure 4.5: Theoretical upper bound of number of bits received in error for the (47, 24, 11) code.

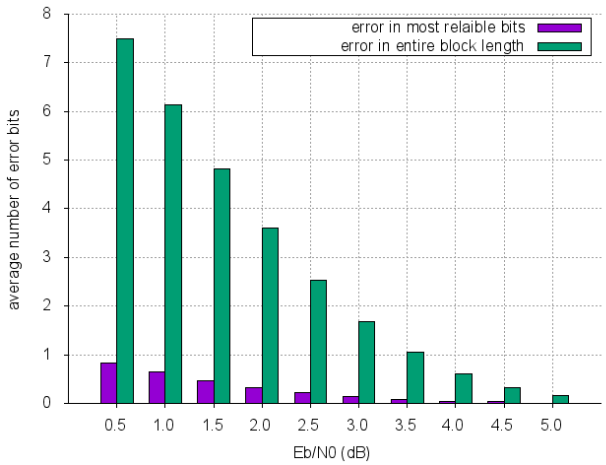


Figure 4.6: Actual number of bits received in error for the (47, 24, 11) code before error correction.

4.5.4 Comparison with state-of-the-art LDPC codes

It is understood from Shannon theorem that the error probability of a code tends to zero as the block length tends to infinity. For this reason, long length LDPC codes in particular, have good performance. However, this work proposes a model with short length codes and the highest known minimum distances for a given number of information symbols, a situation not suitable for iterative decoding. Simulation results are compared with those of LDPC codes of block lengths 1024 (Serener, Natarajan & Gruenbacher, 2008), 32 (Jin *et al.*, 2018) and 512 (Guo, 2018; Serener, Natarajan & Gruenbacher, 2008). Although this code is of length 47, it performs better than some of the longer LDPC codes as shown in Figure 4.7. The lower the curve, the better the performance. Take for instance, at E_b/N_o = 10 dB, (Serener, Natarajan & Gruenbacher, 2008) of block length, $n = 512$, recorded a bit in error after less than 100,000 bits were transmitted. Both (Serener, Natarajan & Gruenbacher, 2008) of block length, $n = 1024$ and the proposed model of block length, $n = 47$ recorded a bit in error after almost 1,000,000 bits were transmitted. The proposed

model of block length, $n = 24$ (the most reliable bits) recorded just one error bit after 1,000,000,000 bits were transmitted.

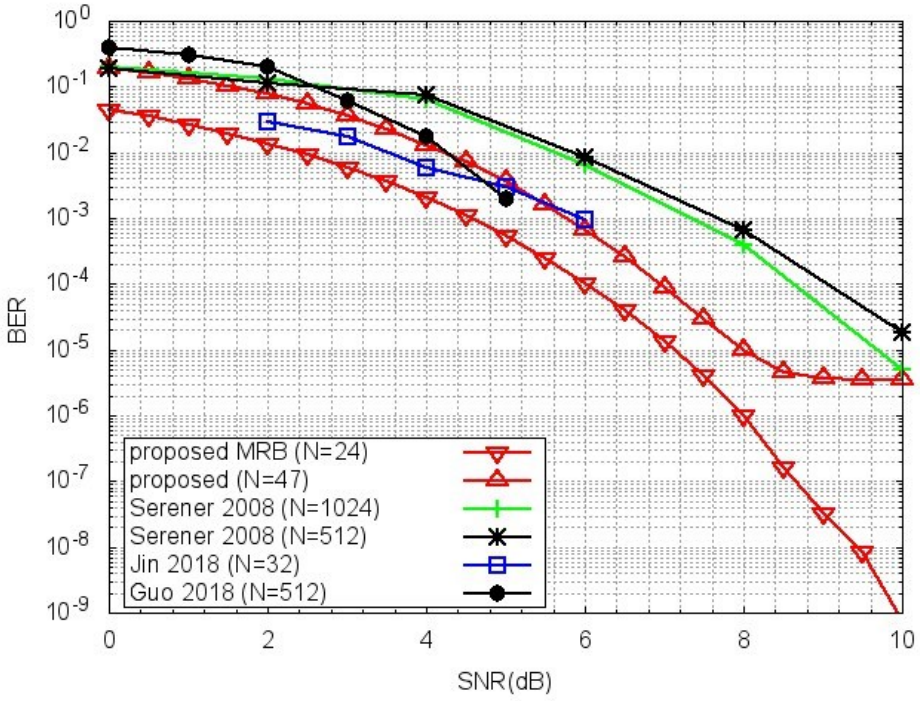
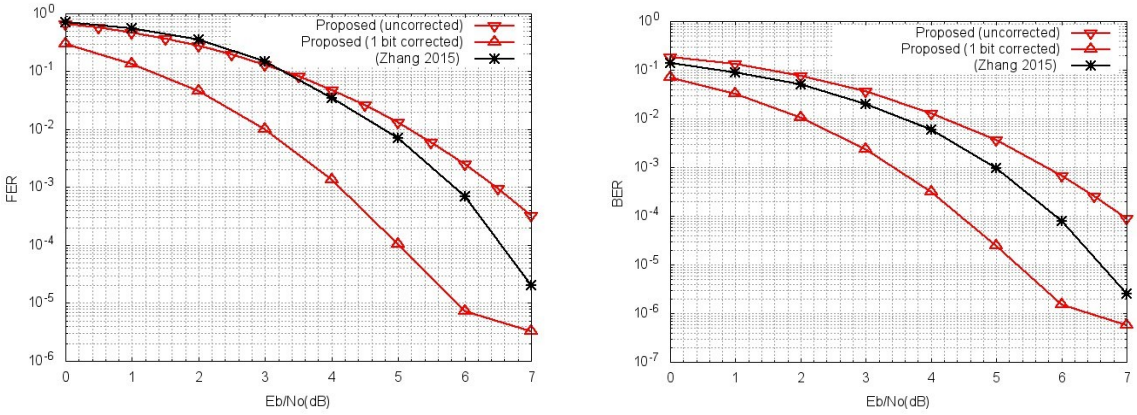


Figure 4.7: Performance comparison of the proposed model with other LDPC codes.

4.5.5 Correction of a Single Bit of Error of the [47, 24, 11] QR Code

Maximum likelihood decoding involves the evaluation of several candidate codewords and the codeword with the minimum decision metric (Euclidean distance for soft decision and Hamming distance for hard decision) is output from the decoder. Furthermore, maximum likelihood decoding is realised provided all codewords are evaluated up to a bounded information weight. As an initial step towards maximum likelihood decoding, a single bit of the [47, 24, 11] QR code was corrected. This is achieved by randomly flipping each bit of the most reliable bits, one at a time and then, the erasures are recalculated. The results

obtained before any error correction and after correcting a single bit of error are as presented in Figure 4.8 and Figure 4.9.



(a) (b)

Figure 4.8: Plots of (a) FER and (b) BER against E_b/N_o for the [47, 24, 11] code

Figure 4.8 is plotted alongside the performance from the same [47, 24, 11] code reported from (Zhang *et al.*, 2015) without determining the unknown syndromes. It was observed that the frame error rate is mostly the same for the reliable bits and the entire codelength whereas the is a significant difference in the bit error rate of the two. Furthermore, Figure 4.9 compares the performance with LDPC codes reported in (Jin *et al.*, 2018). It could be observed that, for the proposed (1 bit corrected), at E_b/N_o of 6dB, only 1 bit was received in error from 1,000,000 bits that was sent and received over the channel. This is quite impressive compared to other schemes as shown.

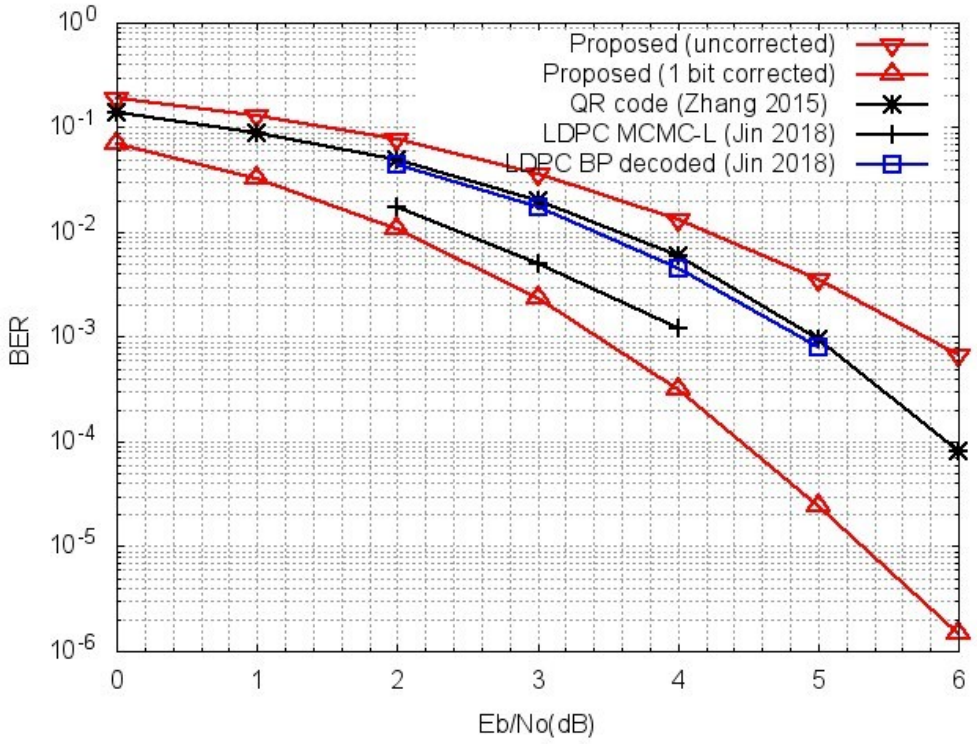


Figure 4.9: Plots of BER against Eb/No for different schemes shows the superiority of the proposed model.

4.5.6 Maximum Likelihood Decoding

Maximum likelihood decoding is realized provided all codewords are evaluated up to a bounded information weight. As far as this implementation is concerned, this would translate to correcting all possible error bits in the received vectors. To do this, the number of errors in each vector is first ascertained. Consequently, Figure 4.10 shows typical instances of the average number of bits received in error over different noise levels. The relevance of this plot is that it indicates the number of times a search is needed to correct any bits received in error. As an illustration, within the entire block-length (green plot),

there was an average occurrence of more than one bit and more than three bits in error at 1.0dB and 0.0dB respectively.

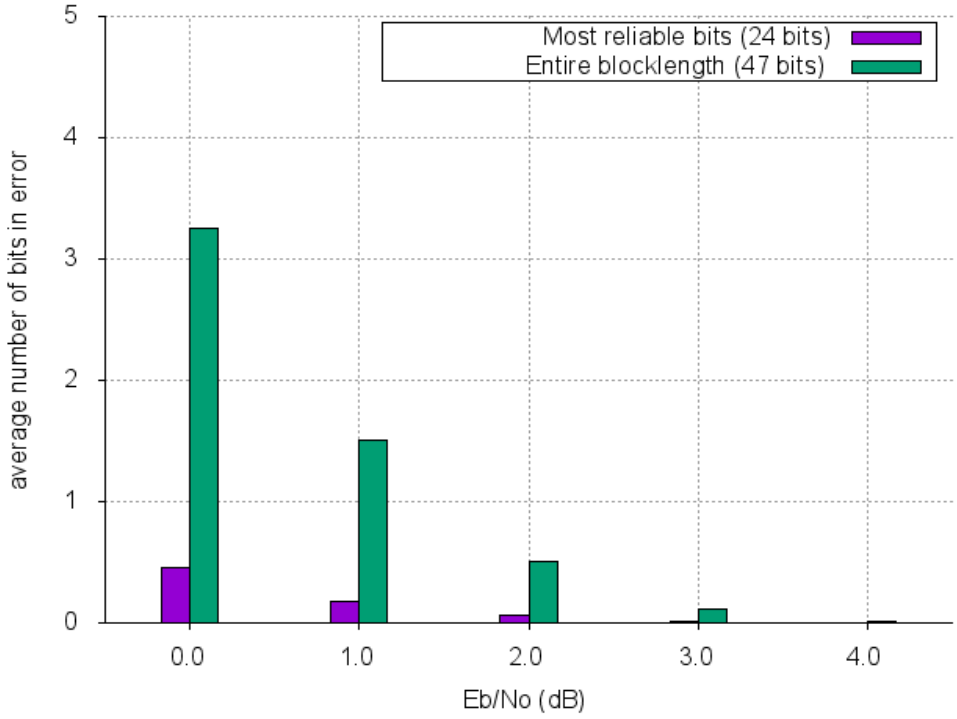
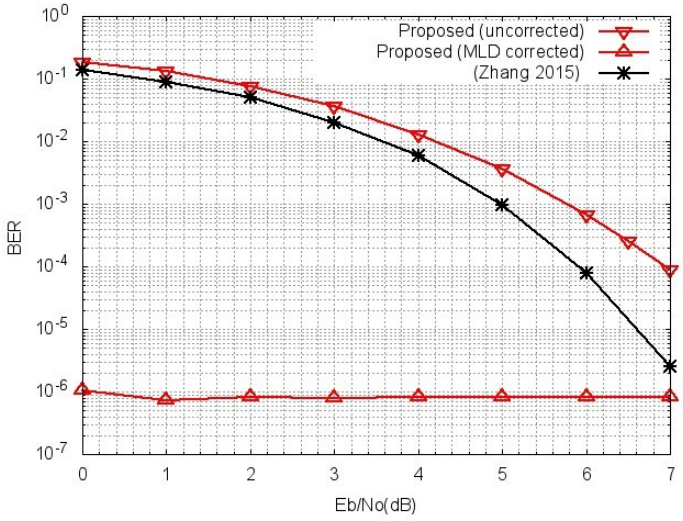


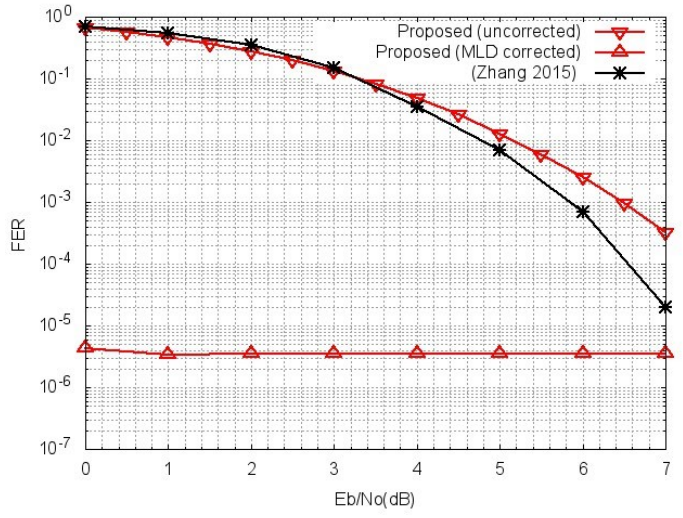
Figure 4.10: Average number of error bits after single error correction for the [47, 24, 11] QR code.

On the other hand, within the most reliable bits (purple plot), a maximum of one bit was received in error across all the Eb/No. This means the number of times that the decoder would search and attempt to correct errors in the received vector is proportional to the number of bits received in error. It is pertinent to note that within this decoder, error bits are searched for only within the most reliable bits (24 bits in this case). The erasures (the remaining 23 bits) are recalculated only after the errors in the most reliable bits have been corrected. Subsequently, an equivalence of maximum likelihood decoding was implemented by means of brute force to ensure there are no errors whatsoever within the

most reliable bits, and the erasures were recalculated. The result obtained is presented in Figure 4.11. It could be observed that, at Eb/No of 7dB, not a single error was detected after running 10 million blocks of code.



(a)



(b)

Figure 4.11: Plots of (a) BER and (b) FER against Eb/No for the (47, 24, 11) code

4.5.7 Complexity of Maximum Likelihood Decoding

The associated complexity of correcting all five bits of error in the [47, 24,11] QR code at Eb/No = 1dB is presented in Figure 4.12. To illustrate this, the number of searches that is necessary to correct a single bit of error would be the number of combinations of one bit that can be drawn from the twenty-four most reliable bits of a received vector.

That is,

$$\binom{24}{1} = 24 \tag{4.15}$$

The corresponding error probability is based on the actual results obtained from simulations. This implies that the complexity associated with extending this scheme to correct multiple error bits is also a function of the number of error bits to be intended to be corrected. Because the [47, 24, 11] QR code can correct up to five bits of error, the complexity associated with correcting all five bits of error is given by

$$\binom{24}{5} = 42504 \tag{4.16}$$

As stated earlier, maximum likelihood decoding is achieved using brute force to ensure there were no errors whatsoever within the twenty-four most reliable bits then, recalculating the erasures. Simulation results indicative of the corresponding bit error probabilities are also presented in Figure 4.12.

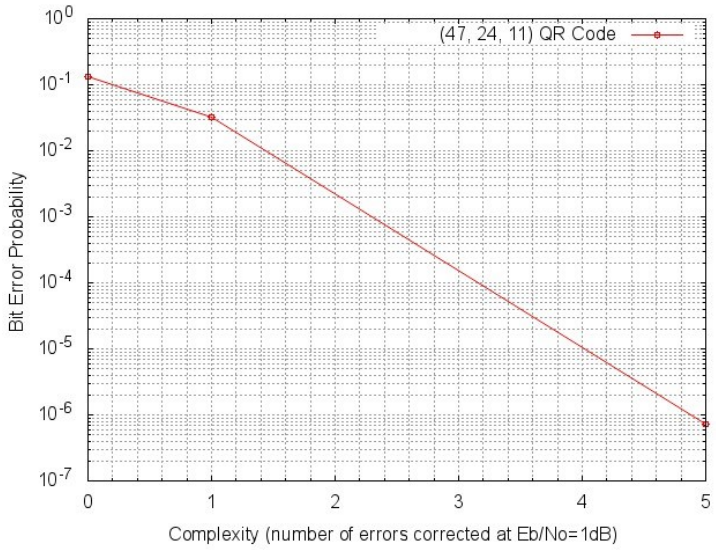


Figure 4.12: Error probability versus complexity associated with maximum likelihood decoding.

4.6 Summary

An improved distributed source coding scheme is presented to correct a single-bit error of the (47, 24, 11) QR code based on the modified Dorsch decoder without implementation of maximum-likelihood decoding. Although, this model corrects just one out of the five correctable error bits of the (47, 24, 11) QR code, simulation results show that it outperforms other schemes for decoding the same code.

Finally, a framework for attaining maximum likelihood decoding in addition to the complexity associated.

Chapter 5

Conclusion and Future Work

5.1 Contributions to Knowledge

- A simplified and detailed explanation of distributed source coding with emphasis on Slepian-Wolf coding of correlated information sources has been given. {Chapter 2}
- An efficient algorithm for decoding correlated information sources using very short, low delay [7, 4, 3] Hamming code that corrects up to a single bit of error. As a by-product, a speed-up to the algorithm is created for the less likely cases where more than one error occurs. This speed-up breaks out and ignore subsequent error(s) that might be present in any other bits of the 7 bits codelength once the first error is encountered. {Section 2.3.3}
- An extension of the Slepian-Wolf coding algorithm to correct two up to two bits of error of the [15, 7] BCH code by the introduction of look-up tables to store all possibilities of error vector patterns. {Section 2.4.2}
- A scheme for distributed source coding to correct multiple errors, using LDPC codes to compress close to the Slepian-Wolf limit for correlated binary sources. A conventional BP algorithm LDPC decoder which takes the syndrome information into account was developed. Subsequently, two novel equivalent channels were modelled; namely the additive white Gaussian noise equivalent channel and the BSC equivalent channel. The received probabilities in the conventional channel were replaced with the cross over probabilities. The performance achieved is seen to be better than previously published schemes for similar block length and code rate. {Chapter 3}

- It has been established that the proposed AWGN equivalent channel has approximately the same performance as that of the conventional AWGN channel for iterative decoding. Even though, the correlation between the sources at this stage is not ideal, a very good compression was achieved with no apparent loss in performance compared to the conventional decoder. This re-enforces the fact that quantisation error had not been introduced prior to the formation of the binary symmetric equivalent channel. {Section 3.5.2}
- A framework for the complexity associated with the proposed binary symmetric equivalent channel is established. It is shown that although the proposed scheme seems to have additional computational overhead due to the formation of syndromes each time random bit sequences are generated; this overhead is offset by the absence of encoding operation all together in the proposed scheme. The scheme does not entail the formation of codewords in the technical sense, only randomly generated bits are formed. {Section 3.5.4}
- Extension and presentation of error floor performance of the proposed binary symmetric equivalent channel as may be necessary for applications requiring such level of operation, for instance, in distributed video coding. {Section 3.6}
- An improved distributed source coding scheme is presented to correct a single-bit error of the (47, 24, 11) QR code based on the modified Dorsch decoder without implementation of maximum-likelihood decoding. This scheme is also best adapted for codes with the highest known minimum distances for given codelength and number of information symbols, and as such, are not suitable for iterative decoding. Although, this model corrects just one out of the five correctable error bits of the (47, 24, 11) QR code, simulation results show that it outperforms other schemes for decoding the same code. {Section 4.5.2}

- Establishment of a framework for attaining maximum likelihood decoding in addition to the complexity associated. {Section 4.5.6}

5.2 Conclusions and Recommendations for Future Work

This thesis focused mainly on developing efficient schemes for distributed source coding in general with special interest for Slepian-Wolf coding in particular. The challenges around schemes available in literature are centered around three major issues namely, the correlation channel model, the problem of systems with stringent delay restrictions and the complexity associated with the solutions offered.

The correction between sources has been modelled in various ways in literature. This thesis has proposed and implemented the AWGN equivalent channel which is not an ideal representation of the correlation between sources but, not withstanding, establishes the performance of the model in relation to the conventional channels. The proposed binary symmetric equivalent channel is an ideal correlation channel, and its efficiency has been established in this thesis.

It is established from Shannon's theory that the performance of codes generally improves as the code length approaches infinity. This explains why a whole lot of work has been carried out on iterative decoding for LDPC codes in literature. This research has implemented a scheme using medium-to-long length LDPC codes with iterative decoding. Furthermore, this scheme has also been implemented on very short, low delay codes like the [7, 4, 3] Hamming code, the [15, 7] BCH code and the [47, 24, 11] QR code.

The problem of striking a compromise between performance and complexity has been well research in literature. This thesis has also established the relationship between improving performance and the associated complexities.

Several new problems have emerged in the course of this research. This is in addition to the challenges that predates this work. A few suggestions to improve the performance and applications of the algorithms presented in this thesis include:

- It has been argued that a good channel error correction code constitutes a good source code. However, there is a need for thorough investigation on the extent of this claim as to establish if the reverse is also true. The question is, does a good source code automatically constitute a good channel code?
- All the schemes implemented in this work are based on the Slepian-Wolf coding, which is essentially a noiseless channel, although additive white Gaussian noises were introduced to establish a correlation between sources. It would be of interest to introduce an error correction element into the model.
- An interesting application of the DSC scheme would be in dirty-paper coding. This application is particularly interesting because, even with known interference (denoted as noise in this research) within the channel, digital data could still be efficiently transmitted and received. This could be achieved through source-splitting of the parity-check matrices, which are already pretty much, separable into the information part and the parity-check part.
- Although, this research has laid down the foundation for achieving maximum likelihood decoding using the Dorsch decoder by brute force, it would be interesting to push forward by means other than brute force, bearing in mind that, ideally information about the exact message transmitted is not available at the decoder.

References

Abdu-Aguye, U.-F., Ambroze, M. A. & Tomlinson, M. (2016a) 'Lowering the error floor of short-to-medium length LDPC codes using optimal low-correlated-edge density (OED) PEG Tanner graphs', *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, pp. 1-5.

Abdu-Aguye, U.-F., Ambroze, M. A. & Tomlinson, M. (2016b) 'Improved minimum weight, girth, and ACE distributions in ensembles of short block length irregular LDPC codes constructed using PEG and cyclic PEG (CPEG) algorithms', *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*. IEEE, pp. 186-190.

Akyol, E., Viswanatha, K. B., Rose, K. & Ramstad, T. A. (2014) 'On zero-delay source-channel coding'. *IEEE Transactions on Information Theory*, 60 (12), pp. 7473-7489.

Ancheta, T. (1976) 'Syndrome-source-coding and its universal generalization'. *IEEE Transactions on Information Theory*, 22 (4), pp. 432-436.

Bennatan, A. & Burshtein, D. (2006) 'Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels'. *Ieee Transactions on Information Theory*, 52 (2), pp. 549-583.

Bhattar, R. K., Ramakrishnan, K. R. & Dasgupta, K. S. (2010) 'Analysis of LDPC Codes for Compression of Nonuniform Sources with Side Information Using Density Evolution', *2010 Data Compression Conference*. 24-26 March 2010. pp. 522-522.

Bose, R. C. & Ray-Chaudhuri, D. K. (1960) 'On a class of error correcting binary group codes'. *Information and control*, 3 (1), pp. 68-79.

Chase, D. (1972) 'Class of algorithms for decoding block codes with channel measurement information'. *IEEE Transactions on Information theory*, 18 (1), pp. 170-182.

Chen, J., He, D. K. & Jagmohan, A. (2009a) 'On the Duality Between Slepian-Wolf Coding and Channel Coding Under Mismatched Decoding'. *Ieee Transactions on Information Theory*, 55 (9), pp. 4006-4018.

Chen, J., He, D. K. & Jagmohan, A. (2009b) 'The Equivalence Between Slepian-Wolf Coding and Channel Coding Under Density Evolution'. *Ieee Transactions on Communications*, 57 (9), pp. 2534-2540.

Chen, J. G., Fossorier, M. P. C. & Ieee (2002) 'Density evolution for BP-based decoding algorithms of LDPC codes and their quantized versions', *IEEE Global Telecommunications Conference (GLOBECOM 02)*. Taipei, Taiwan Nov 17-21. pp. 1378-1382.

Chen, X., Reed, I. S., Helleseth, T. & Truong, T.-K. (1994) 'Use of Grobner bases to decode binary cyclic codes up to the true minimum distance'. *IEEE Transactions on Information Theory*, 40 (5), pp. 1654-1661.

Chen, X., Reed, I. S. & Truong, T.-K. (1994) 'A performance comparison of the binary quadratic residue codes with the 1/2-rate convolutional codes'. *IEEE transactions on information theory*, 40 (1), pp. 126-136.

Chen, X. & Tuncel, E. (2011) 'Zero-delay joint source-channel coding for the Gaussian Wyner-Ziv problem', *2011 IEEE International Symposium on Information Theory Proceedings*. IEEE, pp. 2929-2933.

Chen, Y.-H. & Truong, T.-K. (2011) 'Fast algorithm for decoding of systematic quadratic residue codes'. *IET communications*, 5 (10), pp. 1361-1367.

Cheng, S., Wang, S. A., Cui, L. J. & Ieee (2009) 'Adaptive Slepian-Wolf Decoding using Particle Filtering based Belief Propagation'. *2009 47th Annual Allerton Conference on Communication, Control, and Computing, Vols 1 and 2*, pp. 607-612.

Chou, J., Pradhan, S. S. & Ramchandran, K. (2003) 'Turbo and trellis-based constructions for source coding with side information', *Data Compression Conference 2003 (DCC2003)*. Snowbird, Ut Mar 25-27. pp. 33-42.

Chung, S. Y., Richardson, T. J. & Urbanke, R. L. (2001) 'Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation'. *Ieee Transactions on Information Theory*, 47 (2), pp. 657-670.

Coleman, T. P., Lee, A. H., Medard, M. & Effros, M. (2004) 'On some new approaches to practical Slepian-Wolf compression inspired by channel coding', *Data Compression Conference (DCC 2004)*. Snowbird, UT Mar 23-25. pp. 282-291.

Coleman, T. P., Medard, M. & Effros, M. (2005) 'Towards practical minimum-entropy universal decoding', *IEEE Data Compression Conference*. Snowbird, UT Mar 29-31. pp. 33-42.

Cover, T. (1975) 'A proof of the data compression theorem of Slepian and Wolf for ergodic sources (Corresp.)'. *IEEE Transactions on Information Theory*, 21 (2), pp. 226-228.

Cover, T. M. (2006) *Elements of information theory.* ed. Thomas, J.A., 2nd ed. edn. Hoboken, N.J.: Hoboken, N.J. : Wiley-Interscience.

Cui, L. J., Wang, S., Cheng, S. & Yeary, M. (2011) 'Adaptive Binary Slepian-Wolf Decoding using Particle Based Belief Propagation'. *Ieee Transactions on Communications*, 59 (9), pp. 2337-2342.

Daneshgaran, F., Laddomada, M. & Mondin, M. (2009) 'LDPC-Based Iterative Algorithm for Compression of Correlated Sources at Rates Approaching the Slepian-Wolf Bound', *2009 First International Conference on Advances in Satellite and Space Communications*. 20-25 July 2009. pp. 74-79.

Davey, M. C. & MacKay, D. (1998) 'Low-Density Parity Check Codes over GF (q)'. *Ieee Communications Letters*, 2 (6), pp. 165-167.

Dorsch, B. (1974) 'A decoding algorithm for binary block codes and J-ary output channels (Corresp.)'. *IEEE Transactions on Information Theory*, 20 (3), pp. 391-394.

Dragotti, P. L. & Gastpar, M. (2009) *Distributed source coding: theory, algorithms and applications.* Academic Press.

Dubney, G., Reed, I. S., Truong, T.-K. & Yang, J. (2009) 'Decoding the (47, 24, 11) quadratic residue code using bit-error probability estimates'. *IEEE Transactions on Communications*, 57 (7),

Dupraz, E., Roumy, A. & Kieffer, M. (2013) 'Practical Coding Scheme for Universal Source Coding with Side Information at the Decoder', *Data Compression Conference (DCC)*. Ut Mar 20-22. pp. 401-410.

Dupraz, E., Savin, V. & Kieffer, M. (2015) 'Density Evolution for the Design of Non-Binary Low Density Parity Check Codes for Slepian-Wolf Coding'. *IEEE Transactions on Communications*, 63 (1), pp. 25-36.

El Gamal, A. & Kim, Y.-H. (2011) *Network information theory.* Cambridge university press.

Fang, Y. (2009) 'Crossover probability estimation using mean-intrinsic-LLR of LDPC syndrome'. *IEEE Communications Letters*, 13 (9), pp. 679-681.

Gallager, R. (1962) 'Low-density parity-check codes'. *IRE Transactions on Information Theory*, 8 (1), pp. 21-28.

Guo, D. (2018) 'LDPC Code Design via Masking Technology and Progressive Optimization', *2018 5th International Conference on Systems and Informatics (ICSAI)*. IEEE, pp. 769-773.

Hagenauer, J., Offer, E. & Papke, L. (1996) 'Iterative decoding of binary block and convolutional codes'. *IEEE Transactions on Information Theory*, 42 (2), pp. 429-445.

Hocquenghem, A. (1959) 'Codes correcteurs d'erreurs'. *Chiffres*, 2 (2), pp. 147-156.

Hu, X. Y., Eleftheriou, E. & Arnold, D. M. (2005) 'Regular and irregular progressive edge-growth tanner graphs'. *Ieee Transactions on Information Theory*, 51 (1), pp. 386-398.

Jin, J., Liang, X., Xu, Y., Zhang, Z., You, X. & Zhang, C. (2018) 'LDPC Decoder Based on Markov Chain Monte Carlo Method', *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. IEEE, pp. 219-222.

Lechner, G., Weidmann, C. & Ieee (2008) 'Optimization of Binary LDPC Codes for the q-ary Symmetric Channel with Moderate q'. *2008 5th International Symposium on Turbo Codes and Related Topics*, pp. 221-224.

Li, Y., Chen, G., Chang, H.-C., Chen, Q. & Truong, T.-K. (2015) 'An improved decoding algorithm of the (71, 36, 11) quadratic residue code without determining unknown syndromes'. *IEEE Transactions on Communications*, 63 (12), pp. 4607-4614.

Li, Y., Duan, Y., Chang, H.-C., Liu, H. & Truong, T.-K. (2018) 'Using the difference of syndromes to decode quadratic residue codes'. *IEEE Transactions on Information Theory*, 64 (7), pp. 5179-5190.

Lin, T.-C., Chang, H.-C., Li, Y., Chang, J. & Truong, T.-K. (2016) 'Algebraic decoding of the (71, 36, 11) quadratic residue code'. *IET Communications*, 10 (6), pp. 734-738.

Lin, T.-C., Su, W.-K., Shih, P.-Y. & Truong, T.-K. (2010) 'Fast algebraic decoding of the (89, 45, 17) quadratic residue code'. *IEEE Communications Letters*, 15 (2), pp. 226-228.

Liveris, A. D., Xiong, Z. X. & Georghiades, C. N. (2002) 'Compression of binary sources with side information at the decoder using LDPC codes'. *Ieee Communications Letters*, 6 (10), pp. 440-442.

MacWilliams, F. J. & Sloane, N. J. A. (1977) *The theory of error-correcting codes.* vol. 16. Elsevier.

Marshall, T. (1984) 'Coding of real-number sequences for error correction: A digital signal processing problem'. *IEEE Journal on Selected Areas in Communications*, 2 (2), pp. 381-392.

Matsuta, T., Uyematsu, T. & Matsumoto, R. (2010) 'Universal Slepian-Wolf Source Codes Using Low-Density Parity-Check Matrices'. *Ieice Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E93A (11), pp. 1878-1888.

Micallef, J. J., Farrugia, R. A. & Debono, C. J. (2011) 'LDPCA Code Construction for Slepian-Wolf Coding'. *IEEE Communications Letters*, 15 (10), pp. 1100-1103.

Pradhan, S. S., Chou, J. & Ramchandran, K. (2003) 'Duality between source coding and channel coding and its extension to the side information case'. *IEEE Transactions on Information Theory*, 49 (5), pp. 1181-1203.

Pradhan, S. S. & Ramchandran, K. (2000) 'Distributed source coding: symmetric rates and applications to sensor networks', *Proceedings DCC 2000. Data Compression Conference*. 2000. pp. 363-372.

Pradhan, S. S. & Ramchandran, K. (2003) 'Distributed source coding using syndromes (DISCUS): Design and construction'. *IEEE transactions on information theory*, 49 (3), pp. 626-643.

Pradhan, S. S. & Ramchandran, K. (2005) 'Generalized coset codes for distributed binning'. *Ieee Transactions on Information Theory*, 51 (10), pp. 3457-3474.

Prange, E. (1985) 'Some cyclic error-correcting codes with simple decoding algorithms'. *AFCRC-TN-58-156*,

Reed, I. S. (1959) *Statistical error control of a realizable binary symmetric channel.* Massachusetts Institute of Technology, Lincoln Laboratory.

Reed, I. S., Truong, T.-K., Chen, X. & Yin, X. (1992) 'The algebraic decoding of the (41, 21, 9) quadratic residue code'. *IEEE Transactions on Information Theory*, 38 (3), pp. 974-986.

Richardson, T. J., Shokrollahi, M. A. & Urbanke, R. L. (2001) 'Design of capacity-approaching irregular low-density parity-check codes'. *Ieee Transactions on Information Theory*, 47 (2), pp. 619-637.

Richardson, T. J. & Urbanke, R. L. (2001) 'The capacity of low-density parity-check codes under message-passing decoding'. *Ieee Transactions on Information Theory*, 47 (2), pp. 599-618.

Sartipi, M. & Fekri, F. (2004) 'Source and channel coding in wireless sensor networks using LDPC codes', *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.* 4-7 Oct. 2004. pp. 309-316.

Sartipi, M. & Fekri, F. (2005) 'Distributed source coding in wireless sensor networks using LDPC coding: the entire Slepian-Wolf rate region', *IEEE Wireless Communications and Networking Conference, 2005*. 13-17 March 2005. pp. 1939-1944 Vol. 1934.

Sartipi, M. & Fekri, F. (2008) 'Distributed source coding using short to moderate length rate-compatible LDPC codes: the entire Slepian-Wolf rate region'. *IEEE Transactions on Communications*, 56 (3), pp. 400-411.

Seiler, M. C. & Seiler, F. A. (1989) 'NUMERICAL RECIPES IN C - THE ART OF SCIENTIFIC COMPUTING'. *Risk Analysis*, 9 (3), pp. 415-416.

Serener, A., Natarajan, B. & Gruenbacher, D. M. (2008) 'Lowering the Error Floor of Optimized Short-Block-Length LDPC-Coded OFDM via Spreading'. *IEEE Trans. Vehicular Technology*, 57 (3), pp. 1646-1656.

Shannon, C. E. (1949) 'Communication theory of secrecy systems'. *Bell System Technical Journal, The*, 28 (4), pp. 656-715.

Slepian, D. & Wolf, J. (1973) 'Noiseless coding of correlated information sources'. *IEEE Transactions on Information Theory*, 19 (4), pp. 471-480.

Stankovic, V., Liveris, A. D., Xiong, Z. X. & Georghiades, C. N. (2006) 'On code design for the Slepian-Wolf problem and lossless multiterminal networks'. *Ieee Transactions on Information Theory*, 52 (4), pp. 1495-1507.

Stankovic, V., Liveris, A. D., Xiong, Z. X. & Georghiades, C. N. (2004) 'Design of Slepian-Wolf codes by channel code partitioning', *Data Compression Conference (DCC 2004)*. Snowbird, UT Mar 23-25. pp. 302-311.

Tomlinson, M. & Ambroze, M. (2010) 'Decoding of serial concatenated codes using erasure patterns'.[in Google Patents. (Accessed:Tomlinson, M. & Ambroze, M.

Tomlinson, M., Tjhai, C. & Ambroze, M. (2007) 'Extending the Dorsch decoder towards achieving maximum-likelihood decoding for linear codes'. *IET Communications*, 1 (3), pp. 479-488.

Tomlinson, M., Tjhai, C., Cai, J. & Ambroze, M. (2007) 'Analysis of the distribution of the number of erasures correctable by a binary linear code and the link to low-weight codewords'. *IET communications*, 1 (3), pp. 539-548.

Tomlinson, M., Tjhai, C. J., Ambroze, M. A., Ahmed, M. & Jibril, M. (2017) *Error-Correction Coding and Decoding.* Springer.

Toto-Zarasoa, V., Roumy, A. & Guillemot, C. (2010) 'Non-asymmetric Slepian-Wolf coding of non-uniform Bernoulli sources', *2010 6th International Symposium on Turbo Codes & Iterative Information Processing*. 6-10 Sept. 2010. pp. 304-308.

Vaezi, M. (2014) *Distributed Lossy Source Coding Using BCH-DFT Codes*. McGill University Libraries.

Vaezi, M., Combernoux, A. & Labeau, F. (2013) 'Low-delay joint source-channel coding with side information at the decoder', *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*. IEEE, pp. 228-232.

Vaezi, M. & Labeau, F. (2012a) 'Distributed lossy source coding using real-number codes', *2012 IEEE Vehicular Technology Conference (VTC Fall)*. IEEE, pp. 1-5.

Vaezi, M. & Labeau, F. (2012b) 'Improved modeling of the correlation between continuous-valued sources in LDPC-based DSC', *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, pp. 1659-1663.

Vaezi, M. & Labeau, F. (2013) 'Extended subspace error localization for rate-adaptive distributed source coding', *2013 IEEE International Symposium on Information Theory*. IEEE, pp. 2174-2178.

Vaezi, M. & Labeau, F. (2014) 'Distributed source-channel coding based on real-field BCH codes'. *IEEE Transactions on Signal Processing*, 62 (5), pp. 1171-1184.

Wainwright, M. J. & Martinian, E. (2009) 'Low-Density Graph Codes That Are Optimal for Binning and Coding With Side Information'. *IEEE Transactions on Information Theory*, 55 (3), pp. 1061-1079.

Wang, Z. l., Li, X. m. & Xu, Y. (2009) 'An Improved Decoding Algorithm for Distributed Video Coding', *2009 2nd International Congress on Image and Signal Processing*. 17-19 Oct. 2009. pp. 1-4.

Wolf, J. (1983) 'Redundancy, the discrete Fourier transform, and impulse noise cancellation'. *IEEE Transactions on Communications*, 31 (3), pp. 458-461.

Wyner, A. & Ziv, J. (1976) 'The rate-distortion function for source coding with side information at the decoder'. *IEEE Transactions on Information Theory*, 22 (1), pp. 1-10.

Zhang, K., Tomlinson, M., Ahmed, M. Z., Ambroze, M. & Rodrigues, M. R. (2014) 'Best binary equivocation code construction for syndrome coding'. *IET Communications*, 8 (10), pp. 1696-1704.

Zhang, P., Li, Y., Chang, H.-C., Liu, H. & Truong, T.-K. (2015) 'Fast decoding of the (47, 24, 11) quadratic residue code without determining the unknown syndromes'. *IEEE Communications Letters*, 19 (8), pp. 1279-1282.

Zixiang, X., Liveris, A. D. & Cheng, S. (2004) 'Distributed source coding for sensor networks'. *IEEE Signal Processing Magazine*, 21 (5), pp. 80-94.