JOINT SOURCE CHANNEL CODING USING SEQUENTIAL DECODING

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Bekir Ahmet Doğrusöz

August 1997

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Erdal Arıkan (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Melek D. Yücel

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. İrşadi Aksun

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

JOINT SOURCE CHANNEL CODING USING SEQUENTIAL DECODING

Bekir Ahmet Doğrusöz
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Erdal Arıkan
August 1997

In systems using conventional source encoding, source sequence is changed into a series of approximately independent equally likely binary digits. Performance of a code is bounded with the rate distortion function and improves as the redundancy of the encoder output is decreased. However decreasing the redundancy implies increasing the block length and hence the complexity.

For the systems requiring low complexity at transmitter, joint source channel (JSC) coding can be successfully used for direct encoding of source into the channel for lossless recovery. In such a system, without any distortion, compression depends on the redundancy of the source, and is bounded by the Rényi entropy of the source. In this thesis we analyze transmission of English text with a JSC coding system. Written English is a good example for sources with natural redundancy. Since we are unable to calculate the Rényi entropy of written English, we obtain estimates and compare with the experimental results.

We also work on an alternative source encoding method for accuracy-compression trade-off in joint source channel coding systems. The proposed stochastic distortion encoder (SDE) is capable of achieving accuracy-compression trade-off at any average distortion constraint with very low block lengths, and hence performs better than or as good as an equivalent rate distortion encoder. As block length approaches infinity the performance of stochastic distortion encoder approaches rate distortion function. Formulations for optimal SDE design and results for block lengths 1,2 and 3 are also given.

Keywords : Coding, Information Theory, Lossy Source Encoding, Joint Source-Channel Decoding, Sequential Decoding.

ÖZET

ARDIŞIK KODÇÖZÜMÜ KULLANARAK BİRLEŞİK KAYNAK-KANAL KODLAMA

Bekir Ahmet Doğrusöz
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Erdal Arıkan
Ağustos 1997

Geleneksel kaynak kodlama yöntemleri kullanan sistemlerde kaynak dizini yaklaşık olarak bağımsız ve eş olasılıklı iki basamaklılardan oluşan bir dizine dönüştürülür. Herhangi bir kodun başarı ölçümü, hız-bozulma fonksiyonu ile sınırlanmıştır ve kodlayıcı çıktısının artıklığı azaldıkça artacaktır. Ancak artıklığı azaltmak, kodlayıcının öbek uzunluğunu ve böylece de karmaşıklığını arttırmak anlamına gelir.

Göndericide düşük karmaşıklık gerektiren sistemlerde, birleşik kaynak kanal kodlama metodu başarıyla kullanılabilir. Böyle bir sistemde, hiç bozulma olmadığında, sıkıştırma kaynağın entropisi ile sınırlıdır. Bu tez için üzerinde çalışılan birinci konu, İngilizce metinlerin birleşik kaynak-kanal kodlamalı sistemlerde iletiminin analizidir. İngilizce metinler doğal artıklığa sahip kaynaklara iyi bir örnektir. İngilizcenin Renyi entropisi doğrudan doğruya hesaplanamasa da kestirimler elde edilmiş ve deneysel sonuçlarla karşılaştırılmıştır.

Üzerinde çalışılan ikinci bir konu da, birleşik kaynak-kanal kodlamalı sistemlerde doğruluk-sıkıştırma ödünleşimi için alternatif bir kaynak kodlama metodudur. İleri sürülen olasılıklı bozulum kodlayıcısı, çok düşük öbek uzunluklarıyla bile, istenilen herhangi bir ortalama bozulum sınırlamasında doğruluk-sıkıştırma ödünleşimi gerçekleştirebilir; böylece de eşdeğer bir hız-bozulum kodlayıcısına eşit ya da ondan daha iyi iş verimliliği gösterir. Öbek uzunluğu sonsuza yaklaşırken, olasılıklı bozulum kodlayıcısının iş verimliliği hız-bozulma fonksiyonu ile gösterilen kuramsal üst limite yaklaşmaktadır. En iyi olasılıklı bozulum kodlayıcısı tasarımı için kullanılacak denklemler, ve öbek uzunlukları 1,2 ve 3 için tasarımlar sonuçlar arasında verilmiştir.

Anahtar Kelimeler : Kodlama, Bilişim Kuramı, Yitimli Kaynak Kodlama, Birleşik Kaynak-Kanal Kodlama, Ardışık Kodçözümü.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Erdal Arıkan for his supervision, guidance, suggestions, instruction and encouragement through the development of this thesis.

I would like to thank to Dr. Melek D. Yücel and Dr. İrşadi Aksun for reading the manuscript and commenting on the thesis.

I would like to express my appreciation to all of my friends, for their continuous support through the development of this thesis. Finally, I would like to thank to my parents, sister and my grandmother, whose constant enthusiasm fed my motivation.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

D_{avg} : Average distortion.

D_C : Average distortion constraint.

$Dist(C_i, C_j)$: Hamming distance between two codewords, C_i and C_j of a codebook.

F_J : J-Gram Entropy estimate of English.

$G_n(X|Y)$: Number of computations to decode first n symbols of transmitted sequence $\mathbf{X}$.

$G_n(U|Y)$: Number of computations to decode first n bits of binary source sequence $\mathbf{U}$.

$H_{1/2}$: Rényi entropy of order 1/2.

H_α : Rényi entropy of order α.

$H_{\frac{1}{1+\rho}}$: Rényi entropy of order $\frac{1}{1+\rho}$.

$H^E_{1/2}$: Rényi entropy (of order 1/2) of English.

H^E : Entropy of English.

M_n : Metric for a node at level n of the code tree.

P : Probability mass function of source sequence.

P_N : Joint probability of source vectors of length N.

$\hat{P}$: Probability mass function of distorted source sequence.

$\hat{P}_N$: Joint probability of distorted source vectors of length N.

P_e : Bit error probability of binary symmetric channel.

Q : Probability mass function of channel input sequence X.

q_i^J : J-Gram letter frequencies for the letter i of a guess sequence.

$\mathcal{R}$: Region representing set $\mathcal{W}_N$ in 2^{2^N} dimensional space.

R_B : Bias term of the metric, modified for dictionary aided decoding.

R_c : Rate of the convolutional encoder.

$R_{c_{max}}$: Maximum rate of the convolutional encoder.

$R^1_{c_{max}}$: Maximum rate of the convolutional encoder for unigram JSC coding system.

$R^2_{c_{max}}$: Maximum rate of the convolutional encoder for digram JSC coding system.

$R(D, Q)$: Rate distortion function of a source with distribution Q.

$R(D)$: Rate distortion function.

R_f : Cut-off rate.

R_{SDE} : Rate of the stochastic distortion encoder.

T_J : J-Gram Rényi Entropy (of order 1/2) estimate of English.

T_D: Estimate of Rényi entropy (of order 1/2) for dictionary aided decoding.

$\mathbf{U}$: Binary source sequence.

$\mathbf{U}_N$: Binary source sequence vectors of length N.

$\hat{\mathbf{U}}$: Distorted binary source sequence.

$\hat{\mathbf{U}}_N$: Distorted binary source sequence vectors of length N.

$\mathbf{u}_i$: Source vectors of finite length that are input to a convolutional encoder at any given time i. Also denotes 5-bit vector representations of letters for JSC coding of English text.

W_N : Forward test channel transition matrix for a stochastic distortion encoder with memory length N.

$\mathcal{W}_N$: Set of all possible W_N for a given memory length N.

$\mathbf{X}$: Channel input sequence.

$\mathbf{X}_N$: Channel input vectors corresponding to source sequence vectors $\mathbf{U}_N$ or distorted sequence vectors $\hat{\mathbf{U}}_N$ for lossless and lossy JSC coding respectively.

$\mathbf{x}_i$: Channel input vectors corresponding to $\mathbf{u}_i$.

$\mathbf{Y}$: Channel output sequence.

$\mathbf{Y}_N$: Channel output vectors corresponding to source sequence vectors $\mathbf{U}_N$ or distorted sequence vectors $\hat{\mathbf{U}}_N$ for lossless and lossy JSC coding respectively.

$\mathbf{y}_i$: Channel output vectors corresponding to $\mathbf{u}_i$.

λ_{SDE} : Compression ratio for stochastic distortion encoder.

λ_{RDE} : Compression ratio for conventional rate distortion encoder.

Chapter 1

INTRODUCTION

Shannon entropy of a source is the minimum average number of bits (or nats) required to represent the source symbol. It provides a fundamental lower bound to the expected codeword length of any lossless code. The basic design procedure implied by Shannon's theorems consists of designing a source encoder which changes the source sequence into a series of independent equally likely binary digits, followed by a channel encoder which accepts the binary digits and puts them into a form suitable for reliable transmission. One aspect of this overall system is the increase in system complexity that results from this separation.

In certain communication systems, major concern is the simplicity of the operations performed at the transmitter, rather than the optimization of overall system performance. Conventional source coding is not well suited for such systems due to its complexity. To overcome the difficulty of using high complexity source encoders Hellman [1] and Koshelev [2] propose using a convolutional encoder for encoding the source output directly into the channel and decoding the output of the channel directly into source symbols using the natural redundancy of the source - a method which is known as Joint Source-Channel (JSC) Coding. They show that convolutional codes can be used at rates greater than one to achieve optimal lossless source coding.

As an example consider the English text: 'HE STOOD THERE FOR A MOMENT THEN WITH AN ...'. Ignoring the punctuation, paragraphs, etc. we can send any such text with an average of minimum $\log_2 27$ bits (26 letters and a space, 27 characters total). For that purpose we will have to apply a

source coding algorithm, Huffman coding for example, and obtain a bit sequence. If the channel is noisy we will have to further encode it for adding redundancy for reliable transmission and error free reconstruction of the text at the receiver. This separated encoding scheme might enable us to send text at rates close to the entropy of written English, however has vast complexity.

Consider the case instead, in which each letter is simply represented by 5 bits, and the resulting bit sequence is directly encoded with a convolutional encoder of a rate higher than one, into the channel. With or without errors in the received sequence, a sequential encoder will have to guess which path does the received sequence indicates. Even without errors, there will be more than one possible path that the received sequence indicates since the rate of the convolutional encoder was higher than one. Assume that the decoder made a wrong guess so that 'O' of the 'STOOD' changed to a 'P'. Then by the error propagating nature of the tree codes the following text will appear garbled: 'HE STOPRJKLMNTRE...'. The propagating error will eventually be detected and the sequential decoder will go on guessing until a correct guess is made.

However if the number of possible out-branching paths at any level is too many due to high noise or low transmission rate, the decoder will get stuck in another guessing problem before the first one is solved and the accumulation of these will divert the decoder away from the correct path in the code tree, and cause a tremendous increase in the computational effort. Therefore moments of computations with respect to channel statistics and transmission rate should be bounded for the correct operation of the sequential decoder. Arıkan and Merhav [3] give lower bounds on the moments of computation in terms of transmission rate, channel and source statistics, and specify the limits of performance for any JSC coding system with a sequential decoder based on the requirement that the first moment of computations shall be finite.

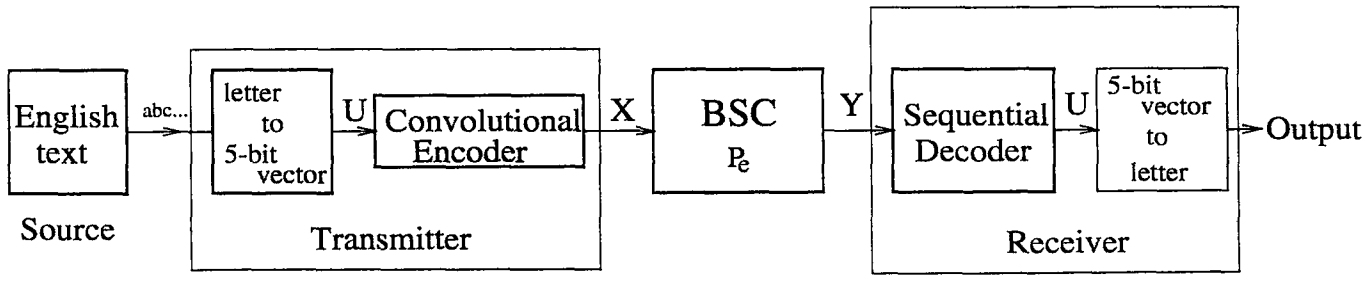

Figure 1.1: Block diagram of the lossless joint source channel coding system for transmission of text.

We use English text as an example for a source with natural redundancy and test the limits of JSC systems that are given in [3]. Consider the system

in Fig. 1.1. We use a text in English as input. Letters are transformed to 5-bit blocks $\mathbf{u}_i = (u_{i1}, \ldots, u_{i5})$ and a binary source sequence $\mathbf{U}$ is obtained. $n/5$ source vectors (where n is an integer multiple of 5) $\mathbf{u}_i$ are encoded into $n/5$ channel input vectors $\mathbf{x}_i$ which have a total length k with a rate $R_c = n/k$, by a convolutional encoder. A sequential decoder observes the output of the binary symmetric channel(BSC) and reconstructs $\mathbf{u}_i$ free of error, using the first and second order statistics of letters in an English text extracted from a training text of more than 26000 characters. We also employ a word-checking sequential decoding algorithm which makes use of a dictionary as a knowledge of higher statistics of English.

In lossy source coding, the role of Shannon entropy in lossless coding is replaced by the rate-distortion function, which gives the minimum number of bits (or nats) per symbol needed to convey in order to reproduce source with an average distortion that does not exceed a certain limit. Conventional source encoders, which are used to achieve accuracy-compression trade-off within the limits of rate-distortion function, require high complexity and are not well suited for simple transmitter systems.

In this thesis, we also propose an alternative low complexity quantizer that will allow accuracy-compression trade-off for JSC coding systems. We model the quantizer as a stochastic distortion encoder (SDE), which would alter the source statistics so as to increase the redundancy without changing the source sequence length. Although no compression is done at the source encoder, a cascaded convolutional encoder which is used for direct coding of source letters into the channel as offered in [1] and [2] would be able to achieve higher rates due to increased redundancy. The stochastic distortion encoder, even with very low complexity, is capable of providing accuracy-compression trade-off at any desired average distortion level unlike a conventional rate distortion encoder.

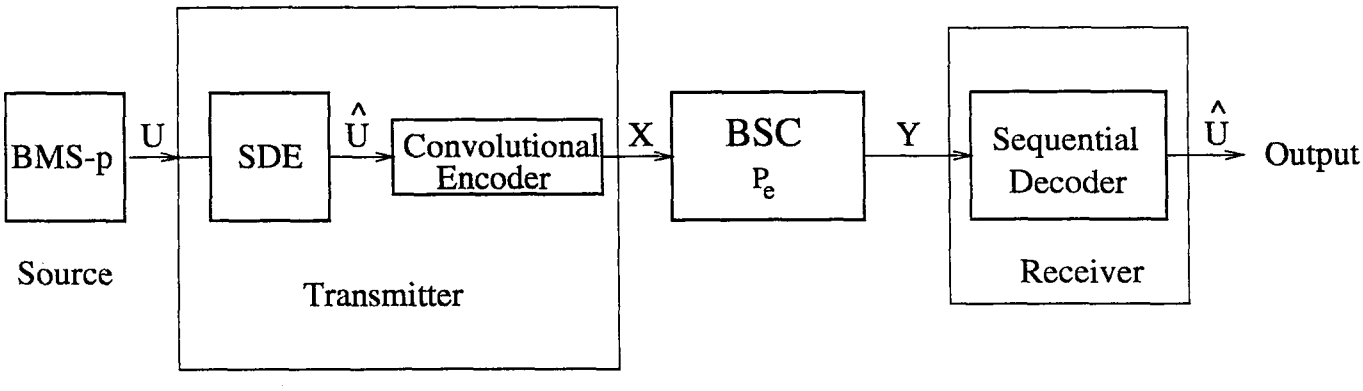

Figure 1.2: Block diagram of the lossy joint source channel coding system.

We consider the joint source-channel coding system in Fig. 1.2. For the sake of simplicity we switch to a binary memoryless source (BMS-p). Data from the binary memoryless source (BMS-p) is to be sent via a binary memoryless channel (BSC) with bit error probability P_e. The source generates random vectors $\mathbf{U}_N = (U_1, \ldots, U_N)$ with joint probabilities P_N, which are encoded into other random vectors $\hat{\mathbf{U}}_N = (\hat{U}_1, \ldots, \hat{U}_N)$ of same length but altered joint statistics $\hat{P}_N$ by a SDE. Vectors $\hat{\mathbf{U}}_N$ are encoded into the channel input vectors $\mathbf{X}_K = (X_1, \ldots, X_K)$ by a convolutional encoder and sent over the binary memoryless channel. The rate of the convolutional encoder is $R_c = N/K$ source bits/channel bits. The sequential decoder at the receiver observes the channel output $\mathbf{Y}_K = (Y_1, \ldots, Y_K)$ and aims to recover the vectors $\hat{\mathbf{U}}_N$.

We use a sequential decoder at the receiver of the JSC coding system. Koshelev [2] proves that, the tree structure of convolutional codes permits the use of a modified sequential decoding algorithm for joint source -channel decoding. For the sequential decoder, we use the simple stack algorithm with an infinite stack [4], and a modification of the Fano metric [5]. In the Fano metric, a constant bias term R (R being the rate of the convolutional encoder at the transmitter) is used assuming that the source sequence bits are independent and equally likely. For the JSC decoding, knowing the non-uniform statistics of the source, we replace the constant bias term with the a-priori probabilities of the source vectors for which the metric is calculated.

Sequential decoding allows us to use convolutional encoders with much higher constraint lengths for greater error correction capability in case of a noisy channel. However, expected number of computations for correct decoding increases exponentially for rates exceeding the computational cut-off rate of sequential decoder and presents an upper bound for rate, below the capacity of the channel. For an ordinary convolutional encoder-sequential decoder system the rate of the convolutional encoder is limited by the cut-off rate, R_f [6] :

$$R_f = max_Q \left(-\log \sum_y \left[\sum_x Q(x) P(y|x)^{\frac{1}{2}} \right]^2 \right) \tag{1.1}$$

where x and y denote the channel input and output respectively.

For the JSC coding system in Fig. 1.2 the rate of convolutional encoder is limited by cut-off rate/Rényi entropy of distorted sequence [3],

$$R_c \leq \frac{R_f}{H_{1/2}(\hat{\mathbf{U}}_N)/N} \tag{1.2}$$

and for the transmission of text (Fig. 1.1), the rate of convolutional encoder is limited by cut-off rate/Rényi entropy of English language. Rényi entropy of

order α for a binary source $\mathbf{U}$ is defined as:

$$H_\alpha(\mathbf{U}) = \frac{\alpha}{1-\alpha} \log \left[\sum_u P(u)^\alpha \right]^{1/\alpha} \tag{1.3}$$

where u denotes output bits. Rényi entropy will be explained in Chapter 2.

Stochastic distortion encoder maps blocks of N bits to some other blocks of N bits, and hence does not alter the length of the source sequence. In other words, the output of the SDE (the distorted sequence) has the same length as the input. Although the source sequence is not compressed in a phsical sense, the entropy (Rényi entropy) of the distorted sequence will be lower than the entropy (Rényi entropy) of the original sequence. As the entropy of the sequence to be transmitted decreases, a convolutional encoder acting as the source-channel encoder of a JSC coding system will be able to transmit the data with a higher rate (Eqn. 1.2). Then the compression ratio of the SDE, λ_{SDE}, which is given in input source letters/output source letters, is in fact a virtual concept, and is defined as the *maximum allowable rate for the convolutional encoder $R_{c_{max}}$, in a JSC coding system with convolutional encoder-sequential decoder pair and a noiseless channel.* In other words, when the channel is noiseless, the maximum rate for convolutional encoder is equal to the compression ratio of SDE, $R_{c_{max}} = \lambda_{SDE}$. The rate of the SDE, R_{SDE}, dual to the rate of a conventional rate distortion encoder (RDE), R_{RDE} is defined as $R_{SDE} = \frac{1}{\lambda_{SDE}}$. The rate of SDE for system in Fig. 1.2 is then, $R_{SDE} = \frac{K}{N}\Big|_{P_e=0}$.

We attempt to optimize the stochastic encoder in the sense that, maximum compression ratio λ_{SDE} (which requires minimum entropy and maximum redundancy for the distorted sequence) is reached with an average distortion that does not exceed a certain limit. Based on the previous studies about JSC coding [3], [2], [1], we attempt to minimize the Rényi entropy of stochastic distortion encoder's output subject to an average distortion constraint. SDE is modeled with a forward test channel transition matrix which has a channel alphabet size of 2^N, where N is defined as the memory length (dual of block length for a conventional rate distortion encoder) of the SDE. Memory length of a SDE is the length N of the blocks that are mapped to other blocks of length N by that SDE. Due to the concavity of Rényi entropy over the set of possible test channel transition matrices, the optimal solutions are searched at the boundaries. The design criterion is given by Eqn. 1.4.

$$\min_{\hat{\mathbf{U}}_N : E[d(\hat{\mathbf{U}}_N, \mathbf{U}_N)] < D} H_{1/2}(\hat{\mathbf{U}}_N)/N. \tag{1.4}$$

Main results obtained can be listed as:

- Simulation results for coding of English text are obtained, and compared with the estimated bounds of compression of written English via JSC coding.
- A stochastic distortion encoder operating at a very low complexity performs better than (or as good as) an equivalent rate distortion encoder, leaving no unused residual redundancy. The advantage is not better rate for the same average distortion but achievability of any average distortion even with a very low complexity.
- As memory length (dual for block length) of a SDE approaches infinity, the rate of SDE approaches to rate distortion function $R(D)$.

$$R_{SDE}|_{N\to\infty} = R(D)$$

- Number of computations necessary to find an optimum SDE design increases hyper-exponentially with increasing memory length.
- Simulation results agree with the achievability of the bounds given for both the SDE and the sequential decoder for a lossy JSC coding system.

This thesis report is organized as follows. In chapter 2 the related works of Koshelev [2], Hellman [1], Arikan and Merhav [3] and Shannon [7] will be summarized. In chapter 3, methods and simulations for transmission of English text with a JSC coding system will be given. In chapter 4, stochastic distortion encoder will be introduced and formulations for optimal SDE design will be given. Chapter 5 consists of the results of the computations for optimal SDE designs with memory lengths 1,2 and 3, and the simulation results.

Chapter 2

BACKGROUND

2.1 Sequential Decoding

We use a sequential decoder at the receiver of the JSC coding system. Sequential decoding is a search algorithm invented by Wozencraft, for finding the transmitted path through a tree code.

The operation of a sequential decoder is explained in [6] as follows: Consider an arbitrary tree code formed by a convolutional encoder from an input sequence $\mathbf{U}$ and let $\mathcal{X}$ denote the set of all nodes at some fixed but arbitrary level, n channel symbols into the tree from the tree origin. Let X be a random variable distributed uniformly on $\mathcal{X}$ and denote the node which lies on the correct path for the transmitted sequence. X, in that case, also denotes the correct channel input sequence of length n. Let Y denote the received channel output sequence when X is sent.

A sequential decoder applied to this code begins its search at the origin and proceeds over the code tree to examine eventually a node $x' \in \mathcal{X}$, and goes on to explore nodes stemming from x'. We assume that if $X \neq x'$, which means that node x' does not lie on the correct path, the decoder , with the guidance of its metric, will trace its way back to nodes below level n and proceed to examine another node $x'' \in \mathcal{X}$. Assuming that the probability of decoding error is zero, the decoder will eventually examine the correct node X. Even though decoding error is never zero in practice, it can be made arbitrarily small by increasing the constraint length of the tree code.

2.1.1 Metric for the Sequential Decoder

The metric we use for the sequential decoder is the Fano metric, due Fano [5] and is given by the equation,

$$M_n = \sum_{i=1}^{n} \log \left(\frac{P(\mathbf{y}_i|\mathbf{x}_i)}{P(\mathbf{y}_i)} - R_B \right) \tag{2.1}$$

for a node at level n. Vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ correspond to estimated channel input and received channel output blocks for the i^{th} tree branch on the explored path respectively. $P(\mathbf{y}_i|\mathbf{x}_i)$ is the channel transition probability and R_B is a bias term equal to the rate of the convolutional encoder R_c. $P(\mathbf{y}_i)$ is given by the equation

$$P(\mathbf{y}_i) = \sum_{\mathbf{u}_i \in \mathcal{U}} P(\mathbf{u}_i) P(\mathbf{y}_i|\mathbf{u}_i) \tag{2.2}$$

where $P(\mathbf{u}_i)$ is the probability of a possible conv. encoder input $\mathbf{u}_i$ that corresponds to the i^{th} level of the code tree.

Massey [8] showed that the Fano metric is the required statistic for minimum error-probability decoding of variable length codes (including convolutional codes) and the natural choice of the bias R_B in the metric is the code rate when the convolutional encoder input is assumed to be memoryless and uniformly distributed.

Koshelev [2] and Hellman [1] suggest a modification to the metric due to source statistics. Since the assumption that the source sequence is memoryless and uniformly distributed is not applicable for the JSC coding system, the bias term R_B is replaced by the a-priori probability of the source vectors $\mathbf{u}_i$. The modified Fano metric for a node at level n can be written as

$$M_n = \sum_{i=1}^{n} \left(\log \frac{P(\mathbf{y}_i|\mathbf{x}_i)}{P(\mathbf{y}_i)} + \log P(\mathbf{u}_i) \right) \tag{2.3}$$

or

$$M_n = \sum_{i=1}^{n} \left(\log \frac{P(\mathbf{y}_i|\mathbf{x}_i)}{P(\mathbf{y}_i)} + \log P(\mathbf{u}_i|\mathbf{u}_{i-1}, \ldots, \mathbf{u}_{i-k}) \right) \tag{2.4}$$

for sources with k^{th} order memory.

2.1.2 Bounds on Sequential Decoding

The computational effort of a sequential decoder is a random variable that depends on the transmitted sequence, received sequence and the search algorithm. The moments of computations are lower-bounded by Arıkan [6].

Let $G_n(X|Y)$ be the number of nodes in $\mathcal{X}$ examined before and including the correct node X. Thus $G_n(X|Y)$ is a lower bound for the computations performed to decode the first n vectors $\mathbf{x}_i$ of the transmitted sequence denoted by X. Let X and Y be connected by a discrete memoryless channel whose channel transition probability is denoted by V_n as

$$V_n(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} P(\mathbf{y}_i|\mathbf{x}_i) \tag{2.5}$$

where $\mathbf{y}$ and $\mathbf{x}$ denote the channel output and input sequences formed by a total of n vectors $\mathbf{y}_i$ and $\mathbf{x}_i$. Then the moments of computations are lower-bounded by

$$E[G_n(X|Y)^\rho] \geq (1 + nR_c)^{-\rho} \exp\{n[\rho R - E_0(\rho)]\} \tag{2.6}$$

where,

$$E_0(\rho) = max_Q \left(-\ln \sum_{\mathbf{y}} \left[\sum_{\mathbf{x}} Q_n(\mathbf{x}) V_n(\mathbf{y}|\mathbf{x})^{\frac{1}{1+\rho}} \right]^{1+\rho} \right) \tag{2.7}$$

and $Q_n(\mathbf{x})$ denotes the joint distribution of the channel input $\mathbf{x}$.

Thus at rates $R_c > E_0(\rho)/\rho$, the ρ^{th} moment of computation performed at level n of the code tree must go to infinity exponentially as n is increased. The infinimum of all real numbers R_c' such that for $R_c > R_c'$, $E[G_n(X|Y)^\rho]$ goes infinity as n, is called the cut-off rate for the sequential decoder, R_f. The cut-off rate is given by [6]

$$R_f = E_0(\rho)/\rho \ , \forall \rho > 0 \tag{2.8}$$

2.2 Joint Source-Channel Coding, Lossless Case

Koshelev [2] and Hellman [1] studied joint source-channel encoding using tree codes. Hellman [9], [1] showed that the natural redundancy of a source could be used for data compression by the use of error propagating sources. He gives written English as an example of a source with high natural redundancy. Suppose the message "I AM NOT ABLE TO PROVIDE..." is encoded by a convolutional encoder of rate higher than one. Some letters (or bits) are not sent and will have to be guessed. Assume that a guess is wrong or channel distorts the signal, then an error occurs and the 'T' of 'NOT' is received as 'W'. But then the

rest of the message will appear garbled:'I AM NOWJ.NXAAVWM,EWTY,R...' . Once the error is safely detected we can guess until the correct guess for the unsent or distorted bit is found. Hellman shows that the upper bound of compression for JSC coding system with a MAP decoder is the entropy of the source. Hellman also proposes to use a sequential decoder for the JSC coding system.

Koshelev [2] uses the natural redundancy of the source for direct encoding of the source into channel with a convolutional encoder, and direct decoding of the received sequence into source symbols, i.e. a convolutional encoder was used at the transmitter and a sequential decoder was used at the receiver for lossless recovery of the source output sequence. In order to review his results let us introduce the following notation.

Let $\mathbf{U}_n = (U_1, \ldots, U_n)$ be the vector representing the source output, $\mathbf{X}_k = (X_1, \ldots, X_k)$ be the channel input vector that $\mathbf{U}_n$ is encoded into, and $R_c = n/k$. $\mathbf{Y}_k = (Y_1, \ldots, Y_k)$ is the channel output sequence , P_n denotes the joint distribution of $\mathbf{U}_n$, Q denotes the distribution of the channel input, and define $G_n(U|Y)$ as the number of guesses made for succesfully decoding the source vector $\mathbf{U}_n$.

Koshelev's [2] results are generalized for the class of Markovian sources and shows that the first moment of the number of computations for decoding first n bits, $E(G_n(U|Y))$ is bounded, if asymptotically as $n \to \infty$;

$$H_{1/2}(\mathbf{U}_n)/n < \frac{E_0(1)}{R_c}, \tag{2.9}$$

where $H_{1/2}(\mathbf{U}_n)$ is the Rényi entropy of order $1/2$ for the source vector $\mathbf{U}_n$ and $E_0(1)$ is the computational cut-off rate of the channel for a sequential decoder (Eqns. 2.7, 2.8). We shall give here a more detailed explanation for Rényi entropy.

Rényi entropy : Rényi entropy (per source letter) of order α for a discrete memoryless source U is defined as:

$$H_\alpha(U) = \frac{1}{1-\alpha} \log \sum_{u \in \mathcal{U}} P(u)^\alpha, \tag{2.10}$$

where u denotes source output symbols and $\mathcal{U}$ is the set of all possible source output symbols u.

If we take the limit as $\alpha \to 1$, we obtain the Shannon entropy (per source

letter),

$$H(U) = H_1(U) = -\sum_{u \in \mathcal{U}} P(u) \log P(u) \tag{2.11}$$

and if we take the limit as $\alpha \to 0$ we obtain the logarithm of the size of the set $\mathcal{U}$,

$$H_0(U) \;=\; \log |\mathcal{U}| \tag{2.12}$$

The bevahiour of $H_\alpha(U)$ with respect to α is shown in Fig. 2.1.

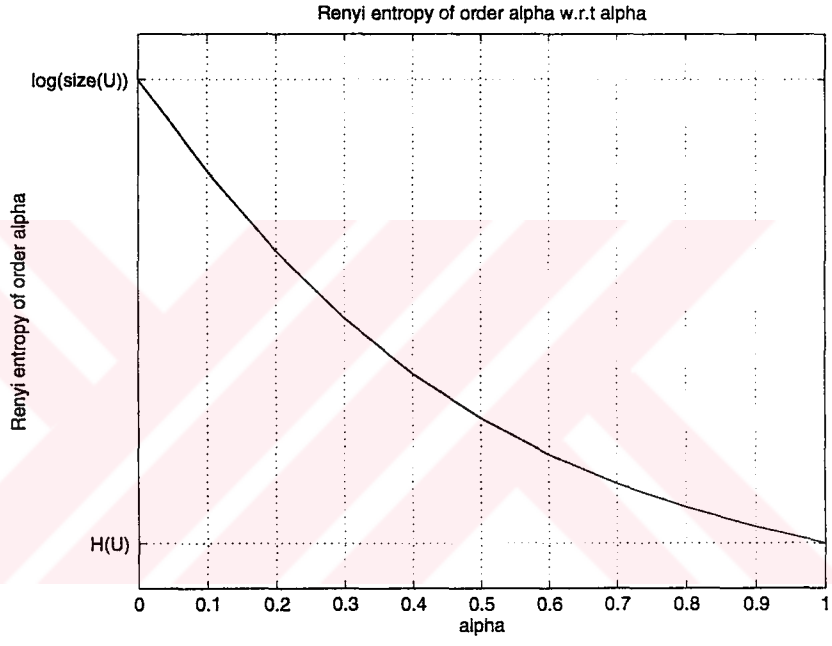

Figure 2.1: Rényi entropy of order α, $H_\alpha(U)$ versus α.

Let P_n be the joint probability of first n bits of the source sequence $\mathbf{U}$. If the source U is memoryless, due product form of P_n, for any n:

$$H_\alpha(U) \;=\; H_\alpha(\mathbf{U}_n)/n \frac{1}{1-\alpha} \log \sum_{\mathbf{u}_n \in \mathcal{U}^n} P_n(\mathbf{u}_n)^\alpha, \tag{2.13}$$

where $\mathbf{u}_n$ denotes vectors of source symbols u, and $H_\alpha(\mathbf{U}_n)$ is the Rényi entropy per block of n source letters. If the source U has memory, then P_n does not have a product form and:

$$H_\alpha(U) \;=\; \lim_{n \to \infty} H_\alpha(\mathbf{U}_n)/n \tag{2.14}$$

As a special case, which will be faced in this thesis' report, when the source is not memoryless, but source vectors of finite lenght N, $\mathbf{U}_N = (U_1, \ldots, U_N)$ with joint probabilities P_N are statistically independent, then even if joint

probability P_N of a vector $\mathbf{U}_N$ is not in product form, the Rényi entropy per source symbol can be given as,

$$H_\alpha(U) = H_\alpha(\mathbf{U}_N)/N \frac{1}{1-\alpha} \log \sum_{\mathbf{u}_N \in \mathcal{U}^N} P_N(\mathbf{u}_N)^\alpha, \tag{2.15}$$

Arıkan and Merhav [3] proves a converse result complementing Koshelev's, not only for $\rho = 1$ but also for higher moments of of $G_n(U|Y)$. They showed that for any discrete source with a possibly non-memoryless probability mass function P_n for first n source letters, and for any $\rho \geq 0$ there exists a lossless guessing function $G_n(U|Y)$ such that,

$$E[G_n(U|Y)^\rho] \leq c(\rho) \exp\left\{ n \left[\rho H_{\frac{1}{1+\rho}}(\mathbf{U}_n)/n - E_0(\rho)/R_c \right]^+ \right\} \tag{2.16}$$

and

$$E[G_n(U|Y)^\rho] \geq \exp\left\{ n \left[\rho H_{\frac{1}{1+\rho}}(\mathbf{U}_n)/n - E_0(\rho)/R_c - o(n) \right]^+ \right\} \tag{2.17}$$

where $[x]^+ = max\{0, x\}$ and $c(\rho)$ is constant with respect to n.

Equations 2.16 and 2.17 implies that, for a source not necessarily memoryless, with joint distribution P_n, the ρ^{th} moment of the guessing function $G_n(U|Y)$ will be bounded if asymptotically as $n \to \infty$;

$$H_{\frac{1}{1+\rho}}(\mathbf{U}_n) < n \frac{E_0(\rho)/\rho}{R_c} \tag{2.18}$$

and conversely ,the ρ^{th} moment of the guessing function $G_n(U|Y)$ must increase exponentially with n if asymptotically as $n \to \infty$;

$$H_{\frac{1}{1+\rho}}(\mathbf{U}_n) > n \frac{E_0(\rho)/\rho}{R_c} \tag{2.19}$$

2.3 Joint Source-Channel Coding, Lossy Case

Arıkan and Merhav [3] generalizes their results , to the case when some certain distortion is introduced to the source data for further compression. They show that for a Discrete Memoryless Source with a probability mass function P , the ρ^{th} moment of amount of computation by a sequential decoder, to generate a

D-admissible reconstruction of first n source letters , must grow exponentially with n if;

$$E(D,\rho) \;>\; \frac{E_0(\rho)}{R_c}, \tag{2.20}$$

where;

$$E(D,\rho) \;=\; max_Q\left[\rho R(D,Q) - D(Q||P)\right], \tag{2.21}$$

$R(D,Q)$ is the Rate Distortion Function of a source with Probability Mass Function Q, and $D(Q||P)$ is the relative entropy between two probability mass functions Q and P, which is defined as:

$$D(Q||P) \;=\; \sum_{u\in\mathcal{U}} \log\frac{Q(u)}{P(u)} \tag{2.22}$$

They also conjecture that a direct result complementing 2.20 can be proved.

One of the important implications of the results of [2] and [3] is, for a system employing tree coding and sequential decoding, for which the first moment of number of computations is bounded independently of n , the rate of the convolutional encoder R_c is bounded with;

$$R_c \;<\; \frac{nE_0(1)}{H_{1/2}(\mathbf{U}_n)} \tag{2.23}$$

and R_c, which may well be above 1, is actually the compression ratio of the joint source-channel encoder in terms of source letters per channel letters.

2.4 Entropy of Written English

We aim to test the compression capabilities of lossless JSC coding system with sequential decoder, using written English as an example for a source with a high natural redundancy. According to the results of Hellman [1] , Koshelev [2], Arıkan and Merhav [3] the compression ratio for any source is bounded by the Rényi entropy of the source (Eqns. 2.18,2.19). In order to obtain estimates for Rényi entropy of written English we refer to the work of Shannon on estimating entropy of written English.

Shannon [7] shows a method to estimate the entropy and redundancy of written English. His method depends on experimental results in prediction of the next letter when J preceding letters are known. He estimates upper

and lower bounds on entropy of English using the results of the experiments. Shannon's experiments can be explained as follows:

A short passage unfamiliar to the person to do the predicting is selected. Subject (the person to do the predicting) is then asked to guess the first letter in the passage. If he/she is wrong he/she is asked to guess again. If true than he/she is asked to guess the second letter using his knowledge about the first, i.e:subject knows the text up to the current point and asked to guess the next. This goes on until the last letter of the text is guessed correctly. For each letter the number of guesses is recorded.

If each letter of sample text is replaced by number of guesses for that letter, the sequence obtained, the *guess sequence* will have the same information. Utilizing an identical twin of our subject(identical in thoughts) we ask him/her in each state to guess as many times as given in the guess sequence, and recover the original text.

In order to determine how predictability depends on the number J of preceding letters known to the subject Shannon carried out a more extensive experiment. One hundred samples of English text, each fifteen letters in length, were chosen. The subject was required to guess the text letter by letter for each sample as in the previous experiment. That way, one hundred samples were obtained in which the subject (the same person for all 100 samples) had available 0,1,...,14 preceding letters. The subject made use of any reference about the statistics and rules of English language he/she wished.

The results of Shannon's experiment from [7] is given in Table 2.1. The column corresponds to the number of preceding letters known plus one .The row indicates number of guesses (maximum number of guesses is 27, for 26 letters of alphabet ignoring capitals, and a space). The entry in column J at row S is the number of times the subject found the correct letter at the S^{th} guess when $J-1$ letters were known (J-Gram prediction). The first and second columns were not obtained by the experiments explained above but were calculated by Shannon, directly from the known letter and digram frequencies.

The data of Table 2.1 can be used to obtain upper and lower bounds to the *J-Gram entropies* F_J. J-gram entropy is the amount of information or entropy due the statistics extending over J adjacent letters of text. F_J is given by,

$$F_J = -\sum_{i,j} p(b_i, j) \log_2 p(j|b_i) \tag{2.24}$$

in which b_i is a block of $J-1$ letters, and j is an arbitrary letter following b_i.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	18.2	29.2	36	47	51	58	48	66	66	67	62	58	66	72	60
2	10.7	14.8	20	18	13	19	17	15	13	10	9	14	9	6	18
3	8.6	10.0	12	14	8	5	3	5	9	4	7	7	4	9	5
4	6.7	8.6	7	3	4	1	4	4	4	4	5	6	4	3	5
5	6.5	7.1	1	1	3	4	3	6	1	6	5	2	3	0	0
6	5.8	5.5	4	5	2	3	2	0	0	1	4	2	3	4	1
7	5.6	4.5	3	3	2	2	8	0	1	1	1	4	1	0	4
8	5.2	3.6	2	2	1	1	2	1	1	1	1	0	2	1	3
9	5.0	3.0	4	0	5	1	4	0	2	1	1	2	0	1	0
10	4.3	2.6	2	1	3	0	3	1	0	0	0	0	2	0	0
11	3.1	2.2	2	2	2	1	0	0	1	3	0	1	1	2	1
12	2.8	1.9	4	0	2	1	1	1	0	0	2	1	1	0	1
13	2.4	1.5	1	1	1	1	1	1	1	1	0	1	1	0	0
14	2.3	1.2	0	1	0	0	1	0	0	0	0	1	0	0	0
15	2.1	1.0	1	1	0	0	0	0	0	0	1	1	1	0	0
16	2.0	0.9	0	0	0	0	1	0	0	1	0	0	0	0	1
17	1.6	0.7	1	0	2	1	1	0	0	0	1	0	2	2	0
18	1.6	0.5	0	0	0	0	0	0	0	0	0	0	0	0	1
19	1.6	0.4	0	0	1	1	0	0	1	0	1	0	0	0	0
20	1.3	0.3	0	1	0	1	1	0	0	0	0	0	0	0	0
21	1.2	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0.8	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0.3	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0.1	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.1: Results of Shannon's prediction experiment.

As J is increased F_J includes longer range statistics, and the entropy H^E is given by the limiting value of F_J as $J \to \infty$.

$$H^E = \lim_{J\to\infty} F_J \tag{2.25}$$

The process of reducing a text to a *guess sequence* with an ideal predictor(which will make guesses in the same order for two different cases with equal preceding J-length-texts) consists of a mapping of the letters into the numbers from 1 to 27 in such a way that the most probable next letter (conditional on the preceding (J-1) Gram $i_1, \ldots, i_{J-1}$) is mapped to 1, etc. The frequencies of 1's in guess sequence will then be :

$$q_1^J = \sum p(i_1, \ldots, i_{J-1}, j) \tag{2.26}$$

where the sum is taken over all (J-1) Grams $i_1, \ldots, i_{J-1}$ and the j being the

letter which maximizes $p(i_1, \ldots, i_{J-1}, j)$ for that particular (J-1) Gram. Similarly the frequencies of 2's, q_2^J, is given by the same formula with j chosen to be that letter having the second highest value of $p(i_1, \ldots, i_{J-1}, j)$, etc. Shannon showed that if we know the frequencies of the symbols in the *guess sequence* with an ideal J-Gram predictor, q_i^J , the upper and lower bounds on the J-Gram entropy ,F_J is given by:

$$\sum_{i=1}^{27} i(q_i^J - q_{i+1}^J) \log i \leq F_J \leq -\sum_{i=1}^{27} q_i^J \log(q_i^J) \tag{2.27}$$

Chapter 3

JOINT SOURCE-CHANNEL CODING OF ENGLISH TEXT

Hellman [9], [1] proposes to use the natural redundancy of the source sequence in order to detect and correct errors, and compress the data, and gives the natural redundancy of written English as an example. The joint source channel coding method proposed in [1] utilizes the knowledge of source statistics for decoding the channel output into source data. One of the main results of [1] is that compression is bounded with the entropy of the source. However derivations for a decoder utilizing the statistics of the source is limited to that of sources with simple statistics. Koshelev [2] also proposes a JSC coding system with a sequential decoder at the receiver and use a modified metric for utilizing source statistics of Markovian sources. In this chapter we aim to search for the methods and limits to compress written English (as a source with high natural redundancy) using a JSC coding scheme and a sequential decoder.

Suppose that a piece of English text is to be sent via a binary memoryless channel with bit error probability P_e. Ignoring punctuation, paragraphs ,etc. Since there are 27 letters (including the 'space' between words) each letter can be represented by blocks of 5 bits, $\mathbf{u}_i$. Let the resulting bit sequence $\mathbf{U} = (\ldots, \mathbf{u}_i, \mathbf{u}_{i+1}, \ldots)$ be encoded by a convolutional encoder, in blocks of n bits where n is an integer multiple of 5, into a binary symmetric channel (BSC) with the channel input sequence $\mathbf{X} = (\ldots, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots)$. Let the length of $n/5$ channel input vectors $\mathbf{x}_i$ be k. The rate of the convolutional encoder is then, $R_c = n/k$. Let the received sequence $\mathbf{Y} = (\ldots, \mathbf{y}_i, \mathbf{y}_{i+1}, \ldots)$ be decoded by a sequential decoder which utilizes the statistics of English for guessing the correct path in the code tree, and therefore decoding the channel output into

the bit sequence **U** free of error.

It is shown in [3] that the rate of the convolutional encoder is bounded with,

$$R_c \leq \frac{E_0(1)}{H^E_{1/2}/5} \tag{3.1}$$

where $E_0(1)$ is the cut-off rate R_f of the sequential encoder for the BSC and $H^E_{1/2}$ is the Rényi entropy of written English in bits per letter (hence divided by 5 to obtain Rényi entropy per bit of bit sequence **U**). However the statistics of English language are extremely complex when the correlation between letters, words, and even paragraphs are considered, and it is not possible to compute the entropy or the Rényi entropy of English directly from its statistics, nor is it possible to extract the statistics perfectly and utilize them with a sequential decoder. Still we can make use of lower order statistics of English, for example, probabilities of letters given the previous J-1 letters,i.e. J-Gram statistics. We can also estimate bounds for the Rényi entropy of English. In order to estimate bounds of compression, we take Shannon's results for bounds on entropy of written English [7] as reference. We will now focus on a compression bound estimate for written English.

3.1 Estimates for Rényi Entropy of Written English

Based on the results of Koshelev [2], and Arıkan and Merhav [3] on JSC coding systems, we aim to estimate the Rényi Entropy upper bound which will give an upper bound for the compression according to Eqn. 3.1. We seek to estimate the upper bound of the Rényi Entropy as a worst case estimate, since any estimate for $R_{c_{max}}$ found according to Eqn. 3.1 with an estimate for the upper bound of $H^E_{1/2}$ will have a lower value than the actual $R_{c_{max}}$ found according to Eqn. 3.1 with the actual value of $H^E_{1/2}$.

Shannon [7] showed that the ideal J-Gram prediction collapses the probabilities of various symbols to a small group, better than any other translating operation involving the same number of symbols. The upper bound for J-Gram entropy F_n given in Eqn. 2.27, follows directly from the fact that the maximum possible entropy in a language with letter frequencies q_i^J is $-\sum_i q_i^J \log(q_i^J)$. Thus the J-Gram entropy per symbol of the guess sequence is not greater than

this value and the J-Gram entropy of guess sequence is equal to that of the original text.

It is a corollary of Shannon's result that the Rényi entropy of the J-Gram guess sequence is also equal to that of the original text sequence. And similar to the case for entropy, the maximum possible Rényi entropy of a language with letter frequencies q_i^J is $\log(\sum_i \sqrt{q_i^J})^2$. Then the Rényi entropy of guess sequence is not greater than this value. Define the J-Gram Rényi entropy of English text as T_J, then Shannon's results can be extended for T_J as,

$$T_J \leq \log \left[\sum_{i=1}^{27} \sqrt{q_i^J} \right] \tag{3.2}$$

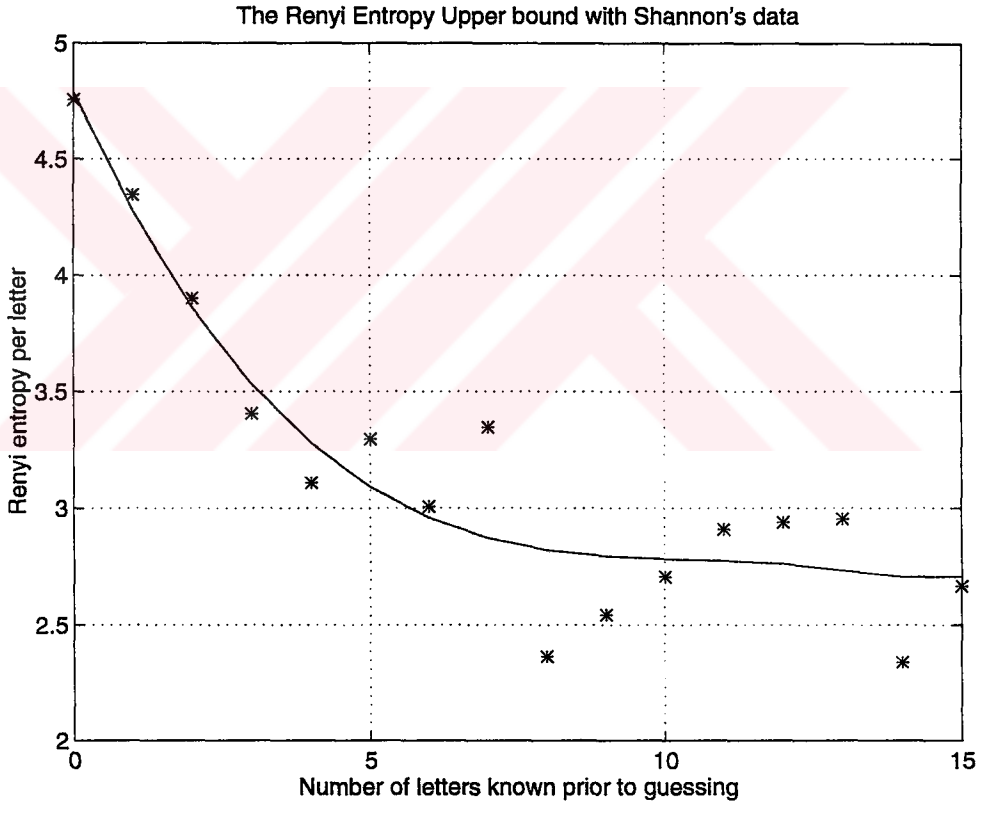

Figure 3.1: Estimates for upper bound of J-Gram Rényi entropy of English $J = 1, \ldots, 15$.

The upper bound estimates for J-Gram Rényi entropy of English, for $J = 1, \ldots, 15$ is given in Fig. 3.1 using the results of Shannon's experiment given in Table 2.1.

3.2 Lossless JSC Coding of Written English

We aim to encode and transmit text written in English, with a JSC coding system using J-Gram statistics derived from a training text. However, for large J, extraction of J-Gram statistics and utilizing them with a sequential decoder is not practical. We propose instead a sequential decoder which will check every path it traces for corresponding words. If the letter sequence corresponding to a path in the code tree is not composed of the words that exist in the dictionary of the sequential decoder, that path and out-branching paths are truncated. In other words, sequential decoder performs a spell check on its estimates along the code tree paths. The dictionary of the sequential decoder must consist all words that are in the transmitted text.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Unigram	19.1	9.9	6.9	6.6	6.0	5.9	5.8	5.2	5.2	4.2	4.2	3.9	2.3	1.9
Digram	28.8	15.8	11.1	7.9	6.7	5.6	4.5	3.7	2.9	2.4	2.1	1.6	1.4	1.2

	15	16	17	18	19	20	21	22	23	24	25	26	27
Unigram	1.9	1.8	1.8	1.7	1.7	1.2	1.1	0.7	0.7	0.1	0.1	0.1	0.1
Digram	1.0	0.8	0.7	0.6	0.5	0.3	0.2	0.2	0.1	0.1	0.0	0.0	0.0

Table 3.1: Extracted unigram and digram letter frequencies.

In order to compare the results of dictionary aided JSC coding method, we also perform unigram and digram JSC coding. We extract unigram probabilities $p(i)$ for $i = 1, \ldots, 27$ and digram statistics $p(i|j)$ for $i, j = 1, \ldots, 27$ from a text of approximately 26000 letters (a short story by James Joyce, "A Little Cloud"), where $p(i)$ denotes the frequency of i^{th} letter and $p(i|j)$ denotes frequency of i^{th} letter following j^{th} letter. Letter frequencies of corresponding guess sequences (per 100 letters) is found using Eqn. 2.26 and is given in Table 3.1. Values are comparable to those found by Shannon (Table 2.1.

3.2.1 Unigram and Digram Decoding

Unigram and digram statistics depend on letter and letter pair frequencies which can be conveniently extracted from a training text and can be easily utilized by a sequential decoder using a modified version of Fano metric (Eqn. 2.4).

Block diagram of simulated unigram and digram JSC coding systems is

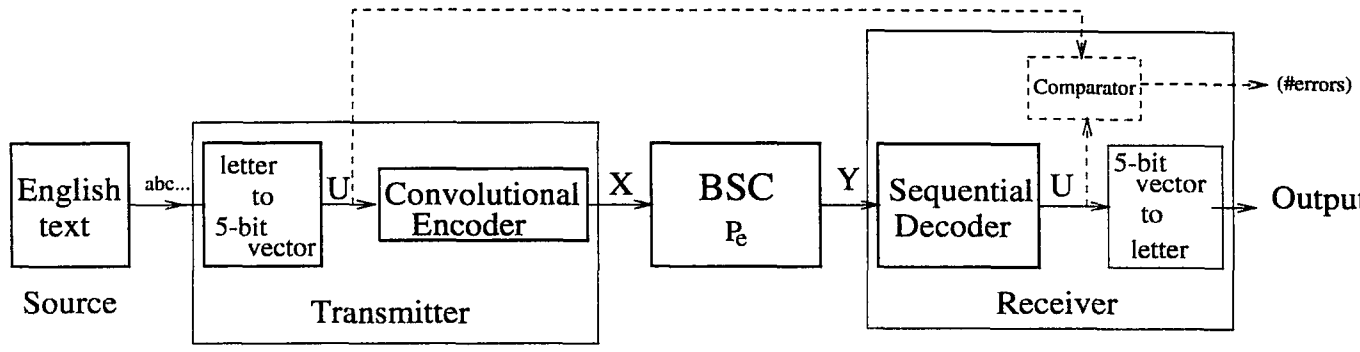

Figure 3.2: Block diagram of the system, simulated for digram and unigram JSC coding.

given in Fig. 3.2. Letters in the text are encoded into 5 bit vectors $\mathbf{u}_i$. The resulting bit sequence $\mathbf{U}$ is encoded by the convolutional encoder into channel input sequence $\mathbf{X}$ with rate R_c. Bit error rate of the binary symmetric channel is demoted by P_e. The output of the channel, $\mathbf{Y}$, is received by the sequential decoder and decoded to reconstruct $\mathbf{U}$. The comparator compares the original sequence $\mathbf{U}$ and the output of the sequential decoder.

In order to avoid encoding large blocks of bits and increasing the complexity of sequential encoder we use the *puncturing* method. The 5 bit vectors $\mathbf{u}_i$ are encoded into variable length channel input vectors $\mathbf{x}_i = (x_{i1}, \ldots, x_{ik(i)})$. k(i) is periodic and completes a full cycle for a total of n encoder input bits, where n is an integer multiple of 5. Let $m = n/5$ and $k = \sum_{i=1}^{m} k(i)$ then the rate of the convolutional encoder is given by $R_c = n/k$.

The metric of the sequential decoder is modified to utilize the unigram and digram statistics. For any node of code tree at level n, the unigram metric is given by the equation:

$$M_n = \sum_{i=1}^{n} \left(\log \frac{P(\mathbf{y}_i|\mathbf{x}_i)}{P(\mathbf{y}_i)} + \log p(\mathbf{u}_i) \right) \tag{3.3}$$

and the digram metric is,

$$M_n = \sum_{i=1}^{n} \left(\log \frac{P(\mathbf{y}_i|\mathbf{x}_i)}{P(\mathbf{y}_i)} + \log p(\mathbf{u}_i|\mathbf{u}_{i-1}) \right) \tag{3.4}$$

where $\mathbf{u}_i$ is the estimate for the n^{th} letter in the text according to the node for which the metric is calculated for, and $\mathbf{x}_i$ is the estimate for the corresponding convolutional encoder output. $p(\mathbf{u}_i)$ and $p(\mathbf{u}_i|\mathbf{u}_{i-1})$ denotes the unigram and digram statistics for the letters represented by 5 bit vectors $\mathbf{u}_i$. $P(\mathbf{y}_i)$ is given by Eqn. 2.2.

Unigram Rényi entropy estimate of English using the data from Table 3.1 and Eqn. 3.2 is obtained as $T_1 = 4.3436$, and the maximum rate of the convolutional encoder for unigram JSC, $R^1_{c_{max}}$, by Eqn. 3.1 is:

$$R^1_{c_{max}} = \frac{R_f}{0.8687} \tag{3.5}$$

Similarly, digram Rényi entropy estimate of English is obtained as $T_2 = 3.9263$, and the maximum rate of the convolutional encoder for digram JSC, $R^2_{c_{max}}$, by Eqn. 3.1 is:

$$R^2_{c_{max}} = \frac{R_f}{0.7853} \tag{3.6}$$

3.2.2 Dictionary Aided Decoding

In order to make better use of natural redundancy of written English , we propose a dictionary aided sequential decoder for the JSC coding system.

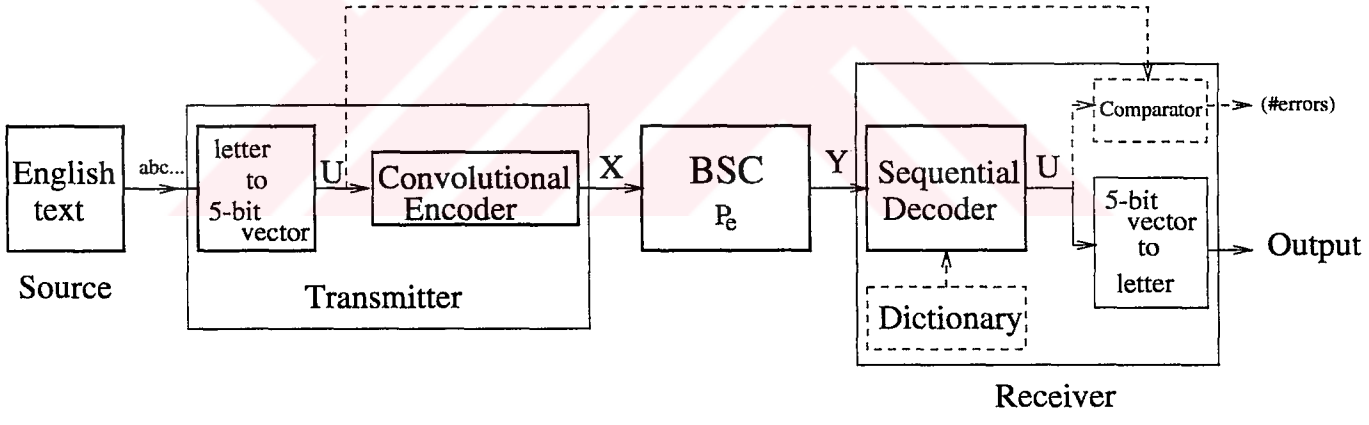

Figure 3.3: Block diagram of the system, simulated for dictionary aided JSC coding.

The block diagram of the system is given in Fig. 3.3. Similar to the digram and unigram case the letters of the text is encoded into 5-bit vectors $\mathbf{u}_i$. All the system is identical to that of Fig. 3.2 except the sequential decoder. Puncturing method is used to avoid encoding large blocks.

The dictionary aided sequential decoding can be explained as follows: Recall the operation of a sequential decoder. It tries to find the transmitted path through a code tree, guided by its metric. In our coding scheme, since every letter is represented by 5 bit vectors $\mathbf{x}_i$ and those vectors are encoded as a whole ,

at each node there are 2^5 outgoing branches and each branch represents a letter (since there are 27 letters 5 of those branches are obsolete, and truncated). Let $\mathcal{A}$ denote the set of all nodes at some fixed but arbitrary level n. Let $\mathcal{B}$ denote a node at level $n-1$, and let $\mathcal{B}$ be on the correct path (transmitted path). Let $A_j(\mathcal{B})$ denote an element of $\mathcal{A}$ that is reached from $\mathcal{B}$ by the j^{th} outgoing branch of $\mathcal{B}$. Let $\mathcal{C}$ denote the last node on the correct path at which a 'space' character was observed and let $\mathcal{C}$ be at level $m \leq n-1$. Node $\mathcal{C}$ clearly marks a word start position. Let $l_{m+1}, \ldots, l_{n-1}$ denote the letters corresponding to the branches on the correct path from node $\mathcal{C}$ to node $\mathcal{B}$. Then $l_{m+1} \ldots l_{n-1}$ clearly forms a prefix or the whole of one of the words in the dictionary. Let l_n^j be the letter corresponding to the j^{th} branch out of node $\mathcal{B}$. Then the metric of node $A_j(\mathcal{B})$ is given by the equation:

$$M_n(A_j(\mathcal{B})) = M_{n-1}(\mathcal{B}) + \log \frac{P(\mathbf{y}_n|\mathbf{x}_n^j)}{P(\mathbf{y}_n)} - R_B + \log\left(I(l_n^j|l_{m+1}, \ldots, l_{n-1})\right) \tag{3.7}$$

where $\mathbf{y}_n$ is the channel output vector corresponding to level n of the code tree and $\mathbf{x}_n^j$ is the vector estimated by the j^{th} branch for the n^{th} level of the transmitted path. R_B is the bias term used to have a finite correct path metric. Value of the bias term in the original Fano metric is equal to the transmission rate R_c, and depends on the $e^{R_c n}$ equally likely paths on the n^{th} level of the code tree (which also means that the entropy per bit for the transmitted sequence is R_c). However, code tree for dictionary aided decoding is truncated, and entropy per bit for the transmitted sequence can be calculated using an estimate of entropy of English and the rate of transmission, R_c. A lower bound for the entropy (in bits per letter) of English is given by Shannon [7] as $H^E \geq 1.2$. When each letter is represented by 5 bits, R_B is calculated as:

$$R_B = R_c * H^E/5. \tag{3.8}$$

Hence the value of the bias used is $R_B = 0.24 * R_c$. $I(l_n^j|l_{m+1}, \ldots, l_{n-1})$ is the indicator function given by,

$$I(l_n^j|l_{m+1}, \ldots, l_{n-1}) = \begin{cases} 1 & \textit{if } l_{m+1} \ldots l_{n-1} l_n^j \textit{ forms a prefix or the whole of one of the words in the dictionary} \\ 0 & \textit{otherwise} \end{cases} \tag{3.9}$$

Although most of the incorrect paths in code tree will be truncated with this metric, there may still be incorrect guesses. Consider the text: 'HE STOOD THERE FOR A MOMENT AND THEN WITH AN ...'. If the 'O' of 'STOOD' is guessed as a 'P' that would still be a word in the dictionary (assuming 'STOP' is in the dictionary). But because of the error propagation property of tree codes the rest will appear garbled like 'HE STOPRJKLMN...'. If the rate of transmission is sufficiently low, the probability that the rest of the message will form words in the dictionary can be made arbitrarily low, by the choice of the constraint length of the convolutional encoder.

The limits for the rate of transmission can be given by means of Rényi entropy of English and Eqn. 3.1. The performance of the dictionary aided sequential decoder, would surely be poorer than an ideal predictor, since we are unable to use the complete statistics of English language. Therefore observing the limits of transmission rate for the dic.-aid seq. decoder we can obtain tighter estimates for the Rényi entropy of written English.

3.2.3 Simulations

We have simulated the unigram,digram and dictionary aided JSC coding systems for both noise free and noisy channels. Unigram and digram statistics were extracted from a short story by James Joyce, 'A Little Cloud' , and the transmitted text is taken from a novel by Agatha Christie, 'The Murder on the Links'. All punctuation, paragraphs and capitals are ignored in the transmitted text which is 1000 letters long (including spaces). Half of the dictionary of the dic. aided sequential decoder consists of all the words in the transmitted text and the other half consists of arbitrary words from 'A Little Cloud'.

Noise Free Channel

R_f	T_1	$R_{c_{max}}$	R_c	# *search*	# *errors*
1.00	4.34	1.15	1.00	32.00	0 bits
1.00	4.34	1.15	1.11	48.64	0 bits
1.00	4.34	1.15	1.25	>400	NO SOL.

Table 3.2: Table of results for unigram JSC coding Simulations. Noise free channel.

R_f	T_2	$R_{c_{max}}$	R_c	# search	# errors
1.00	3.93	1.27	1.00	32.00	0 bits
1.00	3.93	1.27	1.11	39.84	0 bits
1.00	3.93	1.27	1.25	66.91	0 bits
1.00	3.93	1.27	1.43	>400.	NO SOL.

Table 3.3: Table of results for digram JSC coding Simulations. Noise free channel.

R_f	R_c	# search	# errors
1.00	1.00	32.00	0 bits
1.00	1.11	33.57	0 bits
1.00	1.25	36.10	0 bits
1.00	1.43	42.14	0 bits
1.00	1.67	54.24	0 bits
1.00	1.88	78.02	0 bits
1.00	2.00	95.52	0 bits
1.00	2.14	>400	NO SOL.

Table 3.4: Table of results for dictionary aided JSC coding Simulations. Noise free channel.

Tables 3.2-3.4 lists the simulation results for, unigram, digram and dictionary aided JSC coding systems. R_f is the cut-off rate of the channel given by Eqns. 2.8,2.7, which is '1.0' for a noise free channel. The second column gives the J-gram Rényi entropies T_J for unigram and digram simulations. $R_{c_{max}}$ is the estimated upper bound for the rate of convolutional encoder and is computed according to Eqn. 3.1. R_c is the rate of the convolutional encoder in the simulations. '# search' is the number of branches searched by the sequential decoder per letter, and '# errors' is the number of bits in error, counted by the comparator. The minimum number for # search is 32.00 since there are $2^5 = 32$ branches outgoing from any node in code tree. The program terminates when # search exceeds 400.0 (for example, if only 135 letters are decoded with 80000 branches searched, which means sequential decoder is stuck and making exhaustive search while trying to decode the channel output, the program terminates) . Termination of the program is indicated by 'NO SOL.' in # errors column.

Fig. 3.4 compares # search versus R_c curves for unigram, digram and dictionary aided simulations. Simulation results for unigram and digram cases agree with the estimates T_J. Performance of dictionary aided JSC coding system seems to be the best, and observations of the results gives us the best

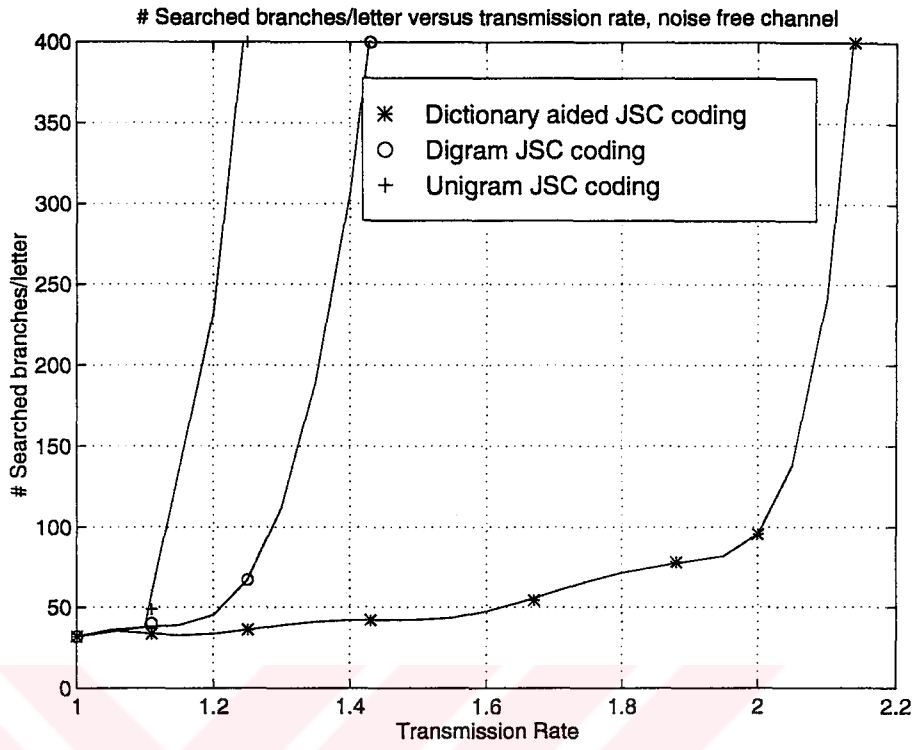

Figure 3.4: Number of searches versus R_c curves for unigram, digram, dictionary aided JSC coding system simulations. Noise free channel

guess for the maximum transmission rate as $R_{c_{max}} = 2.00$, which in turn implies a Rényi entropy upper bound of $T_D \leq 2.5$ by Eqn. 3.1.

Noisy Channel

R_f	T_1	$R_{c_{max}}$	R_c	*# search*	*# errors*
0.50	4.34	0.58	0.50	50.62	0 bits
0.50	4.34	0.58	0.56	61.25	0 bits
0.50	4.34	0.58	0.63	>400	NO SOL.

Table 3.5: Table of results for unigram JSC coding Simulations. Noisy channel.

R_f	T_2	$R_{c_{max}}$	R_c	*# search*	*# errors*
0.50	3.93	0.64	0.50	40.19	0 bits
0.50	3.93	0.64	0.56	38.88	0 bits
0.50	3.93	0.64	0.62	57.66	0 bits
0.50	3.93	0.64	0.71	180.74	0 bits
0.50	3.93	0.64	0.77	>400	NO SOL.
0.50	3.93	0.64	0.83	>400	NO SOL.

Table 3.6: Table of results for digram JSC coding Simulations. Noisy channel.

R_f	R_c	# *search*	# *errors*
0.50	0.50	45.71	0 bits
0.50	0.62	42.03	0 bits
0.50	0.71	42.46	0 bits
0.50	0.83	41.98	0 bits
0.50	0.91	59.10	0 bits
0.50	1.00	64.86	0 bits
0.50	1.11	80.61	0 bits
0.50	1.25	>400	NO SOL.

Table 3.7: Table of results for dictionary aided JSC coding Simulations. Noisy channel.

Simulations are repeated this time with a noisy channel with bit error probability $P_e = 0.0445$, which results in a cut-off rate of $R_f = 0.5$ by Eqns. 2.8, 2.7. Tables 3.5-3.7 lists the simulation results for, unigram, digram and dictionary aided JSC coding systems with a noisy channel. The order of columns are as in noise free case.

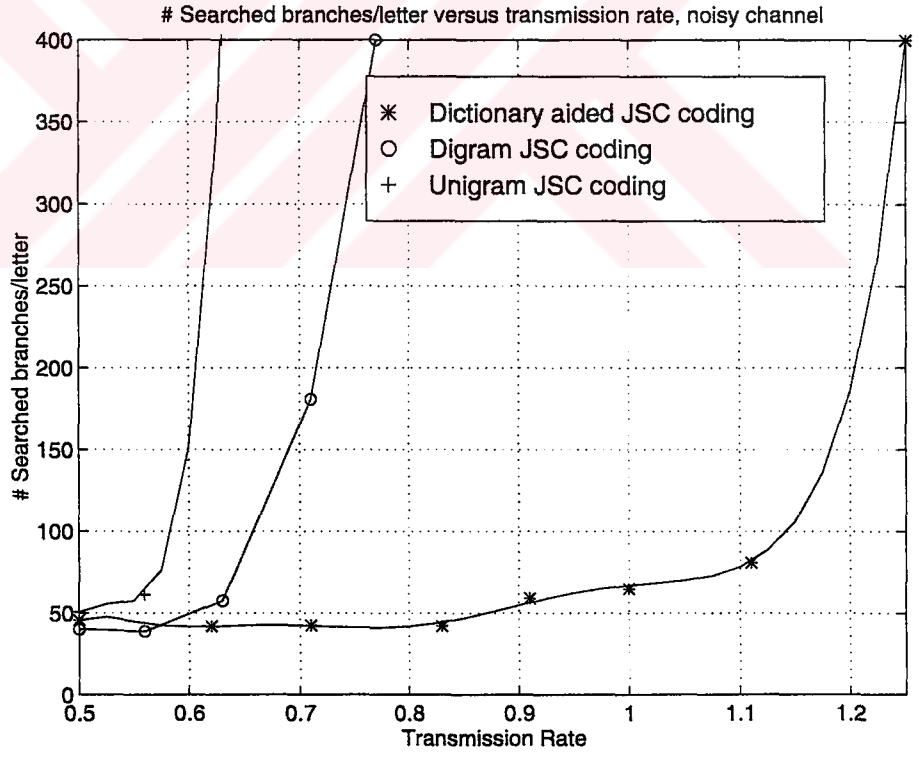

Figure 3.5: Number of searches versus R_c curves for unigram, digram, dictionary aided JSC coding system simulations. Noisy channel.

Fig. 3.5 compares # search versus R_c curves for unigram, digram and dictionary aided simulations. Simulation results for unigram and digram cases agree with the estimates T_J. Performance of dictionary aided JSC coding

system seems to be the best again, and observations of the results gives us the best guess for the maximum transmission rate as $R_{c_{max}} = 1.11$, which in turn implies a Rényi entropy upper bound of $T_D \leq 2.25$ by Eqn. 3.1.

Chapter 4

STOCHASTIC SOURCE ENCODING FOR JSC CODING SYSTEMS

4.1 Source Encoding

Koshelev's [2], Arıkan and Merhav's [3], and Hellman's [1] results give the limit of compression with a convolutional encoder, for lossles recovery of the source output sequence.

Arıkan and Merhav's [3] results also gives the limits of the use of convolutional encoder for compression of data, after the source data sequence is compressed by the use of a Rate Distortion Encoder, at a cost of some distortion.

The process done by the Rate Distortion Encoder (RDE) can be informally explained as altering the source data sequence so as to form a minimum length reconstruction sequence, from which the original data can be reconstructed, with a limited average distortion D_{avg}. This process involves removing of the redundancy present intrinsically in the source data. Rate Distortion Encoder ,as a design criterion, is optimized in the sense that the encoder generates a reconstruction sequence of minimum length, when it is constrained by a certain maximum average distortion at the reconstruction, with some finite

complexity. The Rate Distortion Function, $R(D)$ gives the asymptotic lower bound ,when block length N of the encoder (and so the complexity of the encoder), approaches infinity.

Any RDE-N with finite N would be sub-ideal and will provide sub-optimal compression above $R(D)$. This also means that the intrinsic redundancy of the source data will not be removed totally. The residual redundancy can be used for further compression by Joint Source-Channel Coding. [3] gives the bounds (Eqn. 2.20) on the use of JSC coding for this case, and [10] analyses such a system ,but for increased error-correction capability instead of increased allowable transmission rate.

However, practical use of JSC coding with distortion a constraint D_C in [3] and [10] requires use of a RDE-N ,even though with low N, that is able to trade compression with a certain average distortion.

For certain systems, one of the major design criteria might be the simplicity of the transmitter, rather than the efficiency of the overall system. Joint Source-Channel Encoding methods can be of great use for such systems, except that the use of Rate Distortion Encoder for accuracy-compression trade-off would shadow the simplicity of transmitter. Hence the transmitter requires a low complexity source encoder which would leave most of the compression process to the channel encoder, and would need little hardware. Consider, then, a source encoder which would not attempt to remove the redundancy and compress the data sequence, but would instead leave the original sequence length unchanged and increase the redundancy of it, at a cost of some distortion. A convolutional encoder, used as the encoder of the joint source-channel encoding system, would then be enabled to compress data while forming the channel input, without additional cost to the transmitter. Meanwhile the increase in the redundancy of the original source data sequence provided by the source encoder would increase the maximum compression that can be achieved by the JSC coding.

Proposition 4.1 *Suppose that a Binary Memoryless Source, with probability mass function P (BMS-P) generates output vectors* $\mathbf{U}_N = (U_1, \ldots, U_N)$ *to form an output sequence* $\mathbf{U}$ *that is encoded with a source encoder, which systematically alters the input sequence so as to change the statistics of the resulting*

sequence $\hat{\mathbf{U}}$ *to that of a binary source with joint probability* $\hat{P}_N$ *for blocks of* N *bits which are denoted by vectors* $\hat{\mathbf{U}}_N = (U_1, \ldots, U_N)$*, by mapping source output vectors* $\mathbf{U}_N$ *to* $\hat{\mathbf{U}}_N$*. Since the source is memoryless encoder output vectors* $\hat{\mathbf{U}}_N$ *will be statistically independent. Suppose also that this source encoder, during this process, introduces an average distortion no more than* D_C*. Let the output of this source encoder be encoded by a convolutional encoder, in to a Binary Symmetric Channel (BSC) at a rate of* R_c *source symbols per channel symbol, and a sequential decoder be used at the receiver end to reconstruct the distorted information sequence. Let* G_n *be the amount of computations by the sequential decoder to reconstruct first* n *bits (or first* n/N *vectors* $\hat{\mathbf{U}}_N$*) of the output of the source encoder* $\hat{\mathbf{U}}$ *(the distorted sequence) free of error.*

- *Then the first moment of* G_n *grows exponentially with* n *if;*

$$R_c > \frac{E_0(1)}{H_{1/2}(\hat{\mathbf{U}}_N)/N} \tag{4.1}$$

- *Conversely the first moment of* G_n *will be bounded if;*

$$R_c < \frac{E_0(1)}{H_{1/2}(\hat{\mathbf{U}}_N)/N} \tag{4.2}$$

where $E_0(1)$ *is the computational cut-off rate, and* $H_{1/2}(\hat{\mathbf{U}}_N)/N$ *is the Rényi entropy of the source encoder output per source output bit.*

Proof of Proposition 4.1 follows from the results of [2] and [3], which are reviewed in Chapter 2.

Corollary 4.1 *For the system introduced in Proposition 4.1, an optimal source encoder is the one that would maximize* R_c*, subject to a distortion constraint* D_C*. Since* $E_0(1)$ *is related with channel characteristics and independent of distortion or source statistics, maximizing* R_c *is equivalent to minimizing* $H_{1/2}(\hat{\mathbf{U}}_N)/N$ *where* $\hat{P}_N$ *is the joint probability of blocks of* N *bits at the output of the source encoder,* $\hat{\mathbf{U}}_N = (U_1, \ldots, U_N)$*. Hence the design criterion for the proposed source encoder is the optimization:*

$$\min_{\hat{\mathbf{U}}: E[d(\mathbf{U},\hat{\mathbf{U}})] < D_C} H_{1/2}(\hat{\mathbf{U}}_N)/N. \tag{4.3}$$

4.2 Stochastic Distortion Encoder (SDE)

Let us define the *Stochastic Distortion Encoder with memory length N* (SDE-N) as the source encoder introduced in Proposition 4.1 . *Memory length* is the dual of the block length of a Rate Distortion Encoder (RDE), and denotes the size of the source output vectors, that the SDE-N changes the joint statistics of. In other words, SDE-N changes the Joint Probability Mass Function (PMF) of the set of vectors $\mathcal{U}^N = \{\mathbf{U}^N : (U_1, \ldots, U_N)\}$ where U_i are source output letters. Notice that the SDE does not change the length of the data sequence, but only alters the statistics so as to increase redundancy. Compression process is completely left to a convolutional encoder.

For the sake of simplicity, from now on, the source, reconstruction and the channel alphabets are assumed to be binary. To have an insight to the design problem let us work on a SDE-1 encoder.

4.2.1 SDE-1 Encoder Design

Assume a Binary Memoryless Source (BMS-p) with statistics $P(U = 1) = p$ and $P(U = 0) = 1 - p$. Without loosing generality let us assume that $p \leq 0.5$. The output of the encoder is represented by $\hat{U}$ with statistics $P(\hat{U} = 1) = \hat{p}$ and $P(\hat{U} = 0) = 1 - \hat{p}$. Using the hamming distortion measure, let D_C be the distortion constraint, and $D_C \leq p$. The forward test channel in Fig. 4.1 represents the SDE-1 encoder.

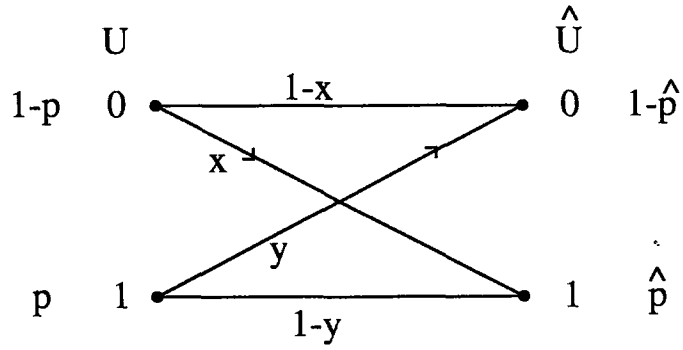

Figure 4.1: Forward Test Channel representing SDE-1; x and y are transition probabilities representing $W(\hat{U} = 1|U = 0)$ and $W(\hat{U} = 0|U = 1)$ respectively, where $W(\hat{U}|U)$ represents the channel statistics.

Define the average distortion, $\overline{d(U, \hat{U})} = D_{avg}$. It is derived from the test

channel that,

$$\begin{aligned} D_{avg} &= (1-p)x + py && (4.4)\\ \hat{p} &= (1-p)x + p(1-y) && (4.5)\\ 1-\hat{p} &= py + (1-p)(1-x) && (4.6) \end{aligned}$$

and $0 \leq y \leq 1$, $0 \leq x \leq 1$. Combining 4.4,4.5 and 4.6;

$$\begin{aligned} \hat{p}(y) &= D_{avg} + p(1-2y) && (4.7)\\ 1-\hat{p}(y) &= (1-p) - D_{avg} + 2yp. && (4.8) \end{aligned}$$

Design criterion was to minimize $H_{1/2}(\hat{U})$ subject to average distortion constraint D_C. Hence we shall minimize $H_{1/2}(\hat{U})$ with respect to y (or x) holding D_{avg} constant. $H_{1/2}(\hat{U})$, the first derivative and second derivative with respect to y is;

$$\begin{aligned} H_{1/2}(\hat{U}) &= \log_2\left(\sqrt{\hat{p}(y)} + \sqrt{1-\hat{p}(y)}\right)^2 \\ &= \log_2\left(\sqrt{D_{avg} + p(1-2y)} + \sqrt{(1-p) - D_{avg} + 2yp}\right)^2 && (4.9)\\ \frac{\partial H_{1/2}(\hat{U})}{\partial y} &= \frac{2\log_2(e)p}{\sqrt{D_{avg} + p(1-2y)} + \sqrt{(1-p) - D_{avg} + 2yp}} \\ &\quad \left(\frac{1}{\sqrt{(1-p) - D_{avg} + 2yp}} - \frac{1}{\sqrt{D_{avg} + p(1-2y)}}\right) && (4.10)\\ \frac{\partial^2 H_{1/2}(\hat{U})}{\partial y^2} &= \frac{-2\log_2(e)p^2}{\left(\sqrt{\hat{p}(y)} + \sqrt{1-\hat{p}(y)}\right)^2}\left(\frac{1}{\sqrt{1-\hat{p}(y)}} - \frac{1}{\sqrt{\hat{p}(y)}}\right)^2 - \\ &\quad \frac{6\log_2(e)p^2}{\left(\sqrt{\hat{p}(y)} + \sqrt{1-\hat{p}(y)}\right)}\left(\frac{1}{(1-\hat{p}(y))^{\frac{3}{2}}} - \frac{1}{(\hat{p}(y))^{\frac{3}{2}}}\right) && (4.11) \end{aligned}$$

Notice that if $D_{avg} \leq p$ then $\forall y \in [0, D_{avg}/p]$, $0 \leq \hat{p}(y) \leq 1$; and $\forall \hat{p}(y) \in [0,1]$ right hand side of Eqn. 4.11 is negative. As expected, $H_{1/2}(\hat{U})$ is concave $\cup$ with respect to y . Hence the minimum must be at one of the endpoints $S1 : y = 0$ or $S2 : y = y_{max}$. The maximum value of y is determined by the Eqn. 4.4 ; $y_{max} = D_{avg}/p$. The results at the two endpoints is,

$$S1: \quad H_{1/2}(\hat{U}) = \log_2\left(\sqrt{p + D_{avg}} + \sqrt{(1-p) - D_{avg}}\right)^2 \tag{4.12}$$

$$S2: \quad H_{1/2}(\hat{U}) = \log_2\left(\sqrt{p - D_{avg}} + \sqrt{(1-p) + D_{avg}}\right)^2 . \tag{4.13}$$

Obviously for $0 \leq D_{avg} \leq p \leq 0.5$, $H_{1/2}(\hat{U})\ |_{y=D_{avg}/2,x=0}$ is smaller and $S2$ is the solution point. Equation 4.13 also shows that the upper bound on transmission rate R_c, being inversely proportional to $H_{1/2}(\hat{U})$, is a monotone increasing function of the average distortion, hence we can conclude the SDE-1 encoder design by taking average distortion equal to the constraint $D_{avg} = D_C$.

The results of the SDE-1 encoder design can be summarized as follows:

- For an average distortion constraint D_C, the forward test channel of Fig. 4.2 represents the operation of the encoder.

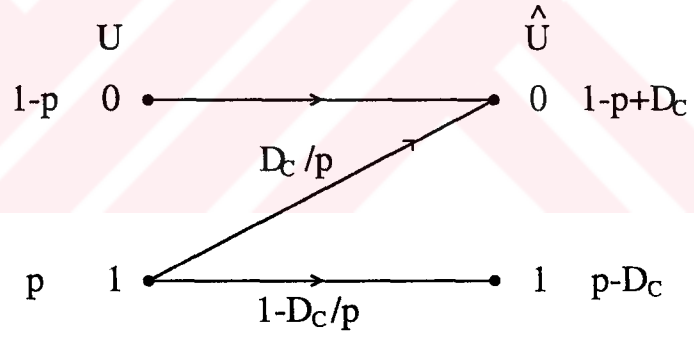

Figure 4.2: Forward Test Channel representing SDE-1 designed for BMS with $P(U = 1) = p$, average distortion constraint D_C and $D_C \leq p \leq 0.5$. Encoder simply spits '0's unchanged and alters '1's to '0's with probability $W(\hat{U} = 0|U = 1) = D_C/p$.

- The upper bound of transmission rate, in terms of source bits per channel bits, is

$$R_c < \frac{E_0(1)}{\log^2\left(\sqrt{p - D_C} + \sqrt{1 - p + D_C}^2\right)} \tag{4.14}$$

The bound is concave $\cup$ and goes infinity as $D_C \to p$ (or equivalently $y \to 1$). That means, compression will be infinite when all '1's are distorted to '0's. It surely makes sense for the entropy of an all zero sequence is zero

and we don't need to send anything for its reconstruction at the receiver side, provided that we know its statistics. As $D_C \rightarrow 0$ the bound is the same with that given by [2].

4.2.2 SDE-N Encoder Design

SDE-1 encoder for a BMS-p is easy to design and implement, and it is also of some use, even though very ineffective. On the other hand a RDE-1 for a BMS-p is obsolete.

Now consider the case when memory length N of SDE is greater than one. Let the encoder input be the vectors $\mathbf{U}_N = (U_1, \ldots, U_N)$ with joint probabilities P_N, and vectors $\hat{\mathbf{U}}_N = (\hat{U}_1, \ldots, \hat{U}_N)$ with joint probabilities $\hat{P}_N$ be the encoder output. Let W_N be the *Forward Test Channel Transition Probability Matrix.* $W_N = \{w_{ij}\}; w_{ij} = P(\hat{U} = C_i | U = C_j);\ i, j = 1, 2, \ldots, 2^N$. Let $C_i = (u_{i_1}, \ldots, u_{i_N}); i = 1, \ldots, 2^N$ represent all possible sequences of length N, which form the reconstruction alphabet $\mathcal{C}$. Assuming hamming distortion measure is used let $Dist(C_i, Cj)$ be the hamming distance between two length-N bit sequences, C_i and C_j.

Define the probability vectors $\mathbf{P}_N$ and $\hat{\mathbf{P}}_N$:

$$\mathbf{P}_N = \{p_i\}_{i=1}^{2^N} \quad ; \quad p_i = P_N \mid_{\mathbf{U}_N = C_i} \tag{4.15}$$

$$\hat{\mathbf{P}}_N = \{\hat{p}_i\}_{i=1}^{2^N} \quad ; \quad \hat{p}_i = \hat{P}_N \mid_{\hat{\mathbf{U}}_N = C_i} \tag{4.16}$$

Then ,

$$\hat{\mathbf{P}}_N \;=\; W_N * \mathbf{P}_N \tag{4.17}$$

and;

$$H_{1/2}(\hat{\mathbf{U}}_N)/N \;=\; \frac{2}{N} \log_2 \left(\sum_{i=1}^{2^N} \sqrt{\hat{p}_i} \right) \tag{4.18}$$

$$D_{avg}(W_N) \;=\; \sum_{j=1}^{2^N} p_j \sum_{i=1}^{2^N} Dist(C_i, Cj). \tag{4.19}$$

Since $\mathbf{P}_N$ is known and $\hat{\mathbf{P}}_N = W_N * \mathbf{P}_N$, the variables of the optimization problem are the transition probabilities w_{ij}, in other words, the elements of

the transition matrix W_N. Notice that $\sum_i w_{ij} = 1, \forall j \in \{1, \ldots, 2^N\}$. Hence the number of independent variables for the optimization problem turns out to be $2^N(2^N - 1)$.

Claim 4.1 *Let $\mathcal{W}_N$ be the set of all possible W_N. Set $\mathcal{W}_N$ forms a convex region $\mathcal{R}$ in 2^{2N} dimensional space. Each element W_N is a point in the region $\mathcal{R}$ and coordinates of any point W_N is denoted by the matrix elements w_{ij} .*

Proof: *Let Λ be a real positive number, $0 < \Lambda < 1$, and $W_{N_3} = \Lambda W_{N_1} + (1 - \Lambda)W_{N_2}$. By the definition of W_N, $0 \leq w_{ij} \leq 1, \forall W_N$. Then if,*

$$w_{ij\,3} = \Lambda w_{ij\,1} + (1 - \Lambda) w_{ij\,2} \tag{4.20}$$

$$then \quad 0 \leq w_{ij\,3} \leq 1 \tag{4.21}$$

$$\Rightarrow \quad W_{N_3} \in \mathcal{W}_N \tag{4.22}$$

Claim 4.2 *$H_{1/2}(\hat{\mathbf{U}}_N)$ is concave over the convex set $\mathcal{W}_N$.*

Proof: Let W_1, W_2 and W_3 be elements of $\mathcal{W}_N$ and points in region $\mathcal{R}$. Let Λ be a real positive number, $0 < \Lambda < 1$, and $W_3 = \Lambda W_1 + (1 - \Lambda)W_2$.

Using 4.17, Eqn. 4.18 can be modified as,

$$H_{1/2}(\hat{\mathbf{U}}_N) = 2\log_2 \sum_{i=1}^{2^N} \sqrt{\sum_{j=1}^{2^N} p_j w_{ij}} \tag{4.23}$$

Assuming it exists, let us find the point W_3 between W_1 and W_2, which will give the minimum value of $H_{1/2}(\hat{\mathbf{U}}_N)|_{W_3}$. Since Λ defines W_3 between W_1 and W_2, we shall take the first and second partial derivatives of $H_{1/2}(\hat{\mathbf{U}}_N)|_{W_3}$, with respect to Λ.

$$H_{1/2}(\hat{\mathbf{U}}_N)|_{W_3} = \log_2 \left(\sum_{i=1}^{2^N} \sqrt{\sum_{j=1}^{2^N} p_j(\Lambda w_{1_{ij}} + (1 - \Lambda) w_{2_{ij}})} \right)^2 \tag{4.2}$$

$$\frac{\partial H_{1/2}(\hat{\mathbf{U}}_N)}{\partial \Lambda} = \frac{2\log_2(e)}{\sum_i \sqrt{\sum_j p_j(\Lambda w_{1_{ij}} + (1 - \Lambda) w_{2_{ij}})}} \left[\sum_i \frac{\sum_j p_j(w_{1_{ij}} - w_{2_{ij}})}{2\sqrt{\sum_j p_j(\Lambda w_{1_{ij}} + (1 - \Lambda) w_{2_{ij}})}} \right] \tag{4.2}$$

$$\frac{\partial^2 H_{1/2}(\hat{\mathbf{U}}_N)}{\partial \Lambda^2} = \frac{-2\log_2(e)}{\left(\sum_i \sqrt{\sum_j p_j(\Lambda w_{1_{ij}} + (1-\Lambda)w_{2_{ij}})}\right)^2} \left[\sum_i \frac{\sum_j p_j(w_{1_{ij}} - w_{2_{ij}})}{2\sqrt{\sum_j p_j(\Lambda w_{1_{ij}} + (1-\Lambda)w_{2_{ij}})}}\right]^2$$
$$+ \frac{-2\log_2(e)}{\sum_i \sqrt{\sum_j p_j(\Lambda w_{1_{ij}} + (1-\Lambda)w_{2_{ij}})}} \left[\sum_i \frac{\left(\sum_j p_j(w_{1_{ij}} - w_{2_{ij}})\right)^2}{4\left(\sum_j p_j(\Lambda w_{1_{ij}} + (1-\Lambda)w_{2_{ij}})\right)^{\frac{3}{2}}}\right] \quad (4.26$$

The second derivative is always non-positive. There is no minimum between W_1 , W_2 and $H_{1/2}(\hat{\mathbf{U}}_N)$ is concave between arbitrary points W_1 , W_2 in region $\mathcal{R}$. Hence $H_{1/2}(\hat{\mathbf{U}}_N)$ is concave over the convex set $\mathcal{W}_N$.

Corollary 4.2 *The solution of the optimization problem (Eqn. 4.3) is then on the boundary* $D_{avg}(W_N) = D_C$.

Definition: Any point W_N of region $\mathcal{R}$, with all its coordinates chosen from a binary set; i.e.: $\forall i,j\ w_{ij} = 0,1$; is a *Vertex* of the region $\mathcal{R}$.

Since $\sum_{i=1}^{2^N} w_{ij} = 1$, any region $\mathcal{R}$ has $(2^N)^{(2^N)}$ vertices. Any point in region $\mathcal{R}$ can be denoted as a linear combination of the vertices of $\mathcal{R}$.

Claim 4.3 *Optimal solution of the optimization problem 4.3 for a set* $\mathcal{W}_N$ *is on a point between two vertices of the corresponding region* $\mathcal{R}$.

Proof: Assume that $W_1 \in \mathcal{W}_N$ is the optimal solution point for the optimization problem (3.3), and let W_1 be denoted by the linear combination of the vertices W_A,W_B,W_C. Let W_A be within the average distortion constraint, W_B and W_A be *out* of the distortion constraint. Then W_1 is in the triangle (ABC) formed by the vertices W_A,W_B,W_C, and is on the constraint boundary. Since the average distortion is linear, the constraint boundary can be denoted by a straight line bisecting the $\triangle(ABC)$ (Fig. 4.3).

W_1 can also be denoted by a linear combination of two points, W_{AB} and W_{AC}, which are on the intersection of the edges of the triangle and the constraint boundary. W_{AB} and W_{AC} are themselves linear combinations of two vertices. But by the concavity of $H_{1/2}(\hat{\mathbf{U}}_N)$ at least one of the points W_{AB},W_{AC} would provide an as good ,or a better solution. Hence, by contradiction, such a point W_1 can not be an optimal solution to the problem.

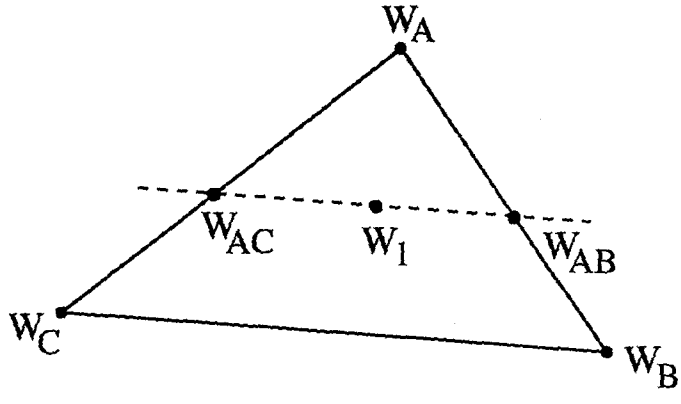

Figure 4.3: The Triangle of three vertices W_A, W_B, W_C . Dashed line is the constraint line.

By induction, it can be seen that, any point $W_N \in \mathcal{W}_N$ which can be denoted only by a linear combination of three or more vertices, can not give the optimal solution.

Claim 4.4 *Any optimal solution for the optimization problem 4.3 from a set $\mathcal{W}_N$ is on a point between two vertices of the corresponding region $\mathcal{R}$, which are at hamming distance 2.*

Note that the hamming distance between any two vertices is never an odd number since, $\sum_i w_{ij} = 1$. If any one element w_{kl} changes from 0 to 1 (or vice-versa), another element (and only one element) $w_{k'l}$ should respond to hold the sum constant.

Proof: Let $W_1 \in \mathcal{W}_N$ be a point in region $\mathcal{R}$, which lies on the line connecting two vertices $W_A, W_B \in \mathcal{W}_N$. Let the hamming distance between W_A and W_B be equal to 4. Then there are two vertices $W_C, W_D \in \mathcal{W}_N$ that are at hamming distance 2 from both of the vertices W_A and W_B. Without loss of generality we can define the four coordinate variables that changes between W_A and W_B as :

$$W_A; w_{A_{kl}} = 1, w_{A_{k'l}} = 0, w_{A_{ml'}} = 1, w_{A_{m'l'}} = 0\,; \tag{4.27}$$

$$W_B; w_{B_{kl}} = 0, w_{B_{k'l}} = 1, w_{B_{ml'}} = 0, w_{B_{m'l'}} = 1\,; \tag{4.28}$$

and other coordinates will be the same for both. According to the definitions of W_C and W_D:

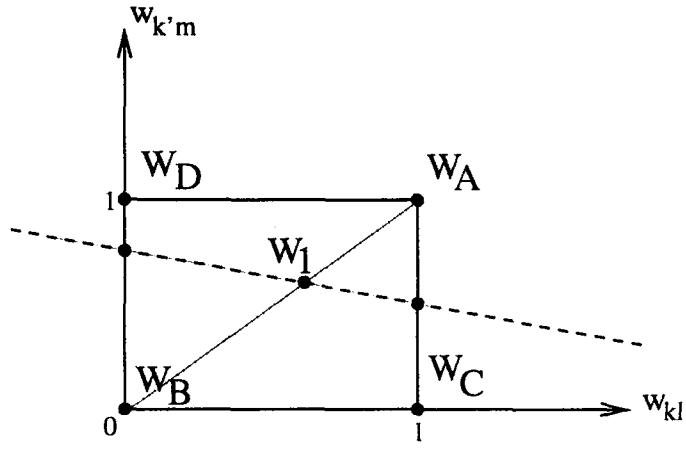

Figure 4.4: The Rectangle of four vertices W_A, W_B, W_C, W_D . Dashed line is the constraint line.

$$W_C; w_{C_{kl}} = 1, w_{C_{k'l}} = 0, w_{C_{ml'}} = 1, w_{C_{m'l'}} = 0\,; \tag{4.29}$$

$$W_D; w_{D_{kl}} = 0, w_{D_{k'l}} = 1, w_{D_{ml'}} = 0, w_{D_{m'l'}} = 1\,; \tag{4.30}$$

The four vertices, W_A, W_B, W_C, W_D form a rectangular plane (Fig. 4.4).

The average distortion constraint is denoted by a straight line bisecting the rectangle. If W_1 is an optimal solution point, then it should lie on the intersection of the line between W_A,W_B and the constraint line. However, by the concavity of $H_{1/2}(\hat{\mathbf{U}}_N)$, at least one of the points at the intersection of the edges of the rectangle and the constraint line, would give a better solution. Obviously any such point is a point between two vertices, one within the constraint bound, the other *out* of the constraint, and which are at hamming distance 2.

The above argument is also applicable for a hamming distance 6 case. For any two vertices at hamming distance 6 , we should define six other vertices which are at hamming distance 4 to one and at hamming distance 2 to other. The resulting 8 vertices form a a rectangular prism, with the suggested optimal solution point lying on the intersection of the distortion plane and the diagonal between initial two vertices. By concavity at least one of the points at the intersection of the constraint plane and the edges of the rectangular prism would give a better solution. Thus ,by induction the optimal solution to the optimization problem lies between two vertices at hamming distance 2 to each other.

Claim 4.5 *If $\mathcal{C}$ is organized so that $p_1 \geq p_2 \geq \ldots \geq p_{2^N}$, then any optimal solution point $W_N \in \mathcal{W}_N$, should be represented by an upper triangular forward test channel transition matrix, i.e.: If $W_N = \{w_{ij}\}$ is an optimal solution then,*

$$w_{ij} = 0; \; if \; i > j. \tag{4.31}$$

The simple solution of SDE-1 design problem obviously agrees with Claim 4.5. Proof for $N \geq 1$ case can be given as follows:

Proof: Let W_N be a point in $\mathcal{R}$ and the corresponding SDE-N output be such that, $\hat{p}_1 \geq \ldots \geq \hat{p}_{2^N}$. In other words ,let W_N represent an upper triangular transition matrix, where the codebook of the source, $\mathcal{C}$, is arranged such that $p_1 \geq \ldots \geq p_{2^N}$. Let the average distortion with respect to W_N be equal to the average distortion constraint, $D_{avg}(W_N) = D_C$. Also, let W'_N be an optimal solution point in R which is represented by a transition matrix ,which is not upper triangular. As an optimal solution point W'_N must be on the constraint boundary, $D_{avg}(W'_N) = D_C$.

Without loosing generality, let the transition matrices of W'_N and W_N differ only in one column of their transition matrices, so that for $k > j \geq l$;

$$w_{lj} Dist(C_l, Cj) = w'_{kj} Dist(C_k, Cj) + w'_{lj} Dist(C_l, Cj) \tag{4.32}$$

$$w_{mn} = w'_{mn} \quad if \quad n \neq j \; and \; m \notin \{k, l\} \tag{4.33}$$

and the average distortion values are equal for both. Let the codewords C_j, C_k, C_l be arranged such that;

$$Dist(C_k, C_j) \geq Dist(C_l, C_j). \tag{4.34}$$

Since W_N is taken to be upper-triangular, w_{kj}=0, and since any optimal solution point, by Claim 4.4, is to be on a line connecting two vertices at hamming distance 2,

$$w_{mn} \in 0, 1 \quad if \quad n \neq j \; and \; m \notin \{k, l\} \tag{4.35}$$

Then using the Eqns. 4.17, 4.32-4.35;

$$\hat{p}_l = \sum_i p_i w_{li} \tag{4.36}$$

$$\hat{p}'_l = \hat{p}_l - \left(p_j w'_{kj} \frac{Dist(C_k, C_j)}{Dist(C_l, C_j)} \right) \quad (4.37)$$

$$\hat{p}_k = \sum_i p_i w_{ki} \quad (4.38)$$

$$\hat{p}'_k = \hat{p}_k + p_j w'_{kj} \quad (4.39)$$

and,

$$\hat{p}'_i = \hat{p}_i \quad for\ i \neq k, l \quad (4.40)$$

Note that, for $0 \leq p_i \leq 1$ and $\sum_i p_i = 1$,

$$\sqrt{\sum_i p_i} \leq \sum_i \sqrt{p_i} \leq \sqrt{N}. \quad (4.41)$$

Interpretation of Eqn. 4.41 is that, as the distribution p_i gets more uniform, value of the square root sum increases.

According to the definitions of W'_n and W_N above, and Eqns. 4.36-4.40 ;

$$\sum_i \sqrt{\hat{p}'_i} - \sum_i \sqrt{\hat{p}'_i} = \sqrt{\hat{p}'_l} + \sqrt{\hat{p}'_k} - \left(\sqrt{\hat{p}_k} + \sqrt{\hat{p}_l} \right) \quad (4.42)$$

and by 4.41

$$\sum_i \sqrt{\hat{p}'_i} \geq \sum_i \sqrt{\hat{p}_i} \ if\ \hat{p}'_k \leq \hat{p}_l. \quad (4.43)$$

Now assume that, $\hat{p}'_k > \hat{p}_l$,or equivalently by 4.39,

$$p_j w'_{kj} > \hat{p}_l - \hat{p}_k \quad (4.44)$$

Note that,

$$\hat{p}_l = p_l w_{ll} + p_j w_{lj} \quad (4.45)$$

$$\hat{p}_l = p_k w_{kk} \quad (4.46)$$

$$w_{lj} = w'_{lj} + w'_{kj} \frac{Dist(C_k, C_j)}{Dist(C_l, C_j)} \quad (4.47)$$

then 4.44 becomes;

$$p_k w_{kk} + p_j (w'_{kj} - w_{lj}) > p_l w_{ll} \quad (4.48)$$

Since $w'_{lj} \geq 0$ and $Dist(C_k, C_j) \geq Dist(C_l, C_j) \geq 0$,

$$w'_{kj} \leq w_{lj} \tag{4.49}$$

and since values of w_{kk} and w_{ll} are taken from a binary alphabet, 4.48 suggests

$$p_k > p_l \quad or \quad 0 > 0 \tag{4.50}$$

which are both contradictory.

Hence $\sum_i \sqrt{\hat{p}'_i} \geq \sum_i \sqrt{\hat{p}_i}$. By (3.18) $H_{1/2}(\hat{\mathbf{U}}'_N) > H_{1/2}(\hat{\mathbf{U}}_N)$ and W'_N can not be an optimal solution. We conclude the proof by adding that such a point W_N can be found for any point W'_N in $\mathcal{R}$ with an upper triangular matrix representation.

It will be shown in Chapter 5 that, the results for $N = 2$ case are in agreement with Claim 4.5.

Search for a solution at the constraint boundary is now reduced to a search among couples of vertices in region $\mathcal{R}$, which lie on the different sides of the constraint boundary and among the vertices which satisfy the average distortion constraint. The complexity of that search can be calculated as follows:

- Total number of vertices: $(2^N)^{(2^N)}$.
- Total number of hamming distance 2 neighbors of any vertex: $(2^N - 1)2^N$.
- By Claim 4.4, it is sufficient to check all *in-constraint* vertices for all of their *out-constraint* neighbors(or vice-versa). Total number of *in-constraint* or *out-constraint* vertices at worst case is half the total. In addition to Claim 4.5 suggests to check only the vertices with an upper triangular transition matrix representation. The total number of such vertices are given as:$(2^N)!$. Hence total number of vertices to be checked for its neighbors is: $\frac{(2^N)!}{2}$.
- For any vertex, total number of hamming distance 2 neighbors with upper triangular transition matrix representation is: $\sum_{i=1}^{2^N-1} i$.
- On the average half of the neighbors of an *in-constraint* vertex is expected to be *out-constraint*. Hence total number of neighbors to be checked for any vertex is: $\frac{\sum_{i=1}^{2^N-1} i}{2}$.

- The total number of couples to be checked is then :

$$\frac{(2^N)!}{2}\frac{\sum_{i=1}^{2^N-1} i}{2}. \tag{4.51}$$

The complexity of the search increases very rapidly with increasing memory length N. The number of vertices for different N is:

$$\begin{aligned} N=2: &\quad 24 &\quad Vertices \\ N=3: &\quad 40320 &\quad Vertices \\ N=4: &\quad 2.0923*10^{13} &\quad Vertices \end{aligned}$$

The search is done by a computer program, for N=2 and N=4. The rapid increase of complexity prevented the computations for higher N.

Zhang, Yang and Wei [11] analyses the bounds for performance of Rate Distortion Encoding with respect to block-length N. For a BMS-p source, and sufficiently large N, the optimum average distortion for a rate R block code is given by ,

$$D_N(R) = d(p,R) - \frac{\partial}{\partial R}d(p,R)\frac{\ln N}{2N} + o\left(\frac{\ln N}{N}\right) \tag{4.52}$$

Results of [11] can not be used to compare with the results of SDE-N, because of the $o\left(\frac{\ln N}{N}\right)$ term.

4.2.3 SDE-N as $N \to \infty$

The problem of SDE-N design for large N is extremely difficult to solve, and we have not been able to compute the optimum performance for large N. Still, we have to consider the case when $N \to \infty$.

It was shown in section 3.2.2 that an optimal solution point (there might be more than one) is between two vertices of hamming distance 2, and on the distortion constraint boundary. As $N \to \infty$, the number of vertices also goes to infinity, much faster. But what happens to the average distortion difference between two vertex? Let $W_A, W_B \in \mathcal{W}_N$ be two vertices at hamming distance 2 to each other and $w_{A_{kl}} = 1, w_{A_{k'l}} = 0, w_{B_{kl}} = 0, w_{B_{k'l}} = 1$ be the differentiating

elements of their transition matrices. The average distortion difference between W_A and W_B is then;

$$\begin{aligned} \triangle D_{avg} &= \mid p_l Dist(C_k, C_l) - p_l Dist(C_{k'}, C_l) \mid \\ &= \frac{p_l}{N} \mid Dist(C_k, C_l) - Dist(C_{k'}, C_l) \mid \end{aligned} \tag{4.53}$$

Assume the worst case to obtain the maximum difference, so that; $C_{k'} = C_l$ and $Dist(C_k, C_l) - N$ (or vice-versa). Assume also that $p_l \geq p_i \forall i$. Then for a BMS-p source with $Pr\{u = 1\} = p \leq 0.5$, C_l is the all zeros sequence with $p_l = (1-p)^N$ .Except for $p = 0$, in which case there would be no information to be compressed, $p_l \rightarrow \infty$ as $N \rightarrow \infty$. Hence ;

$$\lim_{N \rightarrow \infty} \triangle D_{avg} = 0. \tag{4.54}$$

which means that, asymptotically ,the optimal solution point(s) fall *on* the vertices.

The transition matrix of a vertex W_N, by definition, has all its elements w_{ij} chosen from a binary set, i.e.:$w_{ij} = 1, 0; i, j = 1, 2, \ldots, 2_N$. The test channel ,then, does not indicate a probabilistic process, but a deterministic many-to-one mapping from source data sequence to a reconstruction sequence.

Let $\mathcal{C}$ denote the source codebook with all possible codewords C_i of length N ,$C_i = (U_{i1} \ldots U_{iN})$ where U_{ij} are output letters of a BMS-p source. The size of C with a binary alphabet is 2^N.

Let $\hat{\mathcal{C}}$, the reconstruction codebook of SDE-N encoder be the set of all possible codewords $\hat{C}_i$ of length N ,$\hat{C}_i = (\hat{U}_{i1} \ldots \hat{U}_{iN})$ where $\hat{U}_{ij}$ are elements of an encoder output block, $\hat{\mathbf{U}}_N = (\hat{U}_1, \ldots, \hat{U}_N)$. Because of the many-to-one mapping done by the SDE-N:

$$|\hat{\mathcal{C}}| \leq |\mathcal{C}| \tag{4.55}$$

and by the definition of entropy [12],

$$2^{H(\hat{\mathbf{U}})} \leq |\hat{\mathcal{C}}| \tag{4.56}$$

By Asymptotic Equipartition Theory (AEP) [12], for any $\alpha \geq 0$,

$$\lim_{N \rightarrow \infty} H_{\frac{1}{1+\alpha}}(\hat{U}_N) = \log_2 |\hat{\mathcal{C}}| \tag{4.57}$$

Then as $N \rightarrow \infty$,

$$|\hat{\mathcal{C}}| \rightarrow 2^{H_{\frac{1}{1+\alpha}}(\hat{\mathbf{U}}_N)} \tag{4.58}$$

Let $\mathcal{C}'$ denote the reconstruction codebook of a RDE-N encoder. As $N \rightarrow \infty$, by Rate Distortion Theory [12]

$$|\mathcal{C}'| \rightarrow 2^{NR(D)} \tag{4.59}$$

and the codewords C'_i can be denoted by binary blocks of length $NR(D)$.

Let us denote the compression rates of SDE and RDE , in terms of source bits per channel bits, as λ_{SDE} and λ_{RDE} respectively.

$$\lambda_{SDE} = \frac{N}{H_{1/2}(\hat{\mathbf{U}}_N)} \tag{4.60}$$

$$\lambda_{RDE} = \frac{N}{NR(D)} \tag{4.61}$$

The many-to-one mapping of SDE-N, as $N \rightarrow \infty$, is essentially the same with many-to-one mapping done by the RDE. To obtain the same average distortion , codebooks of both encoder should form a similar distortion typical set with the source codebook. The only difference is that, the codewords $\hat{C}_i$ are represented by unnecessarily many bits. Hence as $N \rightarrow \infty$, $|\hat{\mathcal{C}}| \rightarrow |\mathcal{C}'|$. That implies,

$$\frac{H_{1/2}(\hat{\mathbf{U}}_N)}{N} \rightarrow R(D) \tag{4.62}$$

$$\lambda_{SDE} \rightarrow \lambda_{RDE} \tag{4.63}$$

And the rate of the SDE R_{SDE},

$$R_{SDE} = \frac{1}{\lambda_{SDE}} = \frac{H_{1/2}(\hat{\mathbf{U}})}{N} \tag{4.64}$$

then as $N \rightarrow \infty$, the rate of the SDE-N approaches the Rate Distortion Function, $R(D)$.

4.3 SDE-N Optimal Solution Search Program

The program written for to search for the optimal solution for a SDE-N encoder is designed according to the claims in section 3.2.2. C++ programming code

is used to obtain high speed computation. It has three main parts.

1. **Preliminary:** In this part of the program necessary parameters such as Source Output Block Probability vectors $\mathbf{P}_N$ for blocks of N, is computed. The program assumes a BMS-p with $P(U = 1) = p$, and requests the value of p as an input. The program then generates the blocks of length N, with N predefined and calculates the hamming distances $Dist(C_i, C_j)$ for $i, j = 1, \ldots, 2^N$ and stores as a matrix. By the help of this matrix the block probabilities $p_i = P_N \mid_{\mathbf{U}_N = C_i}$ is calculated and stored in a vector $\mathbf{P}_N$. Using the block probabilities the order of codewords C_i in the code book is organized so that, $p_1 \geq p_2 \geq \ldots \geq p_{2^N}$. Both the matrix of hamming distances and $\mathbf{P}_N$ are reorganized accordingly.

   The program stores the important parameters for all the vertices it checks in memory, for speedy search. These parameters are namely: The Rényi Entropy of encoder output $\hat{\mathbf{U}}$ (per block of N bits) corresponding to that vertex, the average distortion $E[d(\mathbf{U}, \hat{\mathbf{U}})]$,and a flag which shows if that vertex was checked for all its neighbors before.

   The coordinates of that vertex, which in fact are elements of the transition matrix W_N gives the index of the vertex for the array in which all vertices are stored. In order to calculate the index from the elements ,certain basis values are calculated and stored for every coordinate variable w_{ij}.

2. **Search:** Search for the optimal solution, with any given distortion constraint is done recursively. Every *in-constraint* vertex with an upper triangular transition matrix is traced and every *out-constraint* neighbor with upper triangular matrix representation is checked. Search begins in the *zero average distortion vertex*, for which

$$w_{ij} = \begin{cases} 1, & if \quad i = j \\ 0, & if \quad i \neq j \end{cases} \tag{4.65}$$

   After the average distortion constraint is defined, a certain subroutine is called to check all hamming distance two neighbors. Inputs to the subroutine is only the coordinates of the vertex.

   Define the *potential* of any point in region $\mathcal{R}$ as the corresponding encoder output's Rényi Entropy per block of N bits,$H_{1/2}(\hat{\mathbf{U}}_N)$. The program searches one of the the lowest potential points within the average

distortion constraint. The *temporary solution* is the best solution obtained at any instant. Any *out-constraint* neighbor is checked to find an optimal solution. Since $H_{1/2}(\hat{\mathbf{U}}_N)$ is concave, *out-constraint* neighbors with potentials higher than that of the temporary solution is useless and disregarded. If a point gives a solution which is better than the temporary solution, it is stored as the new temporary solution with its Transition matrix W_N, average distortion $D_{avg}(W_N)$ (which should be equal to distortion constraint, D_C, and the Rényi Entropy value. Initially, the temporary solution is the *zero average distortion vertex* itself.

When the subroutine is faced with an *in-constraint* neighbor, it calls itself to check the neighbor for its own neighbors. This way all *in-constraint* vertices are spanned. Once an *in-constraint* vertex is checked ,its span-flag is set.

At any level of recursion, if the program is faced with a new vertex, it calculates the potential and $D_{avg}(W_N)$ of the vertex, and stores them with the index calculated from its coordinates. The values are then used directly, when that vertex is to checked for a different neighbor.

The search is done for different values of average distortion constraint. Temporary solutions for different constraints are stored in an array.

3. **Output:** Solutions for all values of distortion constraints are already stored in an array neatly. The stored data is organized and printed to a file. Output file is organized so that, distortion constraint levels and corresponding minimum $H_{1/2}(\hat{\mathbf{U}}_N)/N$ values are printed in vector form without giving a corresponding Forward Test Channel Representations, W_N. After this summary, the solution for all distortion constraint levels are listed in ascending order w.r.t. distortion. For each, the minimum achievable $H_{1/2}(\hat{\mathbf{U}}_N)/N$, and the corresponding Forward Test Channel Representation, W_N of SDE-N is given. Examples of results from output files, for N=2 and N=3 cases can be found at Appendix A.

Chapter 5

RESULTS AND SIMULATIONS FOR SDE

5.1 Results of SDE-N Design

In this section, the minimum obtained $H_{1/2}(\hat{\mathbf{U}}_N)/N$ values for an SDE-N output, with respect to an average distortion constraint will be given. All the results are obtained by the search program explained in section 4.3.

Any SDE-N encoder is completely defined by its Forward Test Channel Transition Probability Matrix, W_N. Some of the transition matrices of the optimal SDE-N encoders corresponding to the given minimum values of $H_{1/2}(\hat{\mathbf{U}}_N)/N$ for $N = 2$ and $N = 3$ are presented in Appendix A.

5.1.1 Results for SDE-2

Results of optimal SDE-2 design is given in Fig. 5.1 ,as an encoder rate R_{SDE} 4.64 versus average distortion D plot. The source is assumed to be a Binary Memoryless source with $P(U = 1) = p$, (BMS-p) with $p = 0.35$.

Since the rate of encoder R_{SDE} converges to Rate Distortion Function $R(D)$, R_{SDE} is compared with $R(D)$ and $R_{SDE}|_{N=1}$ (rate for optimal SDE-1) which are plotted in dashed and dash-dotted line.

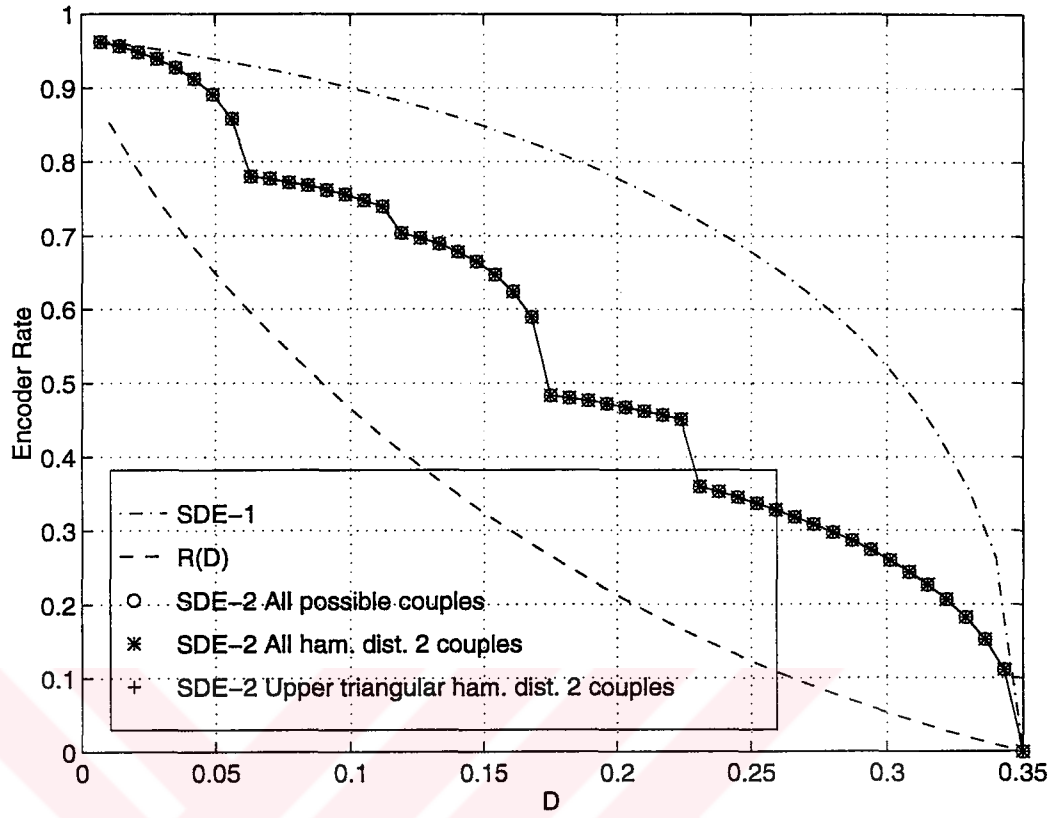

Figure 5.1: R_{SDE-2} v.s. D plots. $R(D)$ v.s. D and R_{SDE-1} v.s. D curves are plotted for comparing.

In order to give further proof for Claims 4.4, 4.5, for $N = 2$:

1. An optimal solution is searched for, among every possible couple of vertices in region $\mathcal{R}$ (Ignoring claims 4.4, 4.5).

2. An optimal solution is searched for, among every hamming distance 2 couples in region $\mathcal{R}$ (Ignoring claim 4.5).

3. An optimal solution is searched for among hamming distance 2 couple of vertices with upper triangular transition matrices.

Although the optimal solution points W_N ,for three cases, are not all equal(Appendix A), the corresponding SDE-2 encoder performances are exactly the same. This is because optimal solution is not unique. As seen in Fig. 5.1, all tree results give further proof for claims 4.4 and 4.5.

There are certain indifferentiable points in the resulting R_{SDE} *vs* D curve.

Corresponding transition matrices, W_n, indicate that such points are due *Vertex crossing.* In other words those indifferentiable points corresponds to a solution point *on* one of the vertices of region $\mathcal{R}$.

5.1.2 Results for SDE-3

Results of optimal SDE-3 design is given in Fig. 5.2, as an encoder rate (R_{SDE}) (3.63) versus average distortion (D) plot. The source is assumed to be a Binary Memoryless source with $P(U = 1) = p$, (BMS-p) with $p = 0.35$.

Since the rate of encoder R_{SDE} converges to Rate Distortion Function $R(D)$, R_{SDE} is compared with $R(D)$,$R_{SDE}|_{N=1}$ (rate for optimal SDE-1) and R_{SDE-2} .

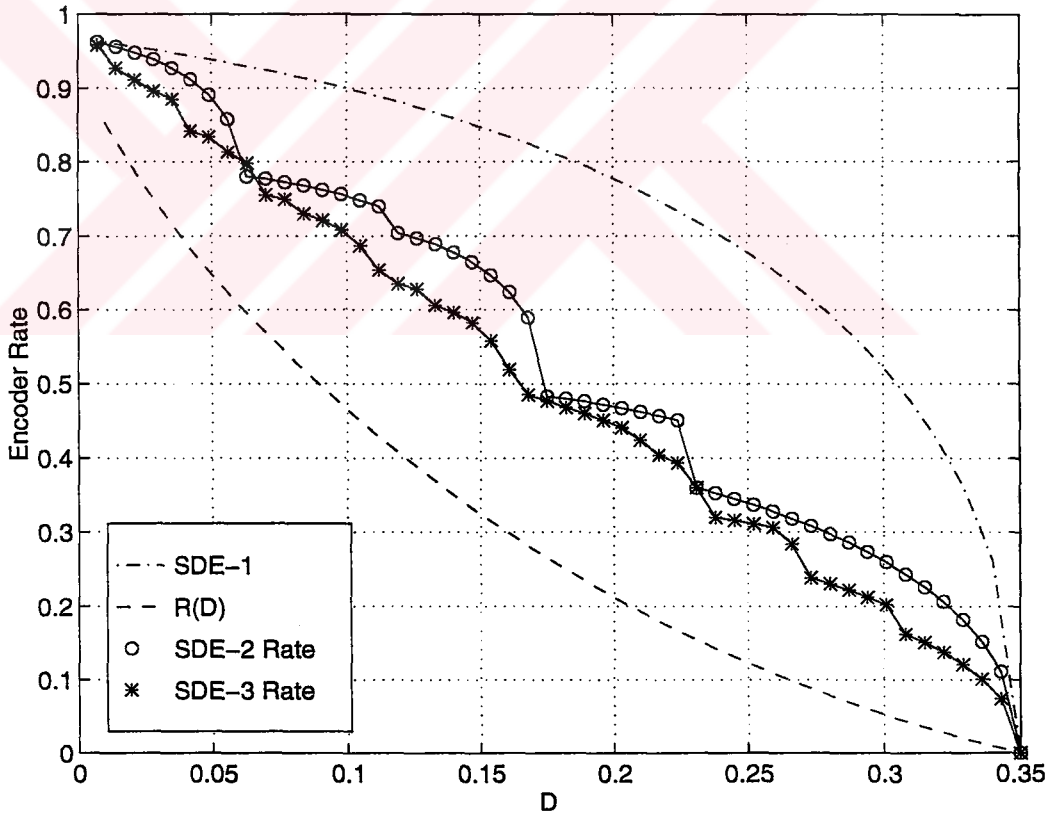

Figure 5.2: R_{SDE-3} v.s. D plot. $R(D)$ v.s. D,R_{SDE-1} v.s. D and R_{SDE-1} v.s. D curves are also plotted for comparing.

5.2 Simulation Results

5.2.1 Simulated System

The block diagram of the simulated communication system is given in Fig. 5.3. Source is assumed to be a Binary Memoryless Source. Transmitter consists of a stochastic distortion encoder, and a convolutional encoder. The transmission channel is a Binary Symmetric Channel(BSC). The receiver consists of a sequential decoder. The constraint length of the convolutional encoder is taken to be 100.

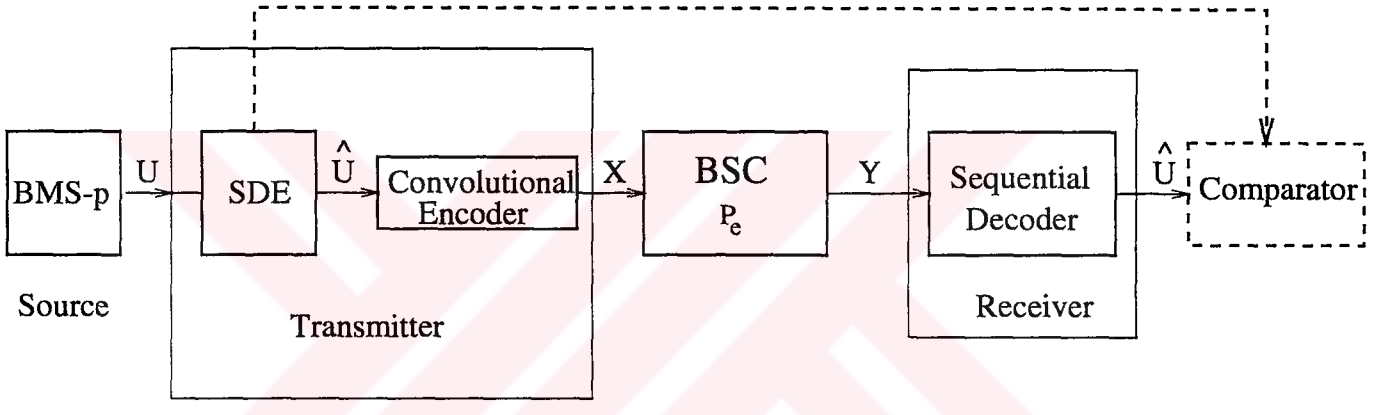

Figure 5.3: Block diagram of the simulated system.

Simulations are made by a C++ program. The program consists of separate modules acting as source, stochastic encoder, convolutional encoder, binary symmetric channel, sequential decoder, and comparator.

The sequential decoder that is used for simulations is based on the simple stack algorithm [4]. Certain modifications for *metric* computations are made for its use in a JSC Coding system ,as explained in [2] and [1].

Simulations are run for SDE-1, SDE-2, SDE-3, using the values from Appendix A.

5.2.2 Simulations for SDE-1

Table 5.1 gives the results of simulations for SDE-1 encoder, with a noiseless channel. Parameters of interest are listed as: average distortion caused by

the stochastic encoder D_{avg}, encoder rate as defined in section 4.3 R_{SDE-1}, number of bits in error at receiver output (# errors), decoder search complexity (# search), source sequence length (seq. length), expected average distortion (D_{exp}),and minimum encoder rate corresponding to that average distortion($minR_{SDE-1}$) . Number of bits in error are counted by the comparator. The comparator compares the SDE output sequence with received sequence, hence gives the additional distortion, independent of the distortion caused by the SDE-1. Search complexity is denoted by the number of branches that the sequential encoder traces to find the correct path in the course of decoding divided by the length of received sequence. Expected distortion is the average distortion value corresponding to the encoder rate as found in section 4.2.1.

Table 5.1 lists the simulations results for memory length 1.

	R_{SDE-1}	$min\ R_{SDE-1}$	D_{avg}	D_{exp}	*# errors*	*seq. length*	*# search*
	0.500	0.500	0.289	0.305	0	512	2.028
	0.500	0.500	0.306	0.305	0	1016	2.158
	0.500	0.500	0.302	0.305	0	1696	2.693
	0.500	0.500	0.306	0.305	0	2600	2.970
Avg.	0.500	0.500	0.301	0.305	0	–	2.462
	0.750	0.750	0.197	0.216	0	512	4.713
	0.750	0.750	0.208	0.216	0	1016	5.647
	0.750	0.750	0.212	0.216	0	1696	6.996
	0.750	0.750	0.216	0.216	0	2600	7.000
Avg.	0.750	0.750	0.208	0.216	0	–	6.089

Table 5.1: Table of results for SDE-1 Simulations.

The rows labeled as 'Avg.' presents an average of the above rows for columns D_{avg}, number of errors, and decoder search complexity. Since the encoding process is stochastic, an average of the performance would best testify the results.

Note that the decoder search complexity is absolutely irrelevant of the decoder complexity, and only listed to show that the sequential decoder works properly. An exponential increase in decoder search complexity (# search), with increasing source sequence length (seq. length) would show that the sequential encoder runs above its cut-off rate. The simulation results show that the sequential decoder works below its cut-off rate. Figure 5.4 compares the expected average distortion and the average distortion found in the simulations

for SDE-1.

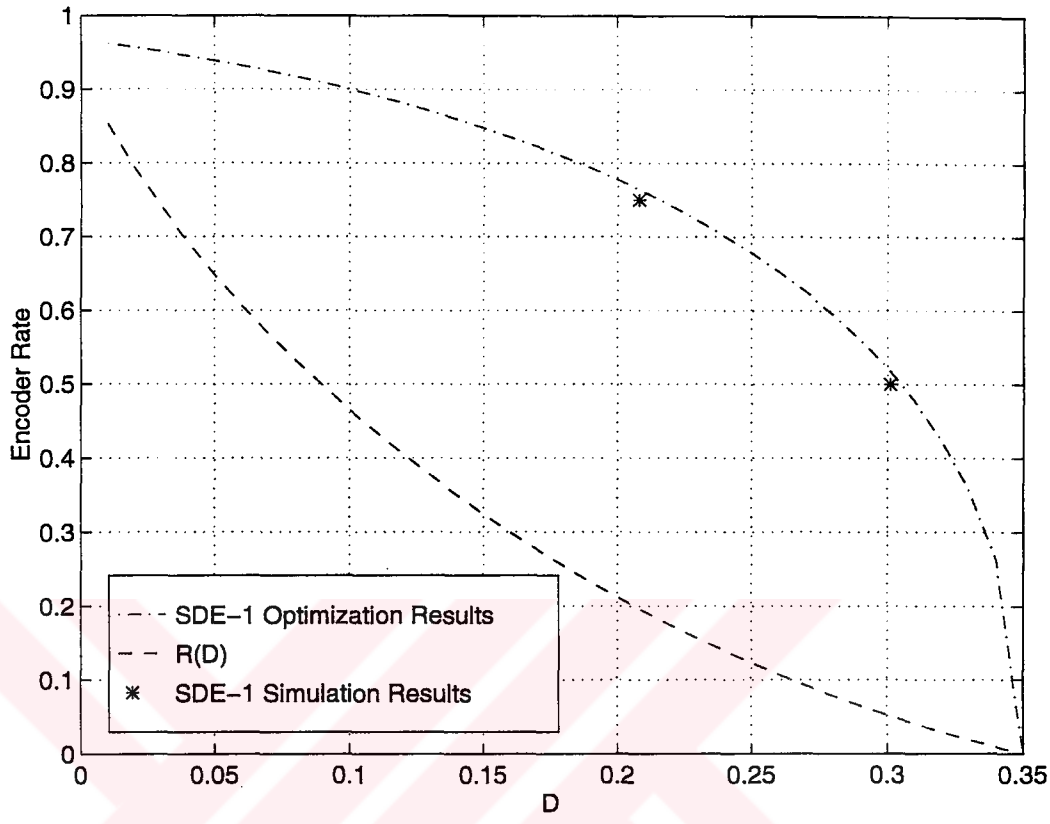

Figure 5.4: Simulation results for SDE-1 compared with computations for optimal encoder.

5.2.3 Simulations for SDE-2

Table 5.2 gives the results of simulations for SDE-2 encoder, with a noiseless channel. Parameters are of interest are listed similar to SDE-1 case. Expected distortion values are taken from our results in section 5.1 . Parameters of SDE-2 encoders used in the simulations are taken from the lists in Appendix A.

Table 5.2 lists the simulations results for memory length 2.

The rows labeled as 'Avg.' presents an average of the above rows for columns D_{avg}, number of errors, and decoder search complexity. Since the encoding process is stochastic, an average of the performance would best testify the results.

	R_{SDE-2}	$min\ R_{SDE-2}$	D_{avg}	D_{exp}	# errors	seq. length	# search
	0.250	0.243	0.311	0.308	0	512	7.362
	0.250	0.243	0.299	0.308	0	1016	5.587
	0.250	0.243	0.307	0.308	0	1696	7.326
	0.250	0.243	0.308	0.308	0	2600	5.817
Avg.	0.250	0.243	0.306	0.308	0	–	6.523
	0.333	0.328	0.270	0.259	0	512	17.900
	0.333	0.328	0.258	0.259	0	1016	21.254
	0.333	0.328	0.262	0.259	0	1696	17.396
	0.333	0.328	0.261	0.259	0	2600	20.106
Avg.	0.333	0.328	0.263	0.259	0	–	19.164
	0.500	0.483	0.171	0.175	0	512	2.000
	0.500	0.483	0.171	0.175	0	1016	2.000
	0.500	0.483	0.167	0.175	0	1696	2.000
	0.500	0.483	0.173	0.175	0	2600	2.000
Avg.	0.500	0.483	0.171	0.175	0	–	2.000
	0.667	0.665	0.147	0.147	0	512	13.850
	0.667	0.665	0.157	0.147	0	1016	19.326
	0.667	0.665	0.150	0.147	0	1696	16.181
	0.667	0.665	0.148	0.147	0	2600	17.403
Avg.	0.667	0.665	0.151	0.147	0	2600	16.690
	0.750	0.748	0.112	0.105	0	512	7.956
	0.750	0.748	0.108	0.105	0	1016	11.295
	0.750	0.748	0.103	0.105	0	1696	12.664
	0.750	0.748	0.105	0.105	0	2600	13.294
Avg.	0.750	0.748	0.107	0.105	0	2600	11.302

Table 5.2: Table of results for SDE-2 Simulations

Figure 5.5 compares the $R_{SDE-2} - D$ curve found in section 5.1.1 and the average distortion values found in the simulations for SDE-2.

5.2.4 Simulations for SDE-3

Table 5.3 gives the results of simulations for SDE-3 encoder, wit a noiseless channel. Parameters are of interest are listed similar to SDE-1, and SDE-2 cases. Expected distortion values are taken from our results in section 5.1 . Parameters of SDE-3 encoders used in the simulations are taken from the lists in Appendix A.

Table 5.3 lists simulation results for memory length 3.

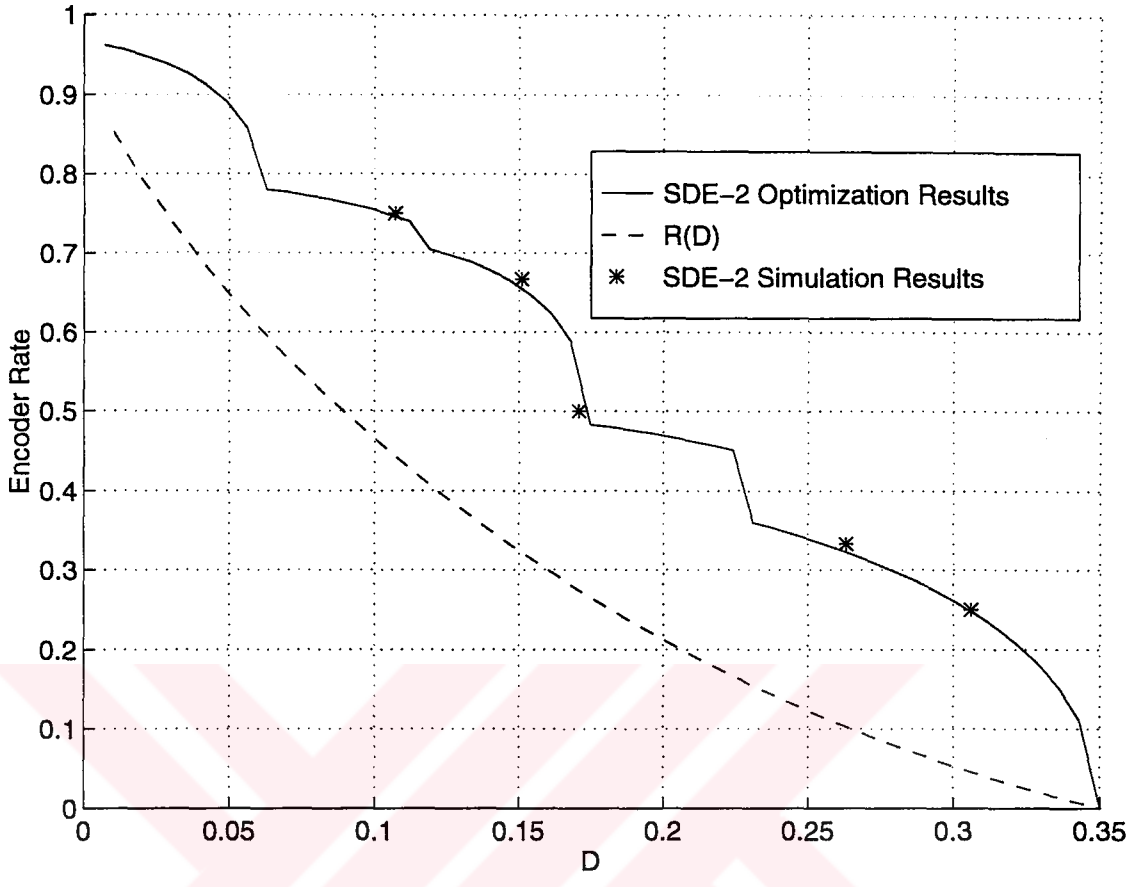

Figure 5.5: Simulation results for SDE-2 compared with computations for optimal encoder.

The rows labeled as 'Avg.' presents an average of the above rows for columns D_{avg}, number of errors, and decoder search complexity. Since the encoding process is stochastic, an average of the performance would best testify the results.

Figure 5.6 compares the $R_{SDE-3} - D$ curve found in section 5.1.2 and the average distortion values found in the simulations for SDE-3 with a noiseless channel.

Table 5.4 gives the results of simulations for SDE-3 encoder, wit a noisy binary symmetric channel. Bit error probability of binary symmetric channel is $P_e = 0.0449$. Corresponding cut-off rate for the sequential decoder is $E_0(1) = 0.5$. Parameters are of interest are listed similar to noiseless SDE-1, SDE-2 and SDE-3 cases, except that instead of the actual and minimum SDE rates , the actual and maximum convolutional encoder rates, R_c and $R_{c_{max}}$. Expected distortion values are taken from our results in section 5.1. Maximum rates

	R_{SDE-3}	*min* R_{SDE-3}	D_{avg}	D_{exp}	*# errors*	*seq. length*	*# search*
	0.167	0.162	0.309	0.308	0	512	12.150
	0.167	0.162	0.307	0.308	0	1016	14.249
	0.167	0.162	0.309	0.308	0	1696	23.094
	0.167	0.162	0.308	0.308	0	2600	22.562
Avg.	0.167	0.162	0.308	0.308	0	–	18.025
	0.333	0.320	0.242	0.238	0	512	2.667
	0.333	0.320	0.237	0.238	0	1016	2.667
	0.333	0.320	0.240	0.238	0	1696	2.667
	0.333	0.320	0.240	0.238	0	2600	2.667
Avg.	0.333	0.320	0.240	0.238	0	–	2.667
	0.500	0.485	0.166	0.168	0	512	17.300
	0.500	0.485	0.168	0.168	0	1016	17.650
	0.500	0.485	0.167	0.168	0	1696	19.441
	0.500	0.485	0.168	0.168	0	2600	19.249
Avg.	0.500	0.485	0.167	0.168	0	–	18.410
	0.667	0.654	0.119	0.112	0	512	2.667
	0.667	0.654	0.115	0.112	0	1016	2.667
	0.667	0.654	0.110	0.112	0	1696	2.667
	0.667	0.654	0.112	0.112	0	2600	2.667
Avg.	0.667	0.654	0.114	0.112	0	–	2.667
	0.833	0.813	0.061	0.056	0	512	13.515
	0.833	0.813	0.059	0.056	0	1016	27.150
	0.833	0.813	0.058	0.056	0	1696	25.048
	0.833	0.813	0.056	0.056	0	2600	27.515
Avg.	0.833	0.813	0.059	0.056	0	–	23.307

Table 5.3: Table of results for SDE-3 Simulations

for convolutional encoder is calculated by Eqn. 2.23 . Parameters of SDE-3 encoders used in the simulations are taken from the lists in Appendix A.

The rows labeled as 'Avg.' presents an average of the above rows for columns D_{avg}, number of errors, and decoder search complexity. Since the encoding process is stochastic, an average of the performance would best testify the results.

Figure 5.7 compares the $R_{c_{max}} - D$ curve, which is obtained from the $R_{SDE-3} - D$ curve using Eqn. 2.23 and the average distortion values found in the simulations for SDE-3 with a noisy channel.

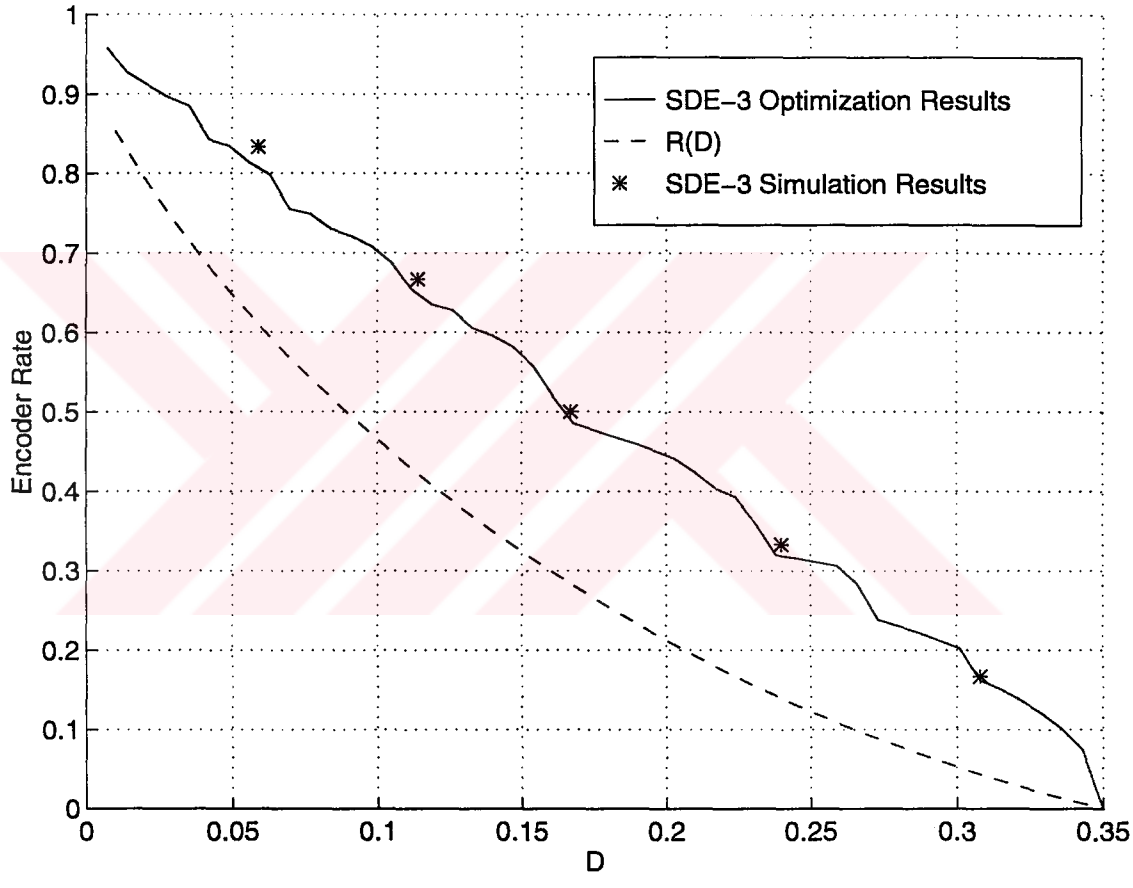

Figure 5.6: Simulation results for SDE-3 compared with computations for optimal encoder.

	R_c	$R_{c_{max}}$	D_{avg}	D_{exp}	# *errors*	*seq. length*	# *search*
	3.000	3.086	0.307	0.308	0	512	3.415
	3.000	3.086	0.307	0.308	0	1016	4.227
	3.000	3.086	0.307	0.308	0	1696	3.429
	3.000	3.086	0.306	0.308	0	2600	4.134
Avg.	3.000	3.086	0.307	0.308	0	2600	3.801
	1.500	1.562	0.233	0.238	0	512	3.253
	1.500	1.562	0.238	0.238	0	1016	4.192
	1.500	1.562	0.240	0.238	0	1696	3.991
	1.500	1.562	0.238	0.238	0	2600	4.500
Avg.	1.500	1.562	0.237	0.238	0	2600	3.984
	1.000	1.031	0.168	0.168	0	512	3.749
	1.000	1.031	0.168	0.168	0	1016	5.184
	1.000	1.031	0.169	0.168	0	1696	4.554
	1.000	1.031	0.166	0.168	0	2600	4.933
Avg.	1.000	1.031	0.168	0.168	0	2600	4.605
	0.750	0.796	0.127	0.126	0	512	6.654
	0.750	0.796	0.123	0.126	0	1016	7.042
	0.750	0.796	0.127	0.126	0	1696	6.572
	0.750	0.796	0.126	0.126	0	2600	6.735
Avg.	0.750	0.796	0.126	0.126	0	2600	6.751
	0.600	0.615	0.053	0.056	0	512	4.317
	0.600	0.615	0.053	0.056	0	1016	4.502
	0.600	0.615	0.056	0.056	0	1696	4.928
	0.600	0.615	0.056	0.056	0	2600	5.127
Avg.	0.600	0.615	0.055	0.056	0	2600	4.719

Table 5.4: Table of results for SDE-3 Simulations, noisy channel

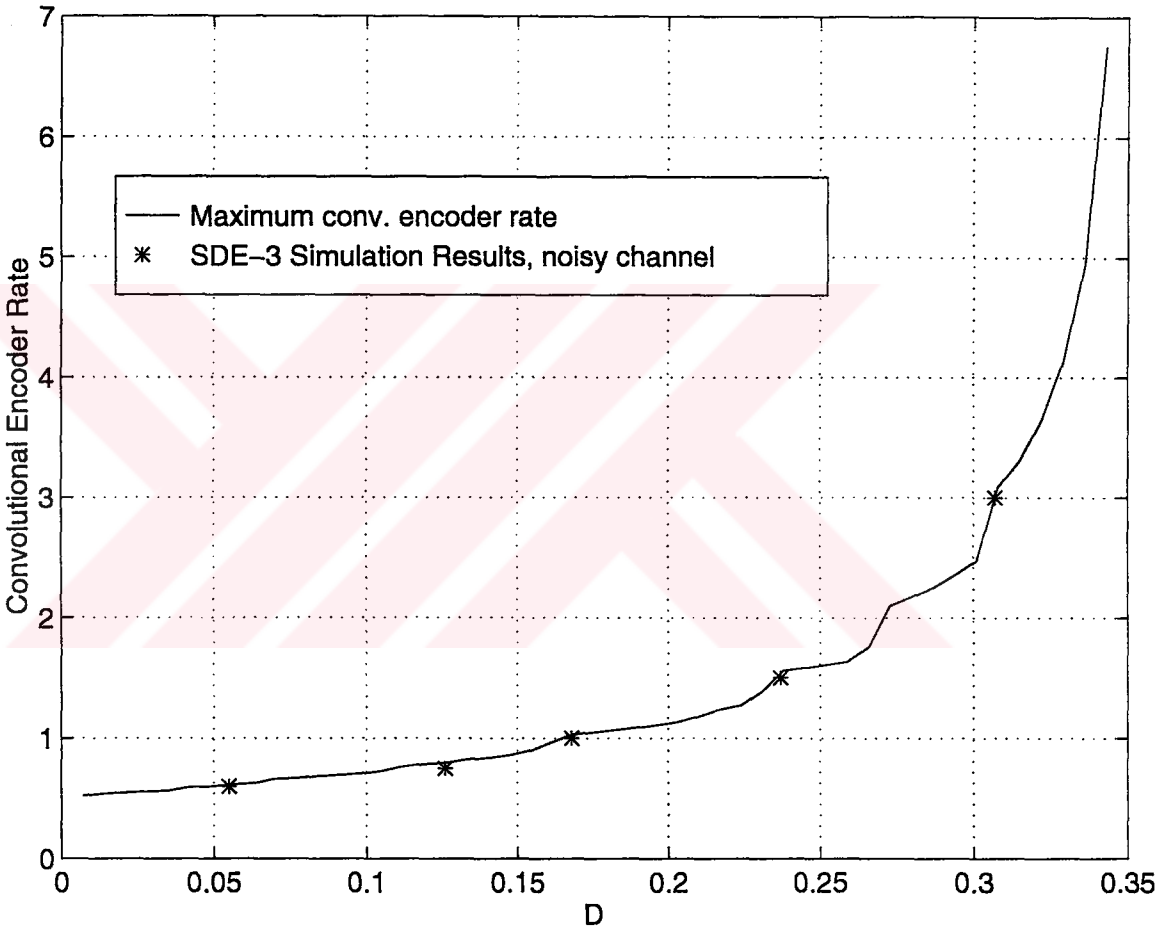

Figure 5.7: Simulation results for SDE-3 with noisy channel, compared with computations for optimal encoder.

Chapter 6

CONCLUSIONS

We considered a joint source-channel coding system with a sequential decoder, and worked on lossless transmission of text written in English. In order to achieve transmission rates above the cut-off rate, we have made use of the natural redundancy of written English, utilizing unigram and digram statistics for the modification of the metric of the sequential decoder as suggested by Hellman [1] and Koshelev [2]. In order to make better use of the statistics of English language we used a dictionary aided sequential decoding method, in which the guesses of the sequential decoder for the transmitted path along the code tree were filtered using a dictionary consisting all the words in the transmitted text.

Simulations were run for unigram, digram and dictionary aided decoding methods, with both noisy and noiseless channels. Results of simulations for unigram and digram methods support the achievability of the upper bound on the transmission rate derived by Arıkan and Merhav [3], and calculated using estimates on the Rényi entropy of English. The estimates on the Rényi entropy of English were found using Shannon's work on the estimation of the entropy of English [7] as reference. Dictionary aided decoding method performs better than digram and unigram decoding methods. Simulation results show that a maximum transmission rate of approximately $2R_f$ can be achieved. Using the upper bounds for transmission rate in JSC coding [3] and the maximum achieved rate with dictionary aided decoding, we can obtain an estimate for an upper bound on Rényi entropy of English. Based on the simulation results such an estimate equals to 2.5 in bits per letter.

We have also worked on an alternative low-complexity quantizer to be used with JSC coding method, for accuracy-compression trade-off in simple transmitter communication systems. We proposed a Stochastic Distortion Encoder (SDE) that makes a probabilistic many-to-many mapping between source codebook and the encoder output codebook, which is a sub-book of the source codebook. The length of the source sequence is not changed, but the redundancy of the sequence is increased, and the convolutional encoder of the JSC coding system is allowed to operate at higher rates. Mapping is modeled with the forward test channel which has a channel alphabet consisting of all the codewords of source codebook. The forward test channel transition matrix is sufficient to express the SDE operation. Probabilistic mapping allows us to obtain any average distortion desired. The main advantage of SDE over conventional rate-distortion encoding is the flexibility of the encoder to be used with any desired average distortion, even with very low memory lengths. We conjecture that the performance of a SDE is as good as an equivalent rate distortion encoder with same block length and average distortion.

SDE design is formulated based on the optimality criterion given by Eqn. 4.3. We showed that the Rényi entropy of the encoder output is concave over the convex set $\mathcal{W}_N$ of all forward test channel transition matrices W_N that represent all possible SDE designs with memory length N. Further inspection shows that, when source codewords are listed in descending order of their observation probability, optimal solution lies on the hyperplane formed by the average distortion constraint and between two vertices of set $\mathcal{W}_N$ which are at hamming distance 2 and have an upper-triangular transition matrix, if such vertices exists, and if not, solution point is one of the vertices satisfying the average distortion constraint. Vertices are defined as those elements of $\mathcal{W}_N$ which has all of its transition matrix elements chosen from a binary set. An SDE with a transition matrix representation that indicates a vertex, is expected to work exactly like a conventional rate-distortion encoder as probabilistic mapping is reduced to a deterministic mapping with probabilities all chosen from a binary alphabet {1,0}.

Formulations for SDE designs and SDE designs for memory lengths 1,2 and 3 are presented, and design parameters are compared with simulation results. The simulation results for both noiseless and noisy channels are in agreement with the design expectations. However, the formulations for optimal SDE

design shows that, the number of required computations for a design increases hyper-exponentially with increasing memory-length. Results for SDE-1,2 and 3 shows that optimal solution doesn't have to be unique.

We have also shown that, as memory length approaches infinity, the distance between vertices of set $\mathcal{W}_N$ diminishes, and optimal solutions approach to the vertices. As a result, the rate of SDE approaches to the lower bound of the rate of conventional rate distortion encoders,given by the rate distortion function.

The bounds for the rate of convolutional encoder of JSC coding system given by Arıkan and Merhav [3] formed a basis for our formulations of optimal SDE design. We present some simulation results that support the achievability of the optimal SDE performance computations for memory lengths 1,2 and 3. Results also suggests that, the average distortion introduced by SDE is a random variable. In other words average distortion is not stable. Yet, as source sequence length increases, variance of the average distortion decreases and becomes negligible for sufficiently long source sequences.

We apply the modification suggested by Koshelev [2] and Hellman [1] on the metric of the sequential decoder, for the JSC coding system. In the simulations with an error free transmission channel, we have seen that, the computational burden of the decoder is increased due to joint source-channel decoding, as it searches extensively over the *atypical regions* of the source sequence, as had been stated by Hellman [1].

When the transmission channel is noisy, the computational burden on a sequential decoder operating above the cut-off rate increases exponentially with increasing source sequence length due to transmission errors [3], [6]. Simulation results show that, for the given communication system model, with the calculated SDE specifications and corresponding maximum convolutional encoder rates, sequential decoders operate below their cut-off rate, and the average number of computations for correct decoding does not increase exponentially with increasing source sequence length. This implies the achievability of the compression rates calculated for stochastic distortion encoders.

It is a subject of further research, to obtain at least sub-optimal SDE designs for memory lengths greater than 3.

Appendix A

Optimization Results

A.1 Results for SDE-2,case 1

The forward test channel transition matrices for an optimal solution of SDE-2 design problem for various values of average distortion D_{avg} are listed. The *Feasible Vertex* denotes a certain transition matrix W_N which satisfies the average distortion constraint. The *Non-Feasible Vertex* is a neighbor of the feasible vertex which is not necessarily at hamming distance 2, and does not satisfy the average distortion constraint. *solution point* denotes the forward test channel transition matrix for an optimal SDE-2 with the given constraint. Solution point is between feasible and non-feasible vertices. Claim 4.5 was ignored during computations and matrices were not necessarily chosen from a set of upper triangular matrices.

For all the results below, the source is assumed to be a BMS-p with $p = P(U = 1) = 0.35$. Floating point numbers are rounded to three digits for values of $H_{1/2}(\hat{\mathbf{U}}_N)/N$ and D_{avg}, rounded to two digits for matrix elements of the solution point.

The codebook $\mathcal{C}$ is arranged as,

$$\begin{aligned} C_1 &= (00) \\ C_2 &= (01) \\ C_3 &= (10) \end{aligned}$$

$$C_4 \;=\; (11) \tag{A.1}$$

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.756;\ D_{avg} = 0.098$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 0 1 0	1.00 0.00 0.32 0.00
0 1 0 1	0 1 0 1	0.00 1.00 0.00 1.00
0 0 1 0	0 0 0 0	0.00 0.00 0.68 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.748;\ D_{avg} = 0.105$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 1 0 0	1.00 0.38 0.00 0.00
0 1 0 0	0 0 0 0	0.00 0.62 0.00 0.00
0 0 1 1	0 0 1 1	0.00 0.00 1.00 1.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.740;\ D_{avg} = 0.112$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 0 1 0	1.00 0.00 0.45 0.00
0 1 0 1	0 1 0 1	0.00 1.00 0.00 1.00
0 0 1 0	0 0 0 0	0.00 0.00 0.55 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.704;\ D_{avg} = 0.119$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 1 0	1.00 1.00 0.05 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 0	0 0 0 0	0.00 0.00 0.95 0.00
0 0 0 1	0 0 0 1	0.00 0.00 0.00 1.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.665;\ D_{avg} = 0.147$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 0 0	1.00 1.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 0	0 0 1 1	0.00 0.00 1.00 0.54
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.46

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.647$; $D_{avg} = 0.154$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 0 0	1.00 1.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 0	0 0 1 1	0.00 0.00 1.00 0.66
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.34

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.483$; $D_{avg} = 0.175$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 0 0	1.00 1.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 1	0 0 1 1	0.00 0.00 1.00 1.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.480$; $D_{avg} = 0.182$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 1 1	1.00 1.00 0.04 0.04
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 1	0 0 0 0	0.00 0.00 0.96 0.96
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.328$; $D_{avg} = 0.259$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 1 0	1 1 1 1	1.00 1.00 1.00 0.26
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.74

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.243$; $D_{avg} = 0.308$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 1 0	1 1 1 1	1.00 1.00 1.00 0.66
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.34

A.2 Results for SDE-2,case 2

The forward test channel transition matrices for an optimal solution of SDE-2 design problem for various values of average distortion D_{avg} are listed. The *Feasible Vertex* denotes a certain transition matrix W_N which satisfies the average distortion constraint. The *Non-Feasible Vertex* is a hamming distance 2 neighbor of the feasible vertex and does not satisfy the average distortion constraint. *solution point* denotes the forward test channel transition matrix for an optimal SDE-2 with the given constraint. Solution point is between feasible and non-feasible vertices. Claim 4.5 was ignored during computations and matrices were not necessarily chosen from a set of upper triangular matrices. The results, however, are all upper triangular matrices,

For all the results below, the source is assumed to be a BMS-p with $p = P(U = 1) = 0.35$. Floating point numbers are rounded to three digits for values of $H_{1/2}(\hat{\mathbf{U}}_N)/N$ and D_{avg}, rounded to two digits for matrix elements of the solution point.

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.756$; $D_{avg} = 0.098$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 0 1 0	1.00 0.00 0.32 0.00
0 1 0 1	0 1 0 1	0.00 1.00 0.00 1.00
0 0 1 0	0 0 0 0	0.00 0.00 0.68 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.748$; $D_{avg} = 0.105$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 1 0 0	1.00 0.38 0.00 0.00
0 1 0 0	0 0 0 0	0.00 0.62 0.00 0.00
0 0 1 1	0 0 1 1	0.00 0.00 1.00 1.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.740$; $D_{avg} = 0.112$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 0 1 0	1.00 0.00 0.45 0.00
0 1 0 1	0 1 0 1	0.00 1.00 0.00 1.00
0 0 1 0	0 0 0 0	0.00 0.00 0.55 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.665$; $D_{avg} = 0.147$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 0 0	1.00 1.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 0	0 0 1 1	0.00 0.00 1.00 0.54
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.46

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.647$; $D_{avg} = 0.154$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 0 0	1.00 1.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 0	0 0 1 1	0.00 0.00 1.00 0.66
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.34

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.483$; $D_{avg} = 0.175$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 1 0	1 0 1 0	1.00 0.00 1.00 0.00
0 1 0 1	0 1 0 1	0.00 1.00 0.00 1.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.480$; $D_{avg} = 0.182$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 1 0	1 1 1 0	1.00 0.06 1.00 0.00
0 1 0 1	0 0 0 1	0.00 0.94 0.00 1.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.328$; $D_{avg} = 0.259$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 1 0	1 1 1 1	1.00 1.00 1.00 0.26
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.74

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.243$; $D_{avg} = 0.308$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 1 0	1 1 1 1	1.00 1.00 1.00 0.66
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.34

A.3 Results for SDE-2,case 3

The forward test channel transition matrices for an optimal solution of SDE-2 design problem for various values of average distortion D_{avg} are listed. The *Feasible Vertex* denotes a certain transition matrix W_N which satisfies the average distortion constraint. The *Non-Feasible Vertex* is a hamming distance 2 neighbor of the feasible vertex and does not satisfy the average distortion constraint. *solution point* denotes the forward test channel transition matrix for an optimal SDE-2 with the given constraint. Solution point is between feasible and non-feasible vertices. All three were presumed to be triangular matrices during the computations, due to Claim 4.5.

For all the results below, the source is assumed to be a BMS-p with $p = P(U = 1) = 0.35$. Floating point numbers are rounded to three digits for values of $H_{1/2}(\hat{\mathbf{U}}_N)/N$ and D_{avg}, rounded to two digits for matrix elements of the solution point.

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.756$; $D_{avg} = 0.098$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 0 1 0	1.00 0.00 0.32 0.00
0 1 0 1	0 1 0 1	0.00 1.00 0.00 1.00
0 0 1 0	0 0 0 0	0.00 0.00 0.68 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.748$; $D_{avg} = 0.105$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 1 0 0	1.00 0.38 0.00 0.00
0 1 0 0	0 0 0 0	0.00 0.62 0.00 0.00
0 0 1 1	0 0 1 1	0.00 0.00 1.00 1.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.740$; $D_{avg} = 0.112$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0	1 0 1 0	1.00 0.00 0.45 0.00
0 1 0 1	0 1 0 1	0.00 1.00 0.00 1.00
0 0 1 0	0 0 0 0	0.00 0.00 0.55 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.665$; $D_{avg} = 0.147$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 0 0	1.00 1.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 0	0 0 1 1	0.00 0.00 1.00 0.54
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.46

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.647$; $D_{avg} = 0.154$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 0 0	1 1 0 0	1.00 1.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 1 0	0 0 1 1	0.00 0.00 1.00 0.66
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.34

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.483$; $D_{avg} = 0.175$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 1 0	1 0 1 0	1.00 0.00 1.00 0.00
0 1 0 1	0 1 0 1	0.00 1.00 0.00 1.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.480$; $D_{avg} = 0.182$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 1 0	1 1 1 0	1.00 0.06 1.00 0.00
0 1 0 1	0 0 0 1	0.00 0.94 0.00 1.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.328$; $D_{avg} = 0.259$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 1 0	1 1 1 1	1.00 1.00 1.00 0.26
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.74

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.243$; $D_{avg} = 0.308$:

Feasible Vertex	Non-Feas. Vert.	Solution Point

1 1 1 0	1 1 1 1	1.00 1.00 1.00 0.66
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 0	0 0 0 0	0.00 0.00 0.00 0.00
0 0 0 1	0 0 0 0	0.00 0.00 0.00 0.34

A.4 Results for SDE-3

The forward test channel transition matrices for an optimal solution of SDE-3 design problem for various values of average distortion D_{avg} are listed. The *Feasible Vertex* denotes a certain transition matrix W_N which satisfies the average distortion constraint. The *Non-Feasible Vertex* is a hamming distance 2 neighbor of the feasible vertex and does not satisfy the average distortion constraint. *solution point* denotes the forward test channel transition matrix for an optimal SDE-3 with the given constraint. Solution point is between feasible and non-feasible vertices.

For all the results below, the source is assumed to be a BMS-p with $p = P(U = 1) = 0.35$. Floating point numbers are rounded to three digits for values of $H_{1/2}(\hat{\mathbf{U}}_N)/N$ and D_{avg}, rounded to two digits for matrix elements of the solution point.

The codebook $\mathcal{C}$ is arranged as,

$$\begin{aligned} C_1 &= (000) \\ C_2 &= (001) \\ C_3 &= (010) \\ C_4 &= (100) \\ C_5 &= (011) \\ C_6 &= (101) \\ C_7 &= (110) \\ C_8 &= (111) \end{aligned} \tag{A.2}$$

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.813$; $D_{avg} = 0.056$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 1 0 0 0 0 0 0	0 1 0 0 0 0 0 0	0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 1 0 0 0 0 0	0 0 1 0 0 0 0 0	0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00
0 0 0 1 0 1 1 0	0 0 0 1 0 1 1 0	0.00 0.00 0.00 1.00 0.00 1.00 1.00 0.00
0 0 0 0 1 0 0 0	0 0 0 0 1 0 0 1	0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.20
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.80

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.654$; $D_{avg} = 0.112$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 0 0 0 0 0 0	1 0 0 1 0 0 0 0	1.00 0.00 0.00 0.08 0.00 0.00 0.00 0.00
0 1 0 0 1 1 0 1	0 1 0 0 1 1 0 1	0.00 1.00 0.00 0.00 1.00 1.00 0.00 1.00
0 0 1 0 0 0 1 0	0 0 1 0 0 0 1 0	0.00 0.00 1.00 0.00 0.00 0.00 1.00 0.00
0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.92 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.485$; $D_{avg} = 0.168$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 1 1 0 0 0 0	1 0 1 1 0 0 0 0	1.00 0.00 1.00 1.00 0.00 0.00 0.00 0.00
0 1 0 0 1 1 0 0	0 1 0 0 1 1 0 1	0.00 1.00 0.00 0.00 1.00 1.00 0.00 0.14
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 0	0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.86
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.320$; $D_{avg} = 0.238$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 0 1 1 0 0 1 0	1 1 1 1 0 0 1 0	1.00 0.09 1.00 1.00 0.00 0.00 1.00 0.00
0 1 0 0 1 1 0 1	0 0 0 0 1 1 0 1	0.00 0.91 0.00 0.00 1.00 1.00 0.00 1.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

$H_{1/2}(\hat{\mathbf{U}}_N)/N = 0.162$; $D_{avg} = 0.308$:

Feasible Vertex	Non-Feas. Vert.	Solution Point
1 1 1 1 1 1 1 0	1 1 1 1 1 1 1 1	1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.02
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.98

REFERENCES

[1] Hellman M. E. "Convolutional Source Encoding," *IEEE Transactions On Information Theory*, vol. IT-21, pp. 651–656, 1975.

[2] Koshelev V. N. "Direct Sequential Encoding and Decoding for Discrete Sources," *IEEE Transactions On Information Theory*, vol. IT-19, pp. 340–343, 1973.

[3] Arıkan E. Merhav N. "Joint Source-Channel Coding and Guessing with Application to Sequential Decoding," *Submitted to IEEE Transactions On Information Theory*, vol. , 1997.

[4] Omura J. K. Viterbi A. J. *Principles Of Digital Communication And Coding.* McGraw-Hill, New York, 1979.

[5] Fano R. M. "A Heuristic Discussion of Probabilistic Decoding," *IEEE Transactions on Information Theory*, vol. IT-9, pp. 64–73, 1963.

[6] Arıkan E. " An Inequality on Guessing and its Application to Sequential Decoding," *IEEE Transactions on Information Theory*, vol. IT-42, pp. 99–105, 1996.

[7] Shannon C. E. "Prediction and Entropy of Printed English," *Bell System Technical Journal*, vol. 30, pp. 50–64, 1951.

[8] Massey J. L. "Variable-Length Codes and the Fano Metric," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 196–198, 1972.

[9] Hellman M. E. "On Using Natural Redundancy for Error Detection," *IEEE Transactions On Communications*, vol. COM-22, pp. 1690–1693, 1974.

[10] Gibson J. D. Khalid S., Liu F. "A Constrained Joint Source/Channel Coder Design," *IEEE Journal On Selected Areas In Communications*, vol. 12, pp. 1584–1593, 1994.

[11] Wei V. K. Zhang Z., Yang E. H. "The Redundancy of Source Coding with a Fidelity Criterion-Part One:Known Statistics," *IEEE Transactions On Information Theory*, vol. IT-43, pp. 71–91, 1997.

[12] Joy A. T. Thomas M. C. *Elements Of Information Theory.* A Wiley-Interscience Publication, 1991.