The binary defect channel: an information-theoretical approach

Janssen, J.J.M.

Award date:
1987

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering

Group: Information and Communication Theory

graduate report

THE BINARY DEFECT CHANNEL

an information-theoretical approach

by

J.J.M. Janssen

coach : Prof. Dr. Ir. J.P.M. Schalkwijk

EINDHOVEN, The Netherlands, August 1987

SUMMARY

In this graduate report a new information-theoretical approach towards the Binary Defect Channel (BDC) is presented. In particular the coding situation is analized in which the locations and types of the defects are available to the encoder and not to the decoder. The approach originates from Schalkwijk in 1986.
It is based on Shannon's results on channels with side information at the transmitter. By using this theorem, the BDC with side information at the transmitter can be replaced by an equivalent Discrete Memoryless Channel (DMC) with noiseless feedback.
The capacity of the derived DMC and the BDC are the same. Furthermore, codes for the DMC can be translated into codes for the original BDC with the same probability of error. Therefore the analysis is focused on the derived DMC. It is assumed that the probability of occurrence of 0-defects and 1-defects are independent and identically distributed.
In case the number of defects f in n memory cells is fixed, the capacity $C_n(f)$ is determined for f=2 and $2 \leq n \leq 7$. It is shown that the upperbound $\overline{C} = 1 - f/n$ is achievable in case $f \in \{1,n-1,n\}$. For all other values of f it is proved that the upperbound is not achievable.
In case of a more general probabilistic model of a memory cell, the capacity C_n is determined for $n = \{1,2,3\}$. Moreover, an upperbound for arbitrary n is derived, which coincides with C_1 and C_2. This upperbound is thightened by using the results of the fixed-number-of-defects model. It is also indicated in this report how we have to choose the

storage strategy in order to achieve capacity in the cases mentioned above.

These storage strategies however cannot avoid errors which occur consequent on defects. Therefore, an error-correcting code is necessary. As the derived DMC with noiseless feedback is symmetrical, so-called, Multiple-Repetition Feedback Codes (MRFC) can be used. Their performance curves are evaluated in case n = 1,2. Furthermore a proof is given of the error-correction capability of the MRFC scheme. Finally the complexity of the coding scheme to be considered is scrutinized.

ACKNOWLEDGEMENTS

I wish to thank Prof.Dr.Ir. J.P.M. Schalkwijk for his supervision during my project. I am grateful to him for introducing me into the interesting field of Information and Communication Theory. Special thanks are due to Dr.Ir. A.J. Vinck for his valuable suggestions during the course of my work, Dr. K.A. Post for both his advice and willingness to discuss in detail every point in this report, Ir. W.M.C.J. van Overveld for suggesting an alternative proof of *theorem* 3.3, all the members of the group Information and Communication Theory: C.M. Bijl-Wind, Ir. W.M.C.J. van Overveld, H.M. Creemers, Ir. J.E. Rooyackers, Prof.Dr.Ir. J.P.M. Schalkwijk, Ir. T.J. Tjalkens, Dr.Ir. A.J. Vinck and Dr.Ir. F.M.J. Willems for their support and Yvonne Janssen for her assistance in preparing this report.
Last but not least I wish to thank Louise, my sisters Marga and Yvonne, and my parents for being there whenever I needed them.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

A-MRFC	Asymmetric MRFC
BDC	Binary Defect Channel
BDC^n	n^{th} power of BDC
BEC	Binary Erasure Channel
bits	binary units of information
BSC	Binary Symmetric Channel
iff	if and only if
DMC	Discrete Memoryless Channel
FSC	Finite State Channel
MDS	Maximum Distance Separable
MRFC	Multiple-Repetition Feedback Code
RHS	right hand side
SS	Strict-Sense
SS-MRFC	Strict-Sense symmetric MRFC
w.l.o.g.	without loss of generality
WS	Wide-Sense
WS-MRFC	Wide-Sense symmetric MRFC

LIST OF SYMBOLS

$\triangleq$	equality by definition
$\oplus$	bit-by-bit modulo-2 addition of two binary n-tuples
$\|A\|$	cardinality of set A
$\binom{n}{f}$	binomial coefficient; n!/{f!.(n-f)!}
$\max_{i}$	maximization over the range of i
$\mathbb{N}$	set of natural numbers
$\mathbb{R}^{+}$	set of positive rational numbers (0 not included)
$\forall$	for all
$\lceil x \rceil$	smallest integer less than or equal to x
$\lfloor x \rfloor$	largest integer greater than or equal to x
$\hat{}$	denotes that the corresponding bit is in error
$(\underline{x}\cdot\underline{y})$	vector product of $\underline{x}$ and $\underline{y}$
(i,j)	error type
$tr(\underline{\underline{X}})$	trace of matrix $\underline{\underline{X}}$
$\subseteq$	subset
$\langle 0,1 \rangle$	all rational numbers between 0 and 1
A	input alphabet of BDC
A^{h}	input alphabet of K'
A^{n}	input alphabet of BDC^{n}
A_{t}	set of elements of I_{m} that appear in the tail sequence
a	$\|A\|$; cardinality of A
	element of $\mathbb{R}^{+}$

$a_f(\underline{y}), a'_f(\underline{y})$	number of elements $\underline{x}_t$ in $\underline{\underline{X}}$ that lead to $\underline{y}$ in case f defects occur
$a^*_f(\underline{y}^*)$	element of optimal $a_f(\underline{y})$-distribution
α	variable
α_1, α_{1opt}	rational number
B	output alphabet of BDC
B^n	output alphabet of BDC^n
b	$\|B\|$; cardinality of B
	element of $\mathbb{R}^+$
C_n	channel capacity of BDC^n
$\overline{C}$	asymptotic upperbound for C_n
C_t	channel capacity of K_t
C	channel capacity of a FSC
$C_n(f)$	channel capacity of BDC^n in case f defects occur
$\overline{C}_n(f)$	upperbound for $C_n(f)$
$\overline{C}_n$	upperbound for C_n
$\overline{C}^t_n$	tightened $\overline{C}_n$
c, c_i	type of error
D_f	set of defect-patterns with $w(\underline{d}) = f$
$\underline{d}$	defect-pattern
d_i	i^{th} component in $\underline{d}$
d	integer ≥ 0
$d_H(\underline{y}_i, \underline{y}_j)$	mutual Hamming-distance of $\underline{y}_i$ and $\underline{y}_j$
E, E', E'_{tc}, E'_{tr}	error matrix
E_m, E'_m	message error matrix
E'_t, E_t, E^0_t, E^1_t	tail error matrix
ϵ	rational number in $\langle 0,1 \rangle$

$\in$	an element of
e,e'	number of errors
e_c	number of errors of type c
$\underline{e}$,$\underline{e}'$	error vector
$e_{i,j}$	number of errors of type (i,j)
e_m	number of message sequence errors
e_t,e_t'	number of tail sequence errors
f	number of defects
f(x)	function with argument x
f'(x)	derivative of f(x)
ℓ_e	error fraction
ℓ_{ec}	fraction of errors of type c
ℓ_{e_o}	fraction of correct transmissions
g_t	probability of being in state t
g	probability
g(f)	probability of defect-pattern $\underline{d} \in D_f$
h	$\|S\|$; cardinality of S
h(x)	binary entropy function
I_m	{1,2,...,m}; set of indices
$\underline{i}$	binary n-tuple
i	index
	elememt of I_m
	binary symbol
$\underline{j}$	binary n-tuple
j	index
	element of I_m
	binary symbol

K_t	component channel corresponding to state t
K	Finite State Channel
K_u	Discrete Memoryless Channel
K'	DMC; superchannel
$\mathcal{K}$	characteristic matrix of repetition factors of A-MRFC
$k_{j,i}$	repetition factor corresponding with error type (i,j)
k, k_{opt}	characteristic repetition factor of SS-MRFC
$\underline{k}, \underline{k}_{opt}$	characteristic vector of repetition factors of WS-MRFC
$k_c, k_{c_{opt}}$	repetition factor corresponding with error of type c
k_{max}	$\max_c k_c$
$\mathcal{L}(\alpha_1, k)$	langrangian
$^2\log$	logarithm to the base-2
$^m\log$	logarithm to the base-m
ln	logarithm to the base-e
ℓ	information length of MRFC
λ, λ_{opt}	langrange multiplier
l	element of I_m
M	$\lvert S_f(\underline{X}) \rvert$; cardinality of $S_f(\underline{X})$
M(n,f)	minimum number of elements of length n ($n \geq 2$) in a f-defect compatible set
M_k	number of elements in a f-defect compatible set
$M_\ell(k)$	number of ℓ-tuples in MRFC
m	cardinality of V_n
$\overline{m}$	sequence of symbols from I_m
N	blocklength of a code
n	length of a storage strategy

$\bar{n}_H$	average number of bits of a Huffman code
P	$[P_{\underline{i},\underline{j}}]$; transition probability matrix
$P_{\underline{i},\underline{j}}$	$P(\underline{j}\|\underline{i})$; a component of P
$P(\underline{j}\|\underline{i})$	channel transition probability; probability of channel output $\underline{j}$ given input $\underline{i}$
$P_{tx}(y)$	transition probability from x to y in state t
$\overline{P_x(y)}$	$P_{tx}(y)$ averaged over all t in S
$P_{t\underline{x}}(\underline{y})$	transition probability from $\underline{x}$ to $\underline{y}$ with defect-pattern t
p	defect fraction
q	1 - p
$R(N), R_{opt}(N)$	information rate for finite blocklength N
$R(\infty), R_{opt}(\infty)$	information rate for infinite blocklength
$R_n(\underline{\underline{X}}_i)$	rate of BDC^n with superinputs according to $\underline{\underline{X}}_i$
$R_k(f)$	rate of Kusnetsov's scheme with f defects occurring
R_{kus}	$R_k(2)$
$R_{kus}(up)$	upperbound for R_{kus}
R_n	information rate of BDC^n
$\underline{r}_1, \underline{r}_2$	binary (d+1)-tuple
r	element of I_m
$r_{\underline{X}}(y)$	transition probability from $\underline{X}$ to y
$r_{\underline{\underline{X}}}(\underline{y})$	transition probability from $\underline{\underline{X}}$ to $\underline{y}$
S	state space
$S_f(\underline{\underline{X}})$	set of outputs resulting from $\underline{\underline{X}}$ in case f defects occur
$S(\underline{\underline{X}})$	set of outputs resulting from $\underline{\underline{X}}$

$\underline{s}_1, \underline{s}_2$	binary (d+1)-tuple
$\bar{\underline{s}}, \underline{s}$	sequence of symbols from I_m
T_n	positive integer
t	element of S
t_i	i^{th} element of defect-pattern t
U	number of messages
u	message index
V_n	set of all possible binary n-tuples
v	message index
	element of I_m
$w(\underline{d})$	weight of defect-pattern $\underline{d}$
$\underline{X}, \underline{X}_i$	superinput vector of K'
$\underline{\underline{X}}, \underline{\underline{X}}_i, \underline{\underline{X}}', \underline{\underline{X}}^*$	superinput matrix of K'
X	set of binary vectors
x	binary input symbol
	element of $\mathbb{R}^+$
x_1, x_2	element of $\mathbb{R}^+$
$\underline{x}$	binary input vector
x_i	i^{th} component of $\underline{x}$
x_{it}	t^{th} component of $\underline{X}_i$
$\underline{x}_{it}$	t^{th} component of $\underline{\underline{X}}_i$
$\hat{x}, x$	elememt of I_m
y	binary output symbol
$\underline{y}, \underline{y}_j', \underline{y}_j, \underline{y}_j^*$	binary output vector
y_i	i^{th} component of $\underline{y}$
$\underline{y}^{\leftarrow}$	vector which results from $\underline{y}$ by flipping over the elements
Z, Z_{min}	number of cells in the memory

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This graduate report is concerned with the problem of reliable storage of information in a imperfect semiconductor computer memory.
In the semiconductor memory industry the need for higher capacity memory is being addressed both by increasing the number of memory cells per area (density) and by increasing the area consumed by the memory. Despite the improvements in process technology and sophisticated circuit design, serious reductions in the yield (the proportion of working chips) results as these steps are taken. To solve the problem of increasing the capacity of memories while maintaining a high yield, manufactures sometimes incorporate redundancy in the design of their memories. Most of these designs incorporate redundant columns (bit lines) and rows (word lines). The redundant elements are switched in to replace faulty columns and rows that are detected during initial testing of the memory circuit. In this thesis, we examine the use of codes as an alternative to some of these methods of switching.

We consider a binary memory as a set of individual units (cells) in which an arbitrary binary value can be stored. The types of imperfections to be considered are both permanent and affecting individual cells. A cell is called defective, if it always produces a '0' (0-defect) or a '1' (1-defect) regardless of the binary symbol originally stored in it. In the literature they often are denoted as stuck-at-0 and stuck-at-1 defects, respectively.
A memory is called defective if one or more cells in the set are defective.

By testing the memory it is possible to determine the locations and the types of the defective cells. We examine how the information, that describes the state of the defects, can be used in the encoding of messages.

As a storage process can be conceived as a transmission in time, we introduce in Chapter 1 the, so-called, Binary Defect Channel (BDC) model of a generic memory cell. We already mentioned, that we are interested in the coding situation in which the encoder has knowledge of the locations and the types of the defective cells. This situation is often referred to a channel with side information at the transmitter.

In Chapter 2 we show that such a channel can be replaced by an equivalent Discrete Memoryless Channel (DMC) with noiseless feedback.

As the maximum number of messages that can be reliably transmitted over both the derived DMC and the BDC are the same, we focuss in Chapter 3 on the determination of this maximum number, by considering the derived DMC.

Furthermore, codes for the DMC can be translated into codes for the original BDC with the same probability of error. As the derived DMC with noiseless feedback is symmetric, so-called, Multiple-Repetition Feedback Codes (MRFC) can be used. This is the topic of Chapter 4.

In Chapter 5 at last we discuss the complexity of the coding scheme presented in former chapters.

PRELIMINARIES

In this thesis the number of messages U that can be reliably stored into a memory of length N (or equivalently : the number of messages U that can be reliably transmitted over a DMC in N transmissions), is determined by the rate function, denoted by R(N).
The rate R(N) is expressed in bits per memory cell and defined as follows

$$R(N) \triangleq \frac{{}^{2}\log U}{N} \tag{P1}$$

The maximum value of R(N) for asymptotic blocklength is called the channel capacity or in short the capacity of a memory (channel).

Furthermore, we distinguish three types of symmetric DMC's : Strict-Sense (SS) symmetric , Wide-Sense (WS) symmetric and A-symmetric (A). Notice that a DMC is characterized by the input and output alphabet and the transition probability matrix $P = [P_{\underline{i},\underline{j}}]$. The symmetric DMC's, to be considered in this thesis, have input and output symbols consisting of binary n-tuples ($n \geq 1$). The entries of the matrix are defined by $P_{\underline{i},\underline{j}} \triangleq P(\underline{j}|\underline{i})$, where $P(\underline{j}|\underline{i})$ denotes the probability of channel output $\underline{j}$ occurring, given a channel input $\underline{i}$.

SS-symmetric

These channels are such that the input and output alphabet V_n are the same. V_n is defined as the set of all possible binary n-tuples. Consequently, the cardinality of V_n is equal to $m = 2^n$.
In addition the entries in P are restricted to the form :

$$P(\underline{j}|\underline{i}) = \begin{cases} 1 - \epsilon & \underline{i} = \underline{j} \text{ and } \underline{i},\underline{j} \in V_m \\ \\ \epsilon/(m-1) & \underline{i} \neq \underline{j} \text{ and } \underline{i},\underline{j} \in V_n \end{cases} \qquad \text{(P2)}$$

$0 < \epsilon < 1$

WS-symmetric

These channels have input and output alphabets V_n. The transition probability $P(\underline{j}|\underline{i})$ depends on the n-tuple resulting from the bit-by-bit modulo-2 addition of $\underline{j}$ and $\underline{i}$. That is,

$$P(\underline{j}|\underline{i}) = P(\underline{i}|\underline{j}) = P(\underline{j} \oplus \underline{i}) \;, \qquad \forall \; \underline{i},\underline{j} \in V_n \qquad \text{(P3)}$$

where $\oplus$ denotes the bit-by-bit modulo-2 addition of two binary n-tuples.

A-symmetric

Again the input and output alphabet are the same (V_n). All the entries in matrix P however may be distinguished.

1 CHANNEL MODEL AND CODING SITUATIONS

1.1 BDC-MODEL

In the Binary Defect Channel model (BDC) we consider the storage process of information in an individual binary memory cell.
Figure 1.1 gives a schematic representation of the generic memory cell, i.e. the BDC.

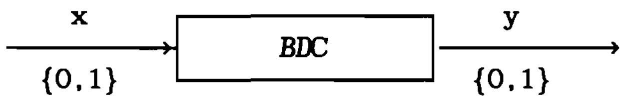


Figure *1.1* : *Binary Defect Channel.*

The channel can be in three different states : the C-state, 0-state and 1-state, corresponding to a non-defective (correct) , a stuck-at-0 and a stuck-at-1 cell respectively.
Depicted in *figure* 1.2 are the three channel states in detail.

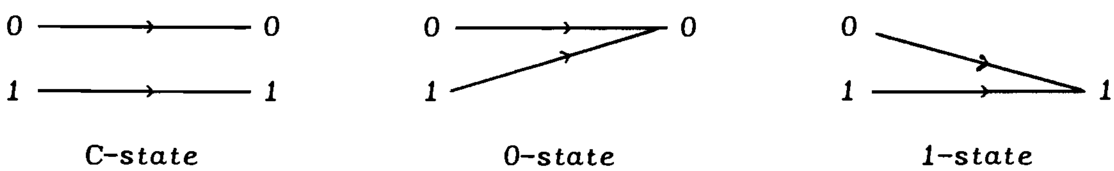


Figure 1.2 : Three states of the BDC.

A stochastic variable $x \in \{0,1\}$ is stored into the cell during the writing cycle. In the reading cycle we obtain the stochastic variable $y \in \{0,1\}$, according to eq.(1.1).

$$y = \begin{cases} x, & \text{if BDC in C-state} \\ 0, & \text{if BDC in 0-state} \\ 1, & \text{if BDC in 1-state} \end{cases} \qquad (1.1)$$

In the subsequent of this thesis it turns out to be convenient, to conceive the BDC as a member of a class of Finite State Channels (FSC).

Let A be the input alphabet with cardinality $|A| = a$, B the output alphabet with $|B| = b$ and S the state space with $|S| = h$. Furthermore we denote a component channel corresponding to the state $t \in S$ by K_t and the probability of being in t by g_t. The transition probabilities of the DMC K_t are denoted by $P_{tx}(y)$, with $x \in A$, $y \in B$ and $t \in S$.
Then, a FSC K is completely described by :

$$K=\{(K_t,g_t)\,|\,t \in S\} \qquad (1.2\text{-a})$$

$$K_t=(A,[P_{tx}(y)],B) \qquad (1.2\text{-b})$$

Referring to the *figures 1.1* and *1.2 TABLE 1.1* gives the channel parameters in case K=BDC.

TABLE 1.1 : Channel parameters of BDC.

alphabets	index $t \in S$
$A = \{0,1\}$, $a = 2$	$t = 1$ → C-state
$B = \{0,1\}$, $b = 2$	$t = 2$ → 0-state
$S = \{1,2,3\}$, $h = 3$	$t = 3$ → 1-state

transition probabilities

C-state

x \ y	0	1
0	1	0
1	0	1

0-state

x \ y	0	1
0	1	0
1	1	0

1-state

x \ y	0	1
0	0	1
1	0	1

The only parameter we still have to specify, is the probability distribution g_t.

In the subsequent of this thesis we assume, unless otherwise is stated, that :

1) The state probability g_t is independent of former states and of input and output history.

2) If the defect fraction is equal to p, then :

$$g_1 = q = 1 - p$$

$$g_2 = p/2$$

$$g_3 = p/2$$

In a memory consisting of N cells the state probabilities of the individual cells are supposed to be independent and identically distributed, according to 1) and 2) above. In this way, the storage process of binary values in consecutive memory cells can be conceived as a multiple use of the BDC.

1.2 CODING SITUATION

In order to reliably store information into a defective memory, we have to use coding strategies.
Figure 1.3 depicts the general coding scheme.

Figure 1.3 : General coding scheme.

A message u, $u \in \{1,2,\dots,U\}$ is transduced by the encoder into a codeword $\underline{x}$, $\underline{x}=(x_1,x_2,\dots,x_N)$, $x_i \in A$ and $1 \leq i \leq N$. This codeword $\underline{x}$ is stored into the memory, which consists of N individual cells.
By reading the memory, we obtain the word $\underline{y}$, $\underline{y} = \{y_1,y_2,\dots,y_N\}$, $y_i \in B$ and $1 \leq i \leq N$. From this word $\underline{y}$ an estimate v, $v \in \{1,2,\dots,U\}$ is made by the decoder.
Notice the great similarity of *figure 1.3* to general transmission schemes, as we already pointed out in the introduction. Because of this, we will often use in this thesis concepts as transmitter and receiver in the obvious way.

In subsequent sections we outline some coding situations. For this we need *definition 1.1*.

Definition 1.1 :

The types of the defects and their locations in the memory are called the defect-pattern. We denote a defect-pattern as a vector $\underline{d}$ = $\{d_1, d_2, \ldots, d_N\}$ *of length N and* $d_i \in S$ *for* $1 \leq i \leq N$.

The coding situations are categorized by the way information about the defect-pattern is used in the storage process.
There are eventually four situations to be considered :

1.2.1 UNKNOWN DEFECTS

In case neither the encoder nor the decoder has information concerning the defect-pattern the BDC changes in a DMC K_u, described by the eq.'s (1.3-a), (1.3-b) and Section 1.1.

$$K_u = (A, [\overline{P_x(y)}], B) \tag{1.3-a}$$

$$\overline{P_x(y)} \triangleq \sum_{t \in S} g_t . P_{tx}(y) \tag{1.3-b}$$

In other words the transition probability $\overline{P_x(y)}$ from input $x \in A$ to output $y \in B$ is the average probability over the different channel states in S.

The equivalent channel K_u is a Binary Symmetric Channel (BSC) with crossover probability p/2 (p is the fraction of defects). From Shannon's channel coding theorem [GAL68, p. 81] we know that there exist codes, that allow essentially error-free storage at rates up to the channel capacity, denoted by C_1 :

$$C_1 = 1 - h(p/2) \quad \text{bits/memory cell} \tag{1.4}$$

Where $h(x) = -x.\log x - (1-x).\log(1-x)$ is the binary entropy function. Note that for $p = 1/2$ we can store at most $1 - h(1/4) = 0.18872$ bits per memory cell. The remaining fraction 0.31128 of non-defective memory space is neccessary to inform the reader about the defect-pattern.
In practice the remaining fraction is used to store redundant checkbits of an error-correcting code.

1.2.2 DEFECTS KNOWN TO BOTH ENCODER AND DECODER

In case both the encoder and decoder know the defect-pattern in advance, the coding scheme in *figure 1.3* changes into *figure 1.4*.

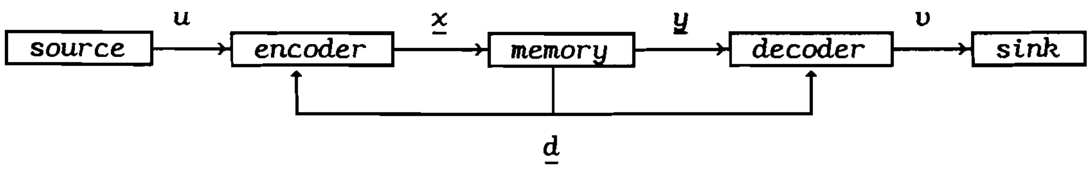


Figure 1.4 : Defect-pattern is known to both encoder and decoder.

Obviously, one can store information with rates up to

$$\bar{C} = \sum_{t \in S} g_t . C_t \qquad \text{bits/memory cell} \qquad (1.5)$$

The capacity of the corresponding component channel is denoted by C_t ($t \in S$).
In case of the BDC we obtain from the trivial C_C = 1 bit/cell and C_0 = C_1 = 0 bit/cell :

$$\bar{C} = 1 - p \qquad \text{bits/memory cell} \qquad (1.6)$$

All non-defective memory space can be used for storage of information. Therefore the bar in the eq.'s (1.5) and (1.6) denotes that $\bar{C}$ is an upperbound for the rate in all different coding situations.

1.2.3 DEFECTS KNOWN TO THE DECODER ONLY

In situations in which only the decoder has total knowledge of the defect-pattern, the general coding scheme in *figure 1.3* changes into *figure 1.5*.

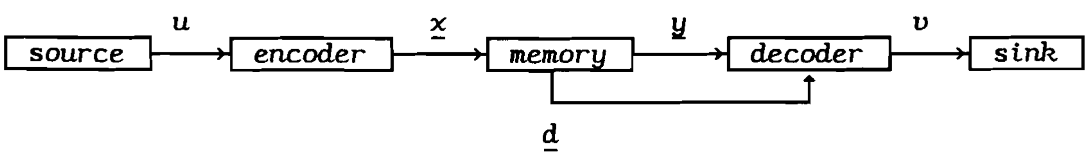


Figure 1.5 : Defect-pattern is known to the decoder only.

The decoder knows the states of the memory and recognizes the stuck-at cells. Now the BDC can be conceived as a Binary Erasure Channel (BEC), depicted in *figure 1.6*.

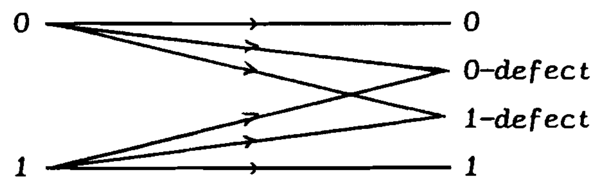


Figure 1.6 : Binary Erasure Channel.

Using erasure-decoding schemes rates up to the channel capacity $\bar{C}$ are achievable. The defect information at the decoder can for example be obtained by rewriting the memory with $\overleftarrow{\underline{y}}$, which results by flipping over the elements of $\underline{y}$, and comparing the new output to $\underline{y}$.

1.2.4 DEFECTS KNOWN TO THE ENCODER ONLY

Before operation the encoder knows the state of the FSC. This situation, depicted in *figure 1.7*, is often referred to a channel with side information at the transmitter.

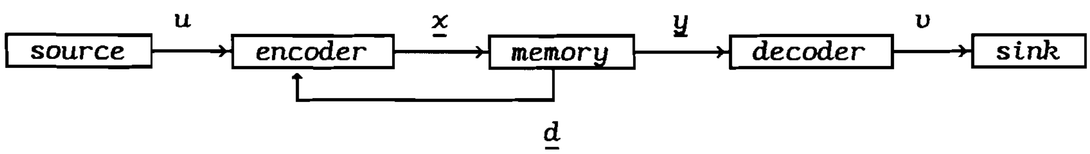


Figure 1.7 : Defect-pattern is known to the encoder only.

As the coding situations, outlined in the Sections 1.2.1 until 1.2.3, are solved we restrict ourselves in the following to the scheme with side information at the transmitter.

First of all, we place this subject in a historical perspective.

1.3 HISTORICAL PERSPECTIVE

The problem of coding for the BDC with side information at the transmitter was originated by Kusnetsov and Tsybakov [KUS74]. They confirmed that, using the information of the defect-pattern in the encoding process, higher efficiencies can be achieved, than the best error-correcting codes.
In fact, by using a random coding argument they derived a lower bound for the efficiency of the storage process. By allowing the size of the memory unit N to become large, this lower bound approaches the upper bound $\bar{C}$, eq.(1.6), arbitrarily tight.
That is, no non-defective memory space has to be wasted in order to inform the reader (decoder) about the defect-pattern.
This remarkable result however, only shows the existence of codes, that are capable of storing information without error for any rate $R \leq \bar{C}$.
Moreover, they proved that such codes can be found within the class of additive coding.

In case the defect fraction $p = 1/N$ and $p = (N-1)/N$ capacity-achieving additive codes were presented, while for $p = 2/N$ a suboptimal code construction was given.

This paper [KUS74] initiated the search for codes that are capable of correcting a fixed number f of defects ([BEL77], [LOS78], [BUS84], [PUL85], [VIN86] and [PUL87]). The main point in this search is the existence of , so-called, separable f-defect compatible matrices, which has not been solved yet. Moreover, these codes will not have a significant impact on the practice of coding for computer memories, due to their non-linearity.
In [TSY75-2] Tsybakov looks at linear block codes for the binary-defect problem and shows that the asymptotic rate $R(\infty)$ for the best linear block code, under the criterion of [KUS74], satisfies

$$R(\infty) \leq 1 - h\left(\frac{1-\sqrt{1 - 2p}}{2}\right) \tag{1.7}$$

for $0 \leq p \leq 1/2$. Since it is generally true that $1-p$ is greater than the RHS of eq.(1.7), we see that these codes are suboptimal by this criterion.
In the same year Tsybakov [TSY75-1] introduces the problem of coding for binary memories with both defects and random errors. Since then almost all attention has been focused on this problem using linear block codes ([KUS78], [HEE83-2], [CHE85], [CHE86], [VIN86] and [PUL87]).

1.4 MOTIVATION

In this thesis we return to the basic problem stated by Kusnetsov and Tsybakov [KUS74]. Instead of considering a defective memory for fixed numbers of defects, we use the more general probabilistic model stated in Section 1.1.
Till now, all results in this field (see Section 1.3) were obtained from an algebraic viewpoint.

Here we present a completely different information-theoretical approach towards this problem, due to Schalkwijk [SCH86].
This new approach is based on a theorem of Shannon's [SHA58], concerning channels with side information at the transmitter.
By using this theorem, the BDC with side information at the transmitter can be replaced by an equivalent DMC with noiseless feedback.
The capacity of the derived DMC and the BDC are the same. Furthermore, codes for the DMC can be translated into codes for the original BDC with the same probability of error.
As the derived DMC with noiseless feedback is SS- or WS-symmetrical, so-called, Multiple-Repetition Feedback Codes (MRFC) can be used, that achieve capacity for particular values of p.

2 SCHALKWIJK'S APPROACH

2.1 SHANNON'S 'SIDE-INFORMATION' THEOREM

The new information-theoretical approach, due to Schalkwijk [SCH86], is based on Shannon's results on channels with side information at the transmitter [SHA58]. *Theorem 2.1* describes the main points of these results.

Theorem 2.1 :

a. *The capacity C of a FSC K with side state information defined by the eq.'s (1.2-a) and (1.2-b) in Section 1.1, is equal to the capacity of the DMC K' (without side information) with the same output alphabet B and an input alphabet* A^h *with* a^h *input letters* $\underline{X} = (x_1, x_2, \ldots, x_h)$, *where each* $x_t \in A$. *Note that* $|S| = h$. *The transition probabilities* $r_{\underline{X}}(y)$ *for the channel K' are given by* :

$$r_{\underline{X}}(y) = \sum_{t \in S} g_t . P_{tx_t}(y) \qquad (2.1)$$

In short K' is denoted as :

$$K' = (A^h, [r_{\underline{X}}(y)], B) \qquad (2.2)$$

b. *Any coding system for K' can be translated into an equivalent coding system for K with the same probability of error.*

For a proof we refer to Shannon's paper [SHA58]. Here, we restrict ourselves to an illustration (*figure 2.1*) of this theorem. That is, a hypothetical configuration for K' with respect to K.
To emphasize the difference between the channels K' and K, we call K' a superchannel. Likewise, the inputs $\underline{X}$ of K' will be called superinputs.

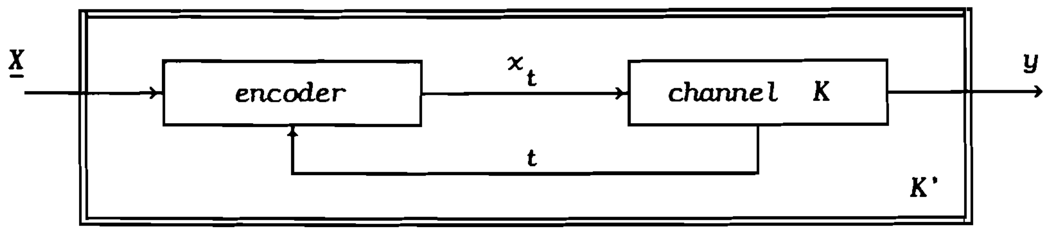


Figure 2.1 : Illustration of Shannon's theorem.

The encoder in *figure 2.1* selects from the superinput $\underline{X}$ the component x_t, that corresponds to the state t∈S of channel K.
In other words, the encoder is only a translation without memory effects, depending on the state of K. This means, codes for K' are just another way of describing certain of the codes for K.
Obviously from above, the statistical situations for K and K' with the translated code are identical, which shows the validity of *theorem 2.1*.

In summary, *theorem 2.1* reduces the analysis of the given channel K with side information to that for a memoryless channel K' with more input letters, but without side information.
One uses known methods to determine the capacity of this derived channel K' and this gives the capacity of the original channel. Furthermore, codes for the derived channel may be translated into codes for the original channel with identical probability of error.

2.2 SIMPLE USE OF THE BINARY DEFECT CHANNEL (BDC^1)

It was Schalkwijk's idea to use the '*side-information*' *theorem* to find constructive code strategies for defective computer memories, in case the writer knows the defect-pattern and the reader does not.

Referring to eq. (2.1) and *TABLE 1.1*, the derived DMC K' has $a^h = 8$ different superinputs $\underline{X}_i = (x_{i1}, x_{i2}, x_{i3})$ with $x_{it} \in \{0,1\}$, $t \in S$, and two outputs y = 0 and y = 1.
From the transition probabilities $r_{\underline{X}}(y)$ the superinputs can be divided into two subsets :

$$\underline{X}_i = (0, x_{i2}, x_{i3})\ ^{*)} \quad \begin{cases} r_{\underline{X}_i}(0) = 1 - p/2 \\ \\ r_{\underline{X}_i}(1) = p/2 \end{cases}$$

$$\underline{X}_i = (1, x_{i2}, x_{i3})\ ^{*)} \quad \begin{cases} r_{\underline{X}_i}(0) = p/2 \\ \\ r_{\underline{X}_i}(1) = 1 - p/2 \end{cases} \tag{2.3}$$

*) $x_{i2}, x_{i3} \in \{0,1)$

The capacity of the superchannel K' (and of the BDC by *theorem 2.1*)) can be computed e.g. by using Arimoto-Blahut's algorithm [VIT79, p.207].

However, using the following property of input-probability assignments, that achieve capacity on a DMC, the capacity of K' is easily found.

Theorem 2.2 :

Achieving capacity on a DMC, it is sufficient to use as many inputs as there are outputs.

A proof of *theorem* 2.2 is given by Gallager [GAL68, p. 96 corollary 3].

By inspection it is readily verified that two superinputs from different subsets in eq. (2.3) form a capacity-achieving set of superinputs; e.g. $\underline{X}_1$ = (0,0,1) and $\underline{X}_2$ = (1,0,1).
Now, the superchannel K' results in a BSC with crossover probability p/2, which is of course SS-symmetric. The capacity C_1 is given by :

$$C_1 = 1 - h(p/2) \qquad \text{bits/memory cell} \qquad (2.4)$$

INTERLUDE

By way of clarification we give in this short interlude an answer to the question of the way we have to use the concept of the superchannel in the actual storage process.
Let us consider e.g. the derived superchannel in case n = 1, with two capacity-achieving superinputs $\underline{X}_1$ = *(0,0,1) and* $\underline{X}_2$ = *(1,0,1).*

First of all we associate with $\underline{X}_1$ *the symbol* 0 *and with* $\underline{X}_2$ *the symbol* 1. *Suppose the symbol to be stored is a* 0. *Then, depending on the defect-pattern of the cell to be written, the encoder stores a* 0 (t = 1,2) *or a* 1 (t = 3). *Only in case* t = 3 *an error occurs. An error-correcting code will take care of this error.*
Obviously, the superinputs $\underline{X}$ *fully describes the storage strategy of the encoder.*

END OF INTERLUDE

On searching for constructive error-correcting codes for K', we may consider K' as a BSC with noiseless feedback.
Knowing the state t of the BDC K and the codeword x_t (*figure 2.1*) the encoder can unambiguously reconstruct the output y.
Therefore, the channel depicted in *figure 2.2* is equivalent to the BDC with state information to the encoder only (*figure 2.1*).

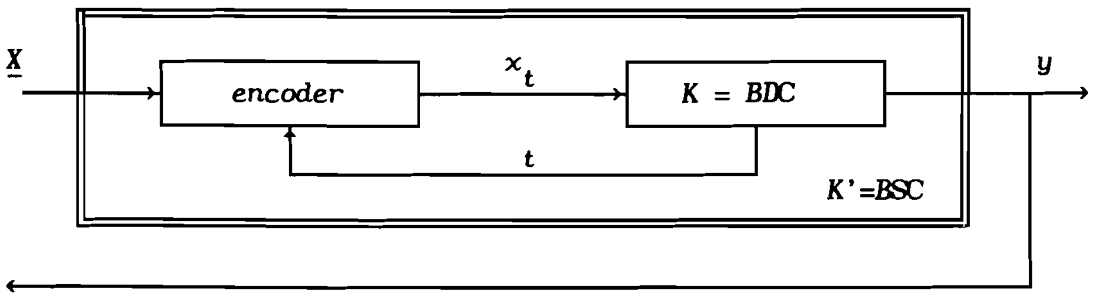


Figure 2.2 : Equivalent BSC with noiseless feedback.

Although a noiseless feedback link facilitates the coding process, it is established by Shannon [SHA56], that the capacity of the forward channel is not increased.

From this we observe, that instead of existence results concerning good codes in case of 'unknown defects' (Section 1.2.1), one can now use constructive feedback strategies to obtain reliable storage at efficiencies up to C_1 bits per memory cell. In Chapter 4 we describe a class of feedback codes, that achieves C_1 for some p.

From Kusnetsov and Tsybakov [KUS74] we know, that there exist codes that achieve $\bar{C}$ bits/memory cell.
To achieve $\bar{C}$, we have to consider in the coding process n memory cells at the same time, instead of coding cell by cell. So, we have to anticipate into the future. This idea is the second step in Schalkwijk's approach. We denote n parallel BDC as BDC^n. In the next section we apply the '*side-information" theorem* to the BDC^n.

2.3 MULTIPLE USE OF THE BINARY DEFECT CHANNEL (BDC^n)

Instead of considering only one memory cell (BDC^1), we take in each coding step n cells (BDC^n, n = 2,3,4,...) into account.
The obvious generalisations in the eq.'s (2.5-a), (2.5-b) of the eq.'s (1.2-a) and (1.2-b) completely describes the n^{th} power of the BDC :

$$K = \{(K_t, g_t) \mid t \in S)\} \tag{2.5-a}$$

$$K_t = (A^n, [P_{t\underline{x}}(\underline{y})], B^n) \tag{2.5-b}$$

2.3 *Multiple use of the Binary Defect Channel* (BDC^n)

The input of the BDC^n is a binary vector $\underline{x} = (x_1, x_2, \ldots, x_n)$, $\underline{x} \in A^n$ and $|A^n| = a^n = 2^n$.

Likewise, the output $\underline{y}$ is a binary vector $\underline{y} = (y_1, y_2, \ldots, y_n)$, $\underline{y} \in B^n$ and $|B^n| = b^n = 2^n$. The total number $|S| = h$ of defect-patterns of length n is equal to 3^n.

The component-channel probabilities g_t and the transition probabilities $P_{t\underline{x}}(\underline{y})$ are respectively described by :

$$g_t = \prod_{i=1}^{n} g_{t_i} \tag{2.6}$$

and

$$P_{t\underline{x}}(\underline{y}) = \prod_{i=1}^{n} P_{t_i x_i}(y_i) \tag{2.7}$$

The index t_i $(1 \leq i \leq n)$ corresponds with the state of the i^{th} memory cell.

TABLE 2.1 illustrates some of the parameters in case $K = BDC^2$.

Now, we apply the obvious generalisation of the '*side-information*' *theorem 2.1* for FSC's with binary vector inputs and outputs to BDC^n.

The derived channel K' has 2^{nh} different superinputs, denoted by $\underline{\underline{X}}_i = (\underline{x}_{i1}, \underline{x}_{i2}, \ldots, \underline{x}_{ih})$, with $\underline{x}_{it} \in \{0,1\}^n$, $t \in S$, $1 \leq i \leq 2^{nh}$ and 2^n outputs $\underline{y}$. The transition probabilities are given by $r_{\underline{\underline{X}}}(\underline{y}) = \sum_{t \in S} g_t \cdot P_{t\underline{x}_t}(\underline{y})$. In case $n = 2$ the number of superinputs is equal to $2^{18} = 262144$, as distinct from only 4 outputs.

TABLE 2.1 : Channel parameters of BDC^2.

$A^2 = \{(0,0),(0,1),(1,0),(1,1)\}$, $a^2 = 4$

$B^2 = \{(0,0),(0,1),(1,0),(1,1)\}$, $b^2 = 4$

$S = \{1,2,\ldots,h\}$, $h = 9$

component channel probabilities

index	defects	probability
t = 1	C C	$g_1 = q^2$
t = 2	C 0	$g_2 = (pq)/2$
t = 3	C 1	$g_3 = (pq)/2$
t = 4	0 C	$g_4 = (pq)/2$
t = 5	1 C	$g_5 = (pq)/2$
t = 6	0 0	$g_6 = p^2/4$
t = 7	0 1	$g_7 = p^2/4$
t = 8	1 0	$g_8 = p^2/4$
t = 9	1 1	$g_9 = p^2/4$

e.g. transition probability $P_{3\underline{x}}(\underline{y})$

$\underline{x}$ \ $\underline{y}$	0 0	0 1	1 0	1 1
0 0	0	1	0	0
0 1	0	1	0	0
1 0	0	0	0	1
1 1	0	0	0	1

Obviously, the Arimoto-Blahut algorithm is impractical for $n \geq 2$.
The problem of finding 2^n capacity-achieving superinputs (*see theorem* 2.2) out of 2^{nh} will be tackled in the next chapter. Similar to the last section each of the 2^n superinputs $\underline{\underline{X}}$ describes a storage strategy for one of the possible binary sequences of length n. In Section 1.2 we indexed these sequences with the letter u.
Moreover, we will see that the maximum storage efficiency C_n in a memory increases due to our anticipation into the future ($n = 2,3,4,\ldots$).

Furthermore, the derived channel K' can be considered a DMC with noiseless feedback. The argument is similar to the BDC-case in the last section.
In Chapter 4 we describe a class of constructive feedback codes. Their achievable rates approaches C_n ($n = 2,3,4,\ldots$) tight for some p.

3 ON THE CAPACITY OF BDC^n

3.1 INTRODUCTION

In Chapter 2 we showed the significance of finding 2^n capacity-achieving superinputs out of 2^{nh}. They not only describe the storage strategy for every input sequence, but also facilitates the computation of the capacity.

As the state-probability distribution g_t (eq. (2.6)) and the state component channels K_t with transition probabilities $P_{t\underline{x}}(\underline{y})$ (eq. (2.7)) are symmetrical, it is established that given a capacity-achieving superinput $\underline{\underline{X}}_i$, 2^n-1 different capacity-achieving superinputs can be obtained, by simply using this symmetry.

The transition probability leading away from an element in this set of capacity-achieving superinputs depend on the n-tuple resulting from the bit-by-bit modulo-2 addition of the symbol associated with this superinput and the output symbol. Therefore, the resulting channel K' is WS-symmetric. This implies, that we only have to determine one capacity-achieving $\underline{\underline{X}}_i$, e.g. the superinput $\underline{\underline{X}}_1$ associated with symbol (0,0,...,0) of length n.

From e.g. Gallager [GAL68, p. 94 theorem 4.5.2] we know, that on achieving capacity we give each of the 2^n capacity-achieving $\underline{\underline{X}}_i$ equal input probability and all other superinputs probability 0. In consequence the resulting probability distribution over all outputs $\underline{y} \in B^n$ is uniformly distributed as well.

Now, by applying Kuhn-Tucker's theorem [GAL68, p. 91 theorem 4.5.1] to the derived channel K', the capacity is given by:

$$C_n = \max_{1 \leq i \leq 2^{nh}} R_n(\underline{\underline{X}}_i) \qquad \text{bits/cell} \tag{3.1}$$

and

$$R_n(\underline{\underline{X}}_i) = 1/n. \sum_{\underline{y} \in B^n} r_{\underline{\underline{X}}_i}(\underline{y}).{}^2\log(2^n.r_{\underline{\underline{X}}_i}(\underline{y})) \tag{3.2}$$

3.2 FIXED NUMBER OF DEFECTS

3.2.1 NOTATIONAL CONVENTIONS AND DEFINITIONS

In a first attempt to find a constructive method, that results in a capacity-achieving superinput, we consider a less general BDC^n model. Instead of the so-called, probabilistic channel model in Section 2.3, this model assumes a fixed number of defects f in n cells to be considered. Obviously, the number of defect-patterns $h = 2^f.\binom{n}{f}$. The state probabilities of the individual cells are indepently and identically distributed, in conformity with the probabilistic model.
Hence, the component-channel probabilities g_t $(1 \leq t \leq h)$ are uniformly distributed, i.e. $g_t \triangleq g = 1/h$. It is readily verified that in this model the derived channel is also WS-symmetric.
Before going into details concerning capacity-achieving superinputs, we give some preliminary notational conventions and definitions.

* The set of all binary sequences of lenght n is denoted by V_n.

* The weight $w(\underline{d})$ of a defect-pattern $\underline{d}$ is equal to the number of defective positions in $\underline{d}$.

* By D_f we denote the set of defect-patterns, for which holds $w(\underline{d}) = f$.

* Consider the defect-pattern $\underline{d}$ as a number of n digits. Each digit can be either 1, 2 or 3, according to *definition 1.1*. From now on, we assume the different defect-patterns, indexed by t, to be arranged in order of increasing magnitude (see e.g. *TABLE 2.1*).

* *Definition 3.1* :

A binary digit masks (is compatible with) a defect, iff its value agrees with the value of the defect.

* *Definition 3.2* :

A binary vector $\underline{x} = (x_1, x_2, \ldots, x_n)$ *masks (is compatible with) a defect-pattern* $\underline{d}$, *iff for each defective position* i *in* $\underline{d}$, x_i *masks the defect.*

* *Definition 3.3* :

A set X *of binary vectors of length* n *covers all defect-patterns* $\underline{d} \in D_f$ *(is f-defect compatible), iff for every* $\underline{d} \in D_f$ *there is an* $\underline{x} \in X$, *that masks the defect-pattern* $\underline{d}$.

* *Theorem* 3.1 :

If a set of binary vectors of length n, covers all $\underline{d}\in D_f$, *it covers all* $\underline{d}\in D_i$ $(0 \le i \le f)$.

Proof: trivial |

* $r_{\underline{\underline{X}}}(\underline{y})$ denotes the transition probability from superinput $\underline{\underline{X}}$ to output sequence $\underline{y}$.

* $S_f(\underline{\underline{X}})$ is the set of outputs $\underline{y}\in V_n$ resulting from superinput $\underline{\underline{X}}$, for which $r_{\underline{\underline{X}}}(\underline{y}) > 0$. The cardinality $|S_f(\underline{\underline{X}})| \triangleq M < 2^n$. Notice, that $S_f(\underline{\underline{X}})$ is f-defect compatible.

* We introduce the variable $a_f(\underline{y})\in\mathbb{N}$, $\underline{y}\in S_f(\underline{\underline{X}})$, as the number of elements $\underline{x}_t$ in $\underline{\underline{X}}$, that lead to outputword $\underline{y}\in S_f(\underline{\underline{X}})$. Hence,

$$r_{\underline{\underline{X}}}(\underline{y}) = a_f(\underline{y})\cdot g \tag{3.3}$$

* Without loss of generality we only consider superinputs $\underline{\underline{X}}$ for which the elements $\underline{x}_t$ masks the corresponding defect-pattern.
The fact is, that the set $S_f(\underline{\underline{X}})$ and the probabilities $r_{\underline{\underline{X}}}(\underline{y})$ are not changed if we alter $\underline{\underline{X}}$ on corresponding defective positions.
Hence, the total number of superinputs, to be considered, is restricted to $2^{(n-f).h}$.

3.2.2 ON THE CAPACITY OF BDC^n

From the last section we rewrite eq. (3.2) in :

$$C_n(f) \triangleq \max_{1 \leq i \leq 2^{nh}} R_n(\underline{\underline{X}}_i) \qquad \text{bits/cell} \qquad (3.4)$$

and

$$R_n(\underline{\underline{X}}_i) = 1 - 1/n.{}^2\log h + 1/(n.h). \sum_{\underline{y} \in S_f(\underline{\underline{X}}_i)} a_f(\underline{y}).{}^2\log a_f(\underline{y}) \qquad (3.5)$$

Obviously from eq. (3.1), a superinput $\underline{\underline{X}}$ is capacity achieving, iff it maximizes the sum-term in eq. (3.4). Noticing this, the problem of finding a capacity-achieving superinput is equivalent to a well-known problem in coding theory. The proof of this equivalence uses the following lemma.

Lemma 3.1 :

$$a.{}^2log\ a + b.{}^2log\ b < (a - \Delta).{}^2log\ (a - \Delta) + (b + \Delta).{}^2log\ (b + \Delta)$$

$$\forall\ a,b \in \mathbb{R}^+ \quad a \leq b$$

$$\forall\ \Delta \in \mathbb{R}^+ \quad 0 < \Delta \leq a$$

Proof: Consider the function f(x) :

$$f(x) = (\gamma + x).{}^2\log\ (\gamma + x) + (\gamma - x).{}^2\log\ (\gamma - x)$$

$$\forall\ x,\gamma \quad 0 \leq x < \gamma$$

As the derivative f'(x) varies proportional to ln ($\frac{\gamma + x}{\gamma - x}$), which is a monotonous function of x, f(x) is an increasing function. So,

$$f(x_1) < f(x_2) \qquad \text{for } 0 < x_1 < x_2 < \gamma \qquad (*)$$

Notice that we can always find a triple x_1, x_2 and γ, obeying the conditions mentioned above and so that :

$$a = \gamma - x_1$$
$$b = \gamma + x_1$$
$$\Delta = x_2 - x_1$$

Substituting in (*) proves the lemma. |

Now, we are ready for the main theorem in this section.

Theorem 3.2 :

The capacity $C_n(f)$ of a binary memory of length n and with f defects, is equal to the upperbound $\bar{C} = 1-f/n$, iff there exists a superinput $\underline{\underline{X}}$ for which holds:

a) $M = 2^f$, $|S_f(\underline{\underline{X}})| = M$

b) $d_H(\underline{y}_i,\underline{y}_j) \geq n - f + 1$, $\forall\ i \neq j$ and $\underline{y}_i,\underline{y}_j \in S_f(\underline{\underline{X}})$

Proof: Suppose there exists a superinput $\underline{\underline{X}}$ for which conditions a) and b) hold.

First of all, we subdivide the set of defect-patterns D_f into $\binom{n}{f}$ different subsets, such that the defect-patterns in each subset have their f defects on the same positions. These subsets contain 2^f elements. Notice further, that each outputword $\underline{y} \in S_f(\underline{\underline{X}})$ masks exactly one defect-pattern in such a subset. From this it follows that

$$0 < a_f(\underline{y}) \leq \binom{n}{f} \qquad \forall\ \underline{y} \in S_f(\underline{\underline{X}}) \tag{3.6}$$

As condition b) holds each defect-pattern in D_f can only result in one outputword $\underline{y} \in S_f(\underline{\underline{X}})$. So, in order to cover D_f, $M \geq 2^f$. Since $M = 2^f$ (condition a)) equality holds in eq. (3.6). Substitution in the eq.'s (3.4) and (3.5) gives :

$$C_n(f) = \overline{C} \tag{3.7}$$

It remains to show that superinput $\underline{\underline{X}}$, with uniform distribution over $a_f(\underline{y})$ ($\underline{y} \in S_f(\underline{\underline{X}})$), is the only possible superinput for which eq. (3.7) holds. This is immediately clear from *lemma 3.1*. |

Searching for superinputs $\underline{\underline{X}}$ satisfying the conditions a) and b) in *theorem* 3.2 is equivalent to looking for binary $(n,2^f,n-f+1)$ - codes, i.e. a code of length n consisting of 2^f codewords, which have mutual Hamming-distance at least $n - f + 1$. In the cases where such a code exists for particular values of n and f, it is easy to construct the original capacity-achieving superinput $\underline{\underline{X}}$.

In case f = {1,n-1,n} there exist, so-called, trivial Maximum - Distance - Separable (MDS) codes (see [MAC77, pp. 318]).

Take, e.g. n = 3 and f = 2 :

$S_2(\underline{X})$ = {000,101,110,011} ([3,2,2] - code)

index	*defects*	*superinput X*
t = 1	C 0 0	0 0 0
t = 2	C 0 1	1 0 1
t = 3	C 1 0	1 1 0
t = 4	C 1 1	0 1 1
t = 5	0 C 0	0 0 0
t = 6	0 C 1	0 1 1
t = 7	1 C 0	1 1 0
t = 8	1 C 1	1 0 1
t = 9	0 0 C	0 0 0
t = 10	0 1 C	0 1 1
t = 11	1 0 C	1 0 1
t = 12	1 1 C	1 1 0

For $2 \leq f \leq n - 2$ we have *theorem* 3.3.

Theorem 3.3 :

Let $f \triangleq n - i$, $i \geq 2$ *and* $n \geq i + 2$, *then binary* $(n, 2^f, n-f+1)$*-codes do not exist.*

Proof: In case $i + 2 \leq n < 2(i + 1)$ and $i \geq 2$ the theorem follows immediately from the Plotkin-bound [LIN82, p.58 eq. (5.24)]. In case $n \geq 2(i + 1)$ and $i \geq 2$, the Hamming-bound proves the non-existence of binary $(n, 2^f, n-f+1)$ - codes [LIN82, p.59 eq. (5.27)]. Some properties of binomial coefficients are necessary to complete this proof.

Here a more elegant proof, due to van Overveld, will be discussed in detail.

Consider a binary $(n, 2^f, d)$ - code C, where d is the Hamming-distance of C. Then, there exist 2^{f-j} binary sequences of length $n-j$ and mutual Hamming-distance d for all $j < f$.

A possibility to obtain these sequences is by taking in j consecutive steps half of the number of sequences, which have the same value in their last position. After each step this last bit is deleted. Taking $j = f - 1$, 2 sequences of length $n-f+1$ and Hamming-distance d remains.

Now suppose, C is a binary $(n, 2^f, d)$ - code with $d = n - f + 1$ and $3 \leq d \leq n-2$. Then, by applying the shortening procedure, outlined above, we obtain two sequences of length $n - f + 1$ and Hamming-distance $d = n - f + 1$.

W.l.o.g. these sequences are :

$$\begin{array}{c} 1\,1\,.\,.\,.\,1 \\ 0\,0\,.\,.\,.\,0 \\ \longleftarrow d \longrightarrow \end{array}$$

Again w.l.o.g. the preceding step contains the sequences $\underline{r}_1$ and $\underline{r}_2$:

$$\begin{aligned} \underline{r}_1 &= 1\ 1\ .\ .\ .\ 1\ 0 \\ \underline{r}_2 &= 0\ 0\ .\ .\ .\ 0\ 0 \\ &\longleftarrow d \longrightarrow \end{aligned}$$

and two other sequences $\underline{s}_1$ and $\underline{s}_2$ of length d + 1.
Both $\underline{s}_1$ and $\underline{s}_2$ must satisfy:

a) $d_H(\underline{s}_i, \underline{r}_1) \geq d \qquad , \ i \in \{0,1\}$

b) $d_H(\underline{s}_i, \underline{r}_2) \geq d \qquad , \ i \in \{0,1\}$

If we confine ourselves to $\underline{s}_1$, then from a) this sequence must contain at least d - 1 zeros and from b) at least d - 1 ones in the first d positions. As there are only d + 1 positions available, $\underline{s}_1$ and therefore code C cannot exist.

In case d = 1 and d = 2, on the contrary, this reconstruction is possible $(2(d - 1) \leq d)$, which is in conformity with the existence of the trivial MDS-codes. |

From the *theorems* 3.2 and 3.3 we conclude, that

$$C_n(f) < \bar{C} \qquad , \ 2 \leq f \leq n - 2 \tag{3.8}$$

Hence, in order to determine $C_n(f)$ in case $2 \leq f \leq n - 2$, we have to search for f-defect compatible sets $S_f(\underline{\underline{X}})$, which have a distribution $a_f(\underline{y})$, $\underline{y} \in S_f(\underline{\underline{X}})$, that maximizes the sum-term in eq. (3.5).

In case f = 2, we present some computer results. They are tabulated in *TABLE* 3.2 and depicted in *figure 3.1*. In Appendix A the capacity-achieving sets $S_2(\underline{\underline{X}})$ and $a_2(\underline{y})$-distributions are given.
The search algorithm is based on the next results of van Pul [PUL87].

Theorem 3.4 :

Let $M(n,f)$ be the minimum number of elements of length n ($n \geq 2$) in a f-defect compatible set, then $M(n,2)$ is unambiguously determined by the following binomial inequalities :

$$\binom{M(n,2)-2}{\left\lceil \frac{M(n,2)-1}{2} \right\rceil} < n \leq \binom{M(n,2)-1}{\left\lceil \frac{M(n,2)}{2} \right\rceil} \qquad (3.8)$$

Proof: A proof is ommited here. We refer to [PUL87].

TABLE 3.1 shows some values of M(n,2).

TABLE 3.1: $M(n,2)$ *for* $2 \leq n \leq 35$

$M(n,2)$	
4	$n = 2,3$
5	$n = 4$
6	$5 \leq n \leq 10$
7	$11 \leq n \leq 15$
8	$16 \leq n \leq 35$

Starting from these values of $M(n,2)$ the $a_2(\underline{y})$-distribution, that maximizes the sum-term in eq. (3.5) and for which a 2-defect compatible set $S_2(\underline{\underline{X}})$ exists, is determined.
Lemma 3.1 proves to be useful in limiting the search time.

3.2.3 COMPARISON WITH OTHER RESULTS

In former sections we showed, that determining $C_n(f)$ for $2 \leq f \leq n - 2$ is equivalent to searching for an f-defect compatible set $S_f(\underline{\underline{X}})$, which has an associated $a_f(\underline{y})$-distribution, that maximizes the rate function in eq. (3.5).

In the additive coding scheme of Kusnetsov and Tsybakov we have a related problem.
Here, the rate of an additive code is determined by the number of elements M_k in a f-defect compatible set. As each element must contain an identification part, the efficiency of a code is maximized when M_k is minimized. The rate is given by:

$$R_k(f) = 1 - \left\lceil \frac{{}^2\log M_k}{n} \right\rceil \qquad \text{bits/cell} \qquad (3.9)$$

Since 1974 many people worked on finding efficient f-defect compatible sets in case $2 \leq f \leq n - 2$ ([KUS74], [BEL77], [LOS78], [BUS84], [PUL85], [VIN86], [PUL87]). The best achievable rates, denoted by R_{kus}, for the additive coding scheme in case $f = 2$, are tabulated in *TABLE* 3.2 and depicted in *figure 3.1*.

An upperbound ($f = 2$) results from *theorem* 3.4 and the following two assumptions:

1) On the average each element in a 2-defect compatible set is used equally likely.
   Referring to [KUS74], this is realised by giving the messages equal probability.

2) The identification parts of the elements form an Huffman code (prefix) for M_k equally likely elements.

From *theorem* 3.4 we determine the minimum number of elements M_k of a 2-defect compatible set. Let $\bar{n}_H$ be the average number of bits in the Huffman code, then the upperbound is determined by (see *TABLE* 3.2 and *figure 3.1*):

$$R_{kus}(\text{up}) \triangleq 1 - \frac{\bar{n}_H}{n} \qquad \text{bits/cell} \qquad (3.10)$$

for each $n \geq 4$

TABLE 3.2: R_{kus}, $R_{kus}(up)$, $C_n(2)$ *and* $\bar{C}_n(2)$ *for* $2 \leq n \leq 10$.

n	R_{kus}	$R_{kus}(up)$	$C_n(2)$	$\bar{C}_n(2)$
2	0	0	0	0
3	0.33333	0.33333	0.33333	0.33333
4	0.25	0.4	0.43068	0.5
5	0.4	0.46667	0.52781	0.6
6	0.5	0.55555	0.60591	0.66667
7	0.57143	0.61905	0.65227	0.71429
8	0.625	0.66667		0.75
9	0.66667	0.70370		0.77778
10	0.7	0.73333		0.8

From Kusnetsov and Tsybakov's lowerbound for R_{kus} it is established, that by increasing n the functions R_{kus}, $R_{kus}(up)$ and $C_n(2)$ approach the upperbound $\bar{C}_n(2)$ arbitrary tight.

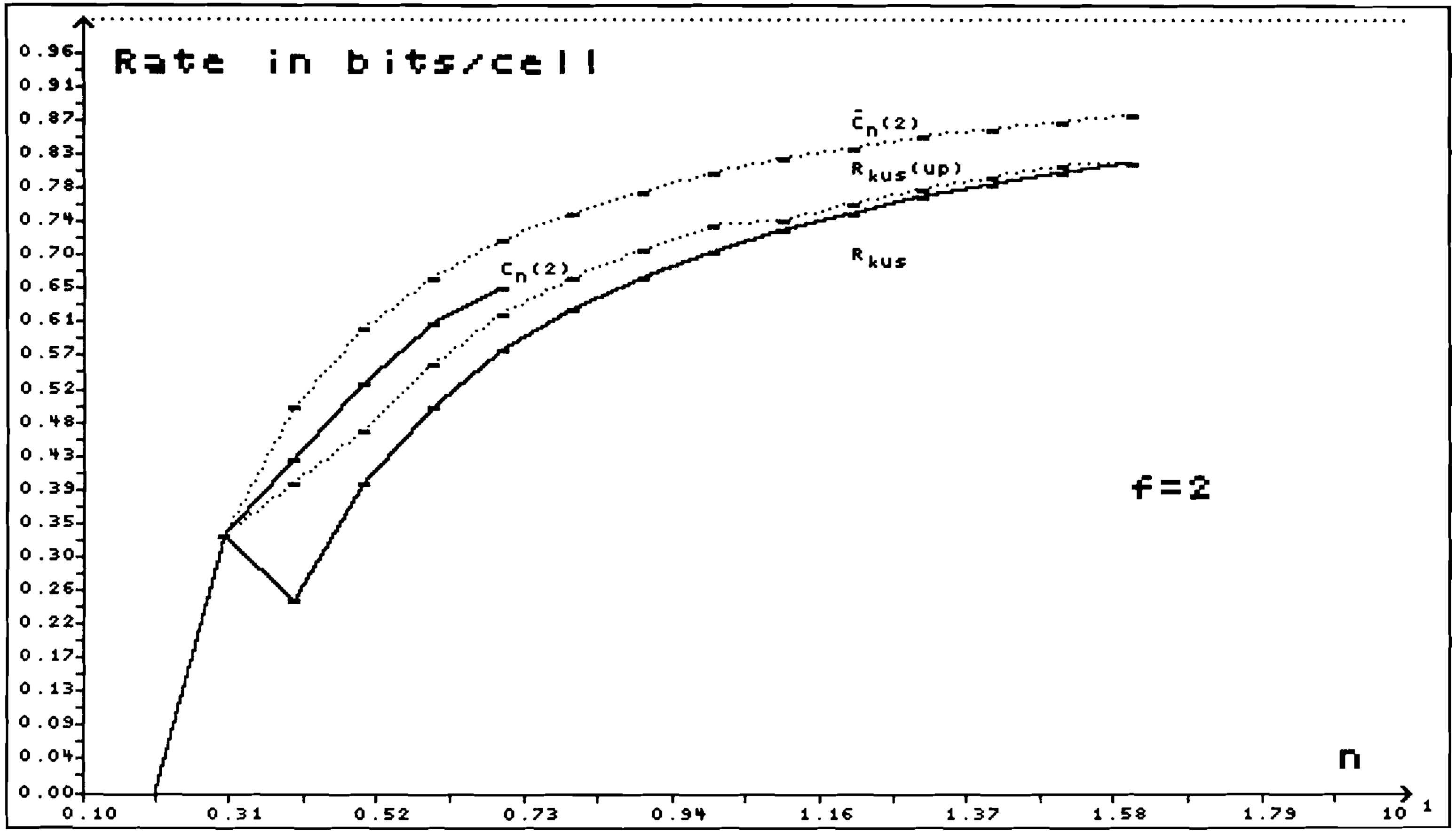


Figure 3.1: R_{kus}, $R_{kus}(up)$, $C_n(2)$ and $\bar{C}_n(2)$ as a function of n.

3.3 PROBABILISTIC MODEL

3.3.1 NOTATIONAL CONVENTIONS AND DEFINITIONS

We now return to the more general probabilistic BDC^n model, stated in the Sections 1.1 and 2.3. The total set of defect-patterns in this model can be subdivided into n+1 sets: D_f, $f \in \{0,1,\dots, n\}$, where n is the number of cells to be considered. The corresponding probability of occurrence for every defect-pattern in the subset D_f, $f \in \{0,1,\dots, n\}$, is given by

$$g(f) = (1 - p)^{n-f}.(p/2)^f \tag{3.11}$$

Noticing the order convention (Section 3.2.1) of the defect-pattern, the relationship between g_t, $1 \leq t \leq 3^h$, in eq. (2.6) and $g(f)$, $0 \leq f \leq n$, is shown in eq. (3.12).

$$\begin{aligned} f = 0 &\longrightarrow g(f) = g_1 \\ 1 \leq f \leq n &\longrightarrow g(f) = g_t \quad \text{for} \sum_{i=0}^{f-1} 2^i.\binom{n}{i} < t \leq \sum_{i=0}^{f} 2^i.\binom{n}{i} \end{aligned} \tag{3.12}$$

* $S(\underline{\underline{X}})$ is the set of outputs $\underline{y} \in V_n$, resulting from superinput $\underline{\underline{X}}$ for which $r_{\underline{\underline{X}}}(\underline{y}) > 0$. In fact $S(\underline{\underline{X}}) = V_n$, because of the defect-patterns in subset D_n. The cardinality $|S(\underline{\underline{X}})| = 2^n$.

We assume the outputs $\underline{y}_j \in S(\underline{\underline{X}})$ to be arranged in order of decreasing $r_{\underline{\underline{X}}}(\underline{y}_j)$. So $r_{\underline{\underline{X}}}(\underline{y}_1) \geq r_{\underline{\underline{X}}}(\underline{y}_2) \geq \ldots \geq r_{\underline{\underline{X}}}(\underline{y}_{2^n})$.

* Instead of attaching only one parameter $a_f(\underline{y})$ to every $\underline{y} \in S(\underline{\underline{X}})$, we define n + 1 parameters $a_f(\underline{y})$, $0 \leq f \leq n$. The variable $a_f(\underline{y})$ denotes the number of elements $\underline{x}_t$, $\sum_{i=0}^{f-1} 2^i.\binom{n}{i} < t \leq \sum_{i=0}^{f} 2^i.\binom{n}{i}$ and $f > 0$, in $\underline{\underline{X}}$, that leads to outputword $\underline{y}$. In case f = 0, only $\underline{x}_1$ may lead to $\underline{y}$. Hence,

$$r_{\underline{\underline{X}}}(\underline{y}) = \sum_{f=0}^{n} a_f(\underline{y}).g(f) \tag{3.13}$$

with

$$0 \leq a_f(\underline{y}) \leq \binom{n}{f} \tag{3.14}$$

and

$$\sum_{\underline{y} \in S(\underline{\underline{X}})} a_f(\underline{y}) = 2^f.\binom{n}{f} \tag{3.15}$$

As an illustration we give in *TABLE* 3.3 an overview of an $a_f(\underline{y})$-distribution in case n = 3.

TABLE 3.3.: *An* $a_f(\underline{y})$*-distribution for* $n = 3$.

	S(X)	f = 0	f = 1	f = 2	f = 3
	$\underline{y}_1$	1	3	3	1
	$\underline{y}_2$		3	3	1
$a_f(\underline{y})$	$\underline{y}_3$			3	1
	$\underline{y}_4$			3	1
	$\underline{y}_5$				1
	$\underline{y}_6$				1
	$\underline{y}_7$				1
	$\underline{y}_8$				1

3.3.2 ON THE CAPACITY OF BDC^n

In general the determination of the set of outputs $S(\underline{\underline{X}})$ with optimal $a_f(\underline{y})$-distribution, $0 \leq f \leq n$ and $\underline{y} \in S(\underline{\underline{X}})$ is a tedious task. We use the word optimal here in a sense, that it maximizes the rate function in eq. (3.5). Written in another way this function becomes:

$$R_n(\underline{\underline{X}}_i) = 1 + 1/n. \sum_{j=1}^{2^n} r_{\underline{\underline{X}}_i}(\underline{y}_j).{}^2\log r_{\underline{\underline{X}}_i}(\underline{y}_j) \tag{3.16}$$

From the "fixed-number-of-defects" case, it is not difficult to give the optimal $a_f(\underline{y})$-distribution. The question concerning the existence of a superinput $\underline{\underline{X}}$, realising this $a_f(\underline{y})$-distibution, is harder to answer.

First of all, we give the optimal $a_f(\underline{y})$-distribution, apart from a possible realisation. It gives an upperbound for C_n. To prove the optimality we use *lemma 3.1* again.

Theorem 3.5:

Let $a_f^*(\underline{y}^*)$ be distributed in such a way, that it maximizes the rate function in eq. (3.16), then:

$$a_f^*(\underline{y}_j^*) = \begin{cases} \binom{n}{f}, & 0 \le f \le n \text{ and } 1 \le j \le 2^f \\ 0, & 0 \le f < n \text{ and } 2^f < j \le 2^n \end{cases}$$

Proof: From eq. (3.14) we know, that the maximimum number of defect-patterns in the set D_f, leading to one particular output $\underline{y}$, is equal to $\binom{n}{f}$.

Furthermore, if $a_f^*(\underline{y}_j^*) = \binom{n}{f}$ then $a_f^*(\underline{y}_i^*) = \binom{n}{f}$ for all $1 \le i \le j$.

It follows that any other $a_f(\underline{y})$-distribution contains at least one output $\underline{y}_j$ $(1 \le j \le 2^f)$ with $a_f(\underline{y}_j) < \binom{n}{f}$ and at least one output $\underline{y}_i$ $(2^f < i \le 2^n)$ with $a_f(\underline{y}_i) > 0$.

In consequence the corresponding transition probabilities, according to eq. (3.13), are : $r_{\underline{\underline{X}}}(\underline{y}_j) < r_{\underline{\underline{X}}^*}(\underline{y}_j^*)$ and $r_{\underline{\underline{X}}}(\underline{y}_i) > r_{\underline{\underline{X}}^*}(\underline{y}_i^*)$.

From *lemma* 3.1 it is clear that the netto contribution of any of these $a_f(\underline{y})$-distributions to the rate function in eq. (3.16) is less than the contribution of the $a_f^*(\underline{y}^*)$-distribution. |

Theorem 3.5 states the optimality of the $a_f^*(\underline{y}^*)$-distribution. Substituting in eq. (3.16), it gives the upperbound $\overline{C}_n$, i.e. the existence of a corresponding superinput $\underline{\underline{X}}^*$ has not been proved. The upperbound $\overline{C}_n$ is tabulated for $1 \leq n \leq 10$ in *TABLE* 3.6.

3.3.3 RESULTS

For n = 1 (see eq. (2.3)) and n = 2 it can be readily verified, that the upperbound $\overline{C}_n$ is achievable. In Appendix B, we give the capacity-achieving superinputs and their transition probabilities in case n = 1 and n = 2 respectively.

Using the eq.'s (3.1) and (3.2), we find the capacities C_1 and C_2 (see also [SCH86]) :

$$C_1 = 1 - h(p/2) \tag{3.17}$$

$$C_2 = 1/2.(2 - \{h(1 - \frac{p^2}{2}) + (1 - \frac{p^2}{2}).h\left[\frac{(1 - \frac{p}{2})^2}{(1 - \frac{p^2}{2})}\right] + \frac{p^2}{2}\}) \tag{3.18}$$

In *TABLE* 3.7 C_1 and C_2 are tabulated as a function of p. As we expected, the inequality $C_2 > C_1$ holds for every $p \in \langle 0,1 \rangle$, due to the anticipation into the future.

For $n \geq 3$ the upperbound $\overline{C}_n$ is not achievable. Computer analysis show, that in case n = 3, there are two sets of superinputs with different $a_f(\underline{y})$-distributions, which are capacity achieving in a limited interval of p.

As an illustration, we give of both sets the outputset $S(\underline{\underline{X}}_1)$ and the $a_f(\underline{y})$-distribution of the superinput $\underline{\underline{X}}_1$, which is associated with the symbol 000..

TABLE 3.4: Capacity-achieving $a_f(\underline{y})$-distribution for $0 \leq p \leq 0.4653$.

$S(\underline{\underline{X}}_1)$	$f = 0$	$f = 1$	$f = 2$	$f = 3$
000	1	3	3	1
111		3	3	1
001			2	1
010			2	1
100			2	1
011				1
101				1
110				1

TABLE 3.5: Capacity-achieving $a_f(\underline{y})$-distribution for $0.4653 < p \leq 1$.

$S(\underline{\underline{X}}_1)$	$f = 0$	$f = 1$	$f = 2$	$f = 3$
000	1	3	3	1
011		2	3	1
101		1	3	1
110			3	1
001				1
010				1
100				1
111				1

The capacity C_3 is tabulated in *TABLE* 3.7.

From Section 3.2 and Appendix A we know that for $n \geq 4$ and $f = 2$, there exists no superinput realising the optimal distribution:

$$a_2^*(\underline{y}_j^*) = \binom{n}{2} \qquad \text{for every } 1 \leq j \leq 2^2.$$

This indeed implies, that $\bar{C}_n > C_n$ for $n \geq 3$.

The remaining part of *TABLE* 3.7 contains the rates R_n for $4 \leq n \leq 10$. These rates results from superinputs $\underline{\underline{X}}'$ with transition probabilities obeying:

$$r_{\underline{\underline{X}}'}(\underline{y}'_j) \geq r_{\underline{\underline{X}}_i}(\underline{y}_j) \qquad \text{for all } 1 \leq j \leq 2^n, \quad 1 \leq i \leq 2^{nh} \quad \text{and } \underline{\underline{X}}' \neq \underline{\underline{X}}_i \tag{3.19}$$

The rate R_n is equal to C_n in case n = 1 and n = 2. Therefore, their $a'_f(\underline{y}')$-distributions are given according to *theorem* 3.5.
For $0 \leq p \leq 0.4653$ $R_3 = C_3$ and their distributions are identical. In case the defect fraction p = 0.5 *figure* 3.2 gives the upperbound $\bar{C}_n$ and the achievable rates R_n as a function of n.
Notice, that the gradient of the R_n-curve drops off very fast for $n \geq 5$ and becomes even negative in case n = 10. To judge the applicability of eq. (3.19) as a good criterion to obtain superinputs, that achieve rates approaching the capacity C_n, we need more results.

To give a better idea of their mutual displacements C_1, C_2, C_3, $\bar{C}_{16}$ and $\bar{C}$ are depicted in *figure* 3.3.
The upperbound $\bar{C}_n$ can be tightened by replacing the optimal $a^*_2(\underline{y}^*_j)$ for $1 \leq j \leq 2^n$ by the optimal distributions found in Section 3.2 and tabulated in Appendix A. The resulting curve $\bar{C}^t_n$ is also given in *figure* 3.2.

TABLE 3.6: $\overline{C}_1, \overline{C}_2, \ldots, \overline{C}_{10}$ *as a function of p.*

p	$\overline{C}_1$	$\overline{C}_2$	$\overline{C}_3$	$\overline{C}_4$	$\overline{C}_5$
0.1	0.71360	0.75276	0.77473	0.78940	0.80014
0.2	0.53100	0.59315	0.62776	0.65033	0.66647
0.3	0.39016	0.46276	0.50363	0.53017	0.54903
0.4	0.27807	0.35110	0.39324	0.42097	0.44079
0.5	0.18872	0.25434	0.29373	0.32034	0.33970
0.6	0.11871	0.17129	0.20463	0.22804	0.24554
0.7	0.06593	0.10225	0.12699	0.14528	0.15946
0.8	0.02905	0.04866	0.06333	0.07490	0.08433
0.9	0.00723	0.01316	0.01818	0.02251	0.02631

p	$\overline{C}_6$	$\overline{C}_7$	$\overline{C}_8$	$\overline{C}_9$	$\overline{C}_{10}$
0.1	0.80845	0.81514	0.82067	0.82534	0.82935
0.2	0.67871	0.68839	0.69629	0.70288	0.70849
0.3	0.56325	0.57444	0.58353	0.59110	0.59753
0.4	0.45579	0.46762	0.47724	0.48525	0.49206
0.5	0.35450	0.36627	0.37588	0.38391	0.39075
0.6	0.25918	0.27016	0.27921	0.28684	0.29336
0.7	0.17083	0.18018	0.18802	0.19471	0.20049
0.8	0.09221	0.09891	0.10468	0.10972	0.11415
0.9	0.02968	0.03269	0.03541	0.03788	0.04013

TABLE 3.7: C_1, C_2, C_3, R_4, R_5, ..., R_{10} *as a function of p.*

p	C_1	C_2	C_3	R_4	R_5
0.1	0.71360	0.75276	0.77233	0.78476	0.79371
0.2	0.53100	0.59315	0.61985	0.63768	0.64881
0.3	0.39016	0.46276	0.48927	0.51144	0.52186
0.4	0.27807	0.35110	0.37328	0.40030	0.40827
0.5	0.18872	0.25434	0.27443	0.30196	0.30627
0.6	0.11871	0.17129	0.19367	0.21484	0.21508
0.7	0.06593	0.10225	0.12218	0.13808	0.13505
0.8	0.02905	0.04866	0.06198	0.07242	0.06837
0.9	0.00723	0.01316	0.01806	0.02223	0.02000

p	R_6	R_7	R_8	R_9	R_{10}
0.1	0.79944	0.80586	0.80701	0.81008	0.81034
0.2	0.65611	0.66656	0.66297	0.66778	0.66601
0.3	0.53148	0.54391	0.54114	0.54672	0.54358
0.4	0.42086	0.43262	0.43370	0.43772	0.43454
0.5	0.32111	0.33043	0.33373	0.33476	0.33376
0.6	0.23018	0.23687	0.24003	0.23902	0.24089
0.7	0.14785	0.15279	0.15561	0.15527	0.15714
0.8	0.07665	0.08050	0.08407	0.08595	0.08413
0.9	0.02297	0.02509	0.02815	0.03025	0.02678

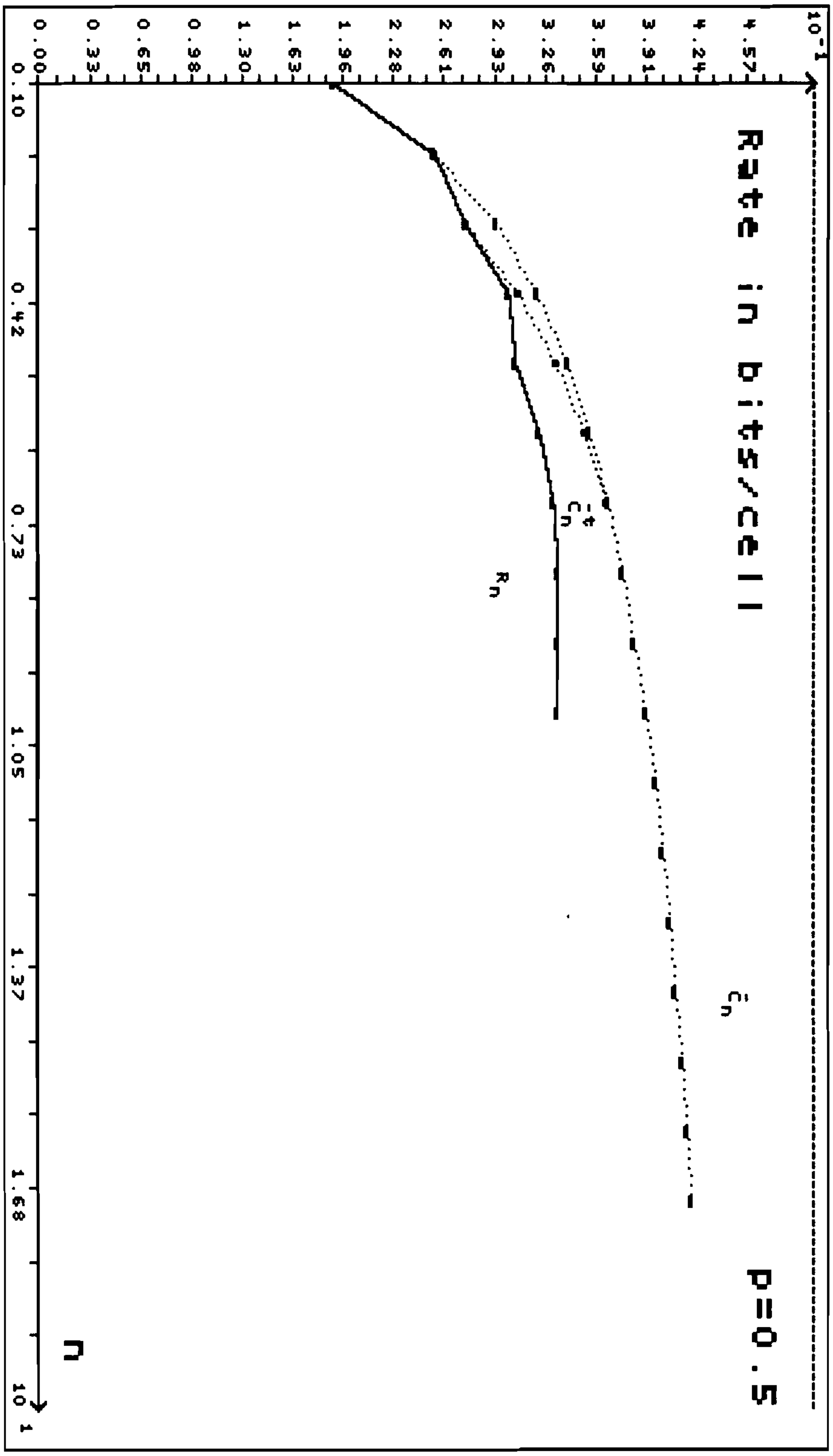


Figure 3.2: $\overline{C}_n$, $\overline{C}_n^t$ and R_n as a function of n in case $p = 0.5$.

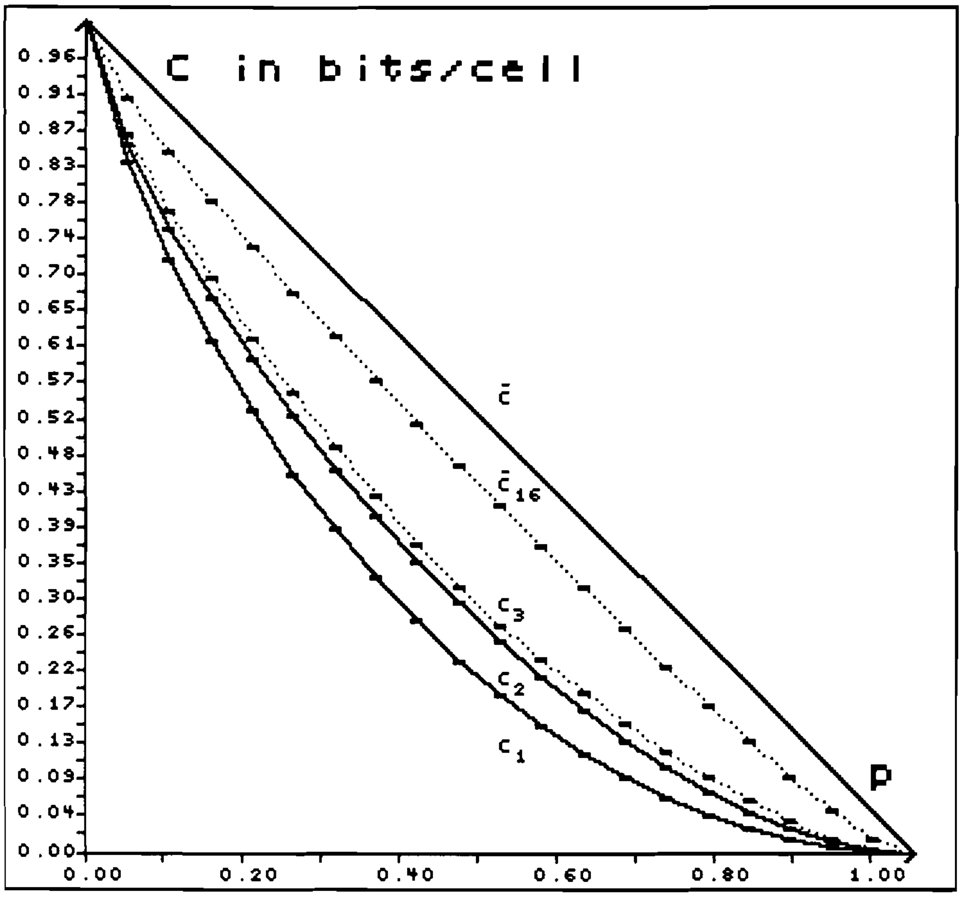


Figure 3.3: C_1, C_2, C_3, $\bar{C}_{16}$ *and* $\bar{C}$ *as a function of* p.

4 MULTIPLE-REPETITION FEEDBACK CODING

4.1 INTRODUCTION

In Chapter 2 we considered the BDC^n ($n \geq 1$) as a superchannel with as many superinputs as outputs: $m = 2^n$. A capacity-achieving set of m superinputs describes the storage strategy for every sequence to be stored in the memory. Furthermore, it was established that the superchannel can be considered a DMC with noiseless feedback. From Chapter 3 at last, the derived channel is SS-symmetric in case n = 1 and WS-symmetric in case $n \geq 2$.

As in general the storage strategy cannot avoid errors an error-correcting code is neccesary. In this chapter we study the class of Multiple-Repetition Feedback Codes (MRFC) with respect to the derived superchannel.

The, so-called, SS-MRFC scheme for the BSC originates from Horstein [HOR63] in 1963. His algorithm is often referred to a binary search (for message points on the interval ⟨0,1⟩) with lies.

In 1971 Schalkwijk [SCH71] simplified Horstein's scheme, by claiming a certain error-correction capability for his scheme.

Assuming this claim, the noisy binary search becomes symmetric and in consequence the feedback strategies resulting from such a search, can be described without using the artifice of message points on the ⟨0,1⟩ interval.

Moreover, Schalkwijk's code strategies are adaptable for both sequential [SCH73] and non-sequential use and their performance corresponds asymptotically with the performance of Berlekamp's optimum codes [BER68]. Even for the non-sequential decision strategy the error probability vanishes exponentially fast at channel capacity [SCH73].
Becker, a Ph.D.-student of Schalkwijk's, generalized the error-correcting mechanism technique described by Schalkwijk for m - ary WS-symmetric channels [BEC73]. It was established by Becker, that the WS-MRFC strategies are also asymptotically optimum for certain rates.

In the Sections 4.2 and 4.3 we adapt the MRFC scheme to the BDC^n ($n\geq 1$).

4.2 SS-MRFC SCHEME FOR BDC^1

4.2.1 ERROR-CORRECTING MECHANISM

The derived superchannel of BDC^1 is a BSC with crossover probability p/2 (see Section 2.2).
Schalkwijk's binary feedback scheme can be applied without any adaption.

From the general MRFC scheme, depicted in *figure 4.1* we describe the error-correcting mechanism.
As most applications in computer memories are word-organized, we consider the non-sequential scheme.

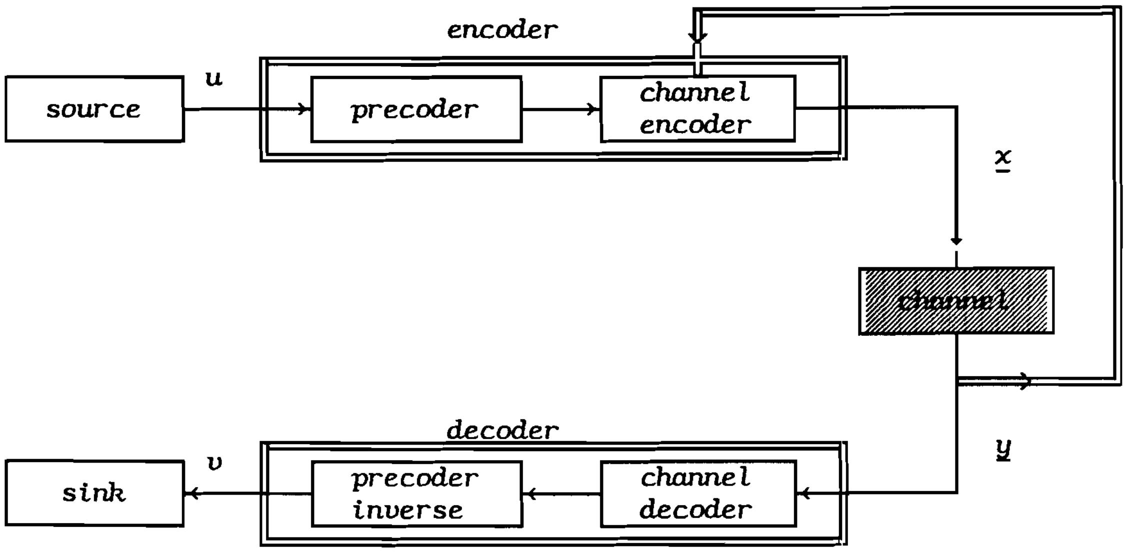


Figure 4.1 : The general MRFC system

The precoder accepts the message source output u and converts it into an unique binary l-tuple with the restriction that there be no subsequences of the form ij^k where $i \neq j$ and $i,j \in \{0,1\}$. Let $M_l(k)$ be the number of possible binary l-tuples (message sequences) described above. Then $U = M_l(k)$ is assumed. This insures that the coding scheme is used most efficiently. An optimal realisation of the precoder (and of the precoder-inverse) is the, so-called, enummerative coding scheme. A detailed describtion is given in [BEC73].

The channel encoder accepts a message sequence and operates as follows. Suppose e errors are to be corrected, then a tail $ke + 1$ binary symbols long is appended to the message sequence to form an appended message sequence. Just like the message sequence the tail sequence does not contain subsequences ij^k. Neither does such a subsequence appear across the junction of the message part and the tail part of the appended message sequence. The tail length plus the message sequence length constitutes the blocklength, N: i.e.

$$N = l + ke + 1 \tag{4.1}$$

Let us give an example: $k = 3$, $e = 3$ and the message sequence is 010010.
Suppose the tail is 1010101010 then the appended message sequence is 0100101010101010.

Now, from the appended message sequence and the defect-pattern $\underline{d}$, the codeword $\underline{x}$ is formed according to the storage strategy, described by the superinputs $\underline{X}_1 = (0,0,1)$ and $\underline{X}_2 = (1,0,1)$. We refer to the interlude in Section 2.2 for a detailed description of the way to use the storage strategy. Obviously, we associate with $\underline{X}_1$ and $\underline{X}_2$ the symbols 0 and 1, respectively.
Notice, that the strategies are compatible with the corresponding defect values. Hence $\underline{x} = \underline{y}$. An error occurs if a symbol of the appended message sequence does not agree with the value of the corresponding defect. If in the course of the codeword formation an error occurs, the appended message sequence remaining to be transmitted, is modified by deleting k symbols from the end and inserting k repetitions of the symbol stored incorrectly at the beginning. These k repetitions are referred to the correcting sequence for the symbol in error. Formation continues in this way, modifying the remaining sequence as described whenever an error occurs, until N (the blocklength) symbols are sent.

In the example above, suppose the defect-pattern $\underline{d}$ =(2,1,3,1,3,1,2,1, 2,1,2,1,3,1,2,1). Now, the codeword $\underline{x}$ = $01\hat{1}(0\hat{1}(000)0)0\hat{0}(111)01$. The $\hat{\ }$ denotes an error, while the parentheses mark the correcting sequences.

The purpose of the channel decoder is to retrieve the message sequence from the received sequence $\underline{y}$. The general procedure employed is very simple. Scan the received sequence from the right for the rightmost subsequence of the form 10^k or 01^k and collapse this into a 0 or 1 respectively, to form a new sequence. Next apply the same collapsing procedure to the new sequence. The collapsing in this manner is continued until no further reduction is possible. It is shown in Section 4.4 that if no more than e errors occur, then the first l symbols of the final reduced received sequence form the original message sequence. This is shown for particular tail constructions.

In the example above the received sequence is reduced as follows:
011(01(000)0)00(111)01 $\longrightarrow$ 011(01(000)0)0101 $\longrightarrow$ 011(000)0101 $\longrightarrow$ 0100101.
Since l = 6 the message sequence is 010010.

The precoder-inverse is just that, the inverse operation of the precoder. It accepts the message sequence which is output from the channel decoder and outputs the appropriate message index [BEC73].

4.2.2 ASYMPTOTIC PERFORMANCE

According to the definition in the preliminary chapter, the rates of the MRFC, just described, have the form :

$$R(N) = \frac{{}^2\log M_\ell(k)}{N} \tag{4.2}$$

$M_\ell(k)$ is the number of possible binary sequences of length ℓ, which do not contain subsequences of the form ij^k for $i \neq j$. The blocklength of the code is N.

Using a simple recurrence relation for $M_\ell(k)$, Becker [BEC73] derives the following rate equation for aymptotic blocklength N :

$$R(\infty) = (1 - k.f_e).{}^2\log \alpha_1 \qquad \text{for } k \geq 3, \tag{4.3}$$

where $f_e = e/N$ is the error fraction and α_1 is the only real root, with absolute value greater than one, of the equation :

$$f(\alpha) = \alpha^k - m.\alpha^{k-1} + m - 1 = 0 \tag{4.4}$$

Eq. (4.4) is the, so-called, characteristic equation of the recurrence relation mentioned above.

In the BDC-case the error fraction $f_e = p/2$.

Optimization of the rate function in eq. (4.3) over parameters α_1 and k, under the condition given in eq. (4.4), is satisfied by maximizing the Lagrangian :

$$\mathcal{L}(\alpha_1,k) = R(\infty) + \lambda.f(\alpha_1) \tag{4.5}$$

There is an unique solution. The optimum multiplier λ_{opt}, α_{1opt} and k_{opt} are respectively :

$$\lambda_{opt} = \frac{(1 - \ell_e)}{\ln(2)} \tag{4.6}$$

$$\alpha_{1opt} = 2(1 - \ell_e) \tag{4.7}$$

$$k_{opt} = \frac{\ln(1 - \ell_e) - \ln(\ell_e)}{\ln(1 - \ell_e) + \ln(2)} \tag{4.8}$$

Substituting the eq.'s (4.7) and (4.8) into eq. (4.3) yields :

$$R_{opt}(\infty) = C_1 = 1 - h(\ell_e) \qquad \text{for } k \geq 3 \tag{4.9}$$

For a particular repetition code, defined by k, the straight line performance curve in eq. (4.3) and the capacity curve in eq. (4.9) have a point in common at $\ell_e = 1 - \alpha_1/2$ (eq. (4.7)). As the performance curve never exceeds the upperbound, this must be a point of tangency. Thus, the performances of the codes for $k \geq 3$ corresponds to the straight lines tangents to the C_1 - curve, which intersects the R-axis at the points $1/k$.

In case $k = 2$ the efficiency claimed by eq. (4.3) cannot be proved. On the contrary, Berlekamp [BER68] has shown that the efficiency is upperbounded by eq. (4.9) and a straight line drawn tangent to the C_1-curve from the point $(\ell_e, R(\infty)) = (1/3, 0)$. That is, for error fractions ℓ_e below the ℓ_e at which this straight line hits the C_1-curve the

efficiency is upperbounded by this straight line and in fact this portion of the bound is attainable ($k = 3$).
The optimal k_{opt} as a function of p ($p = 2.\ell_e$) is illustrated in *figure* 4.2.

Assuming a fixed blocklength N, there are two contrary effects affecting the efficiency of a code :

1) Suppose l is fixed, then $M_l(k)$ is an increasing function of k.
2) On the other hand, l is a decreasing function of k : $l = (1 - k.\ell_e).N - 1$ from eq. (4.1).
   So, neglecting effect 1) $M_l(k)$ is a decreasing function of k.

As only effect 2) is ℓ_e - dependent, the optimal k_{opt} is a decreasing function of ℓ_e. In other words, in case of a high error fraction k_{opt} must be small to preserve from correcting only correcting sequences in the coding process.

Figure 4.3 depicts some performance lines for k being an integer, as we can only repeat a whole number of times. Notice, that in the area of practical importance ($p \leq 0.1$) the capacity C_1 is approached tight for all values of p. Taking a closer look in this area we obtain *figure* 4.4.

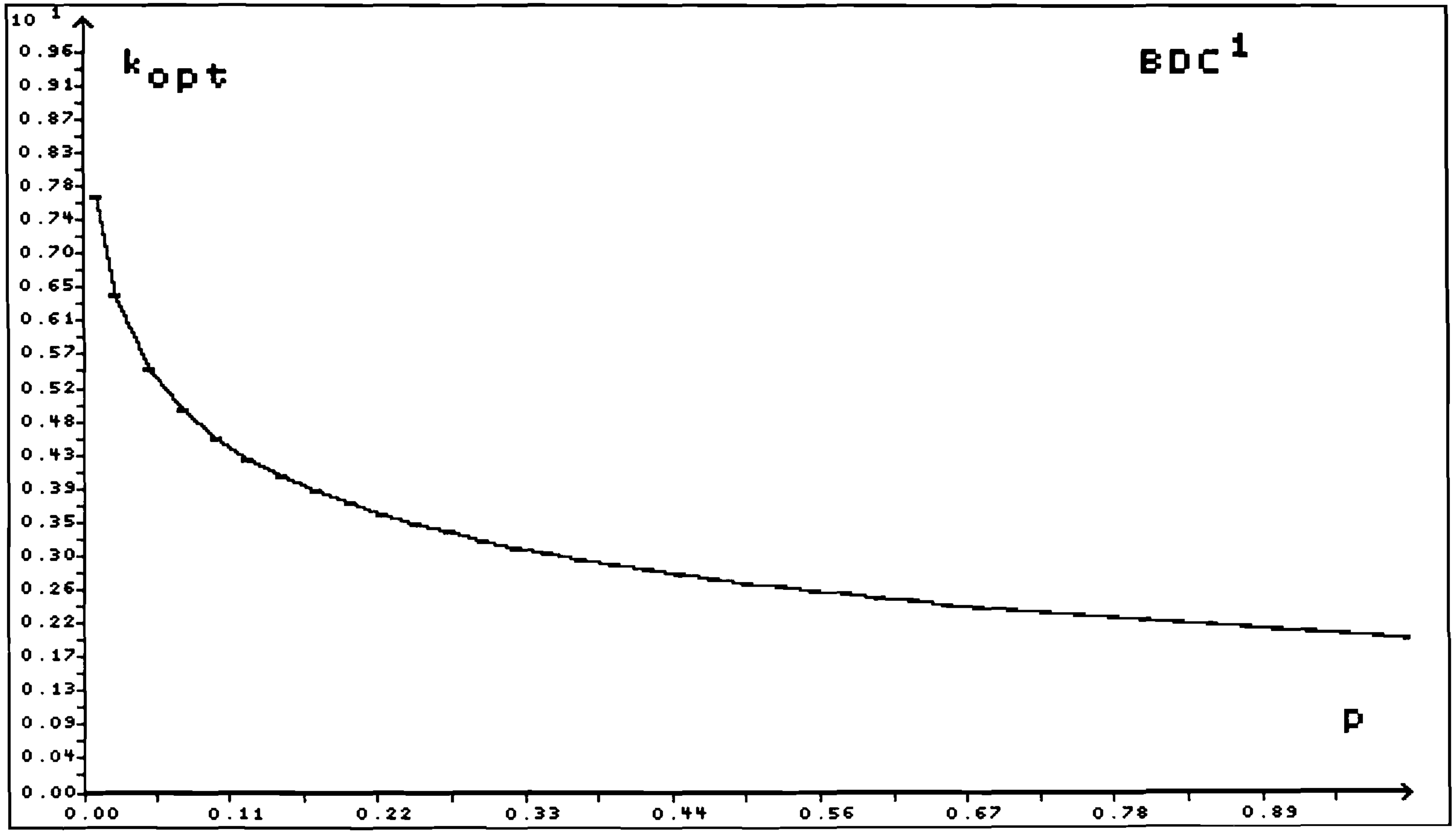


Figure 4.2 : The optimal repetition factor k_{opt} as a function of p.

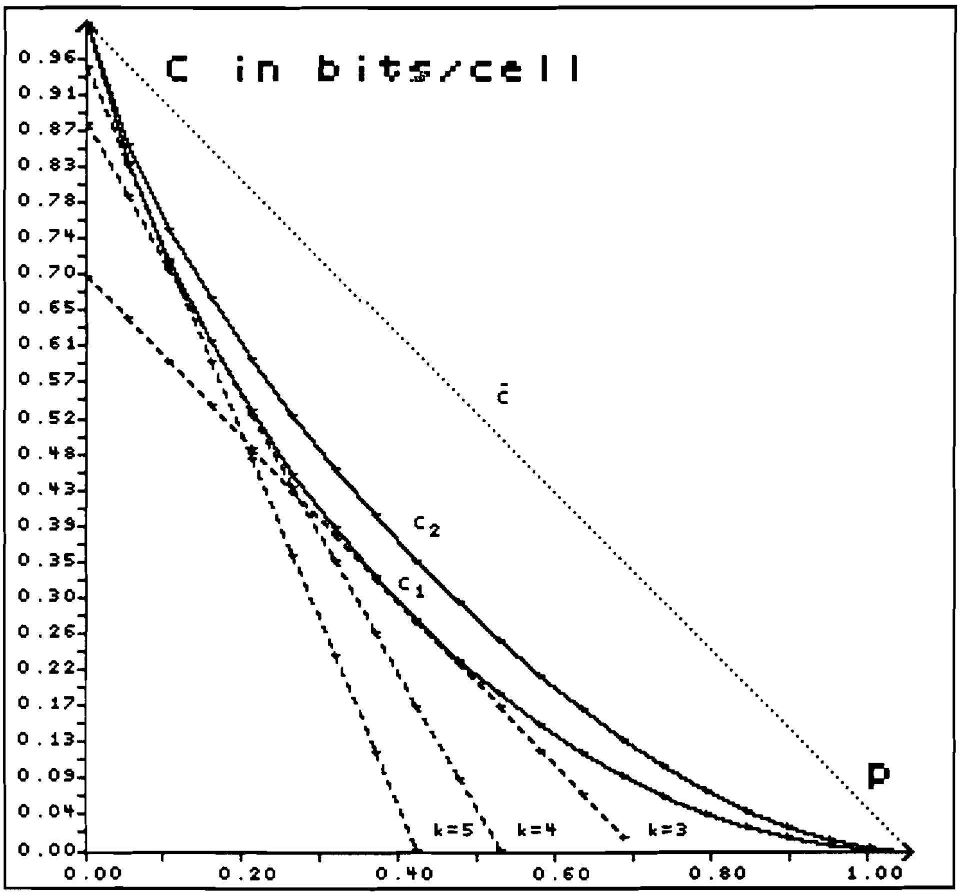


Figure 4.3 : SS-MRFC performance in case of BDC^1.

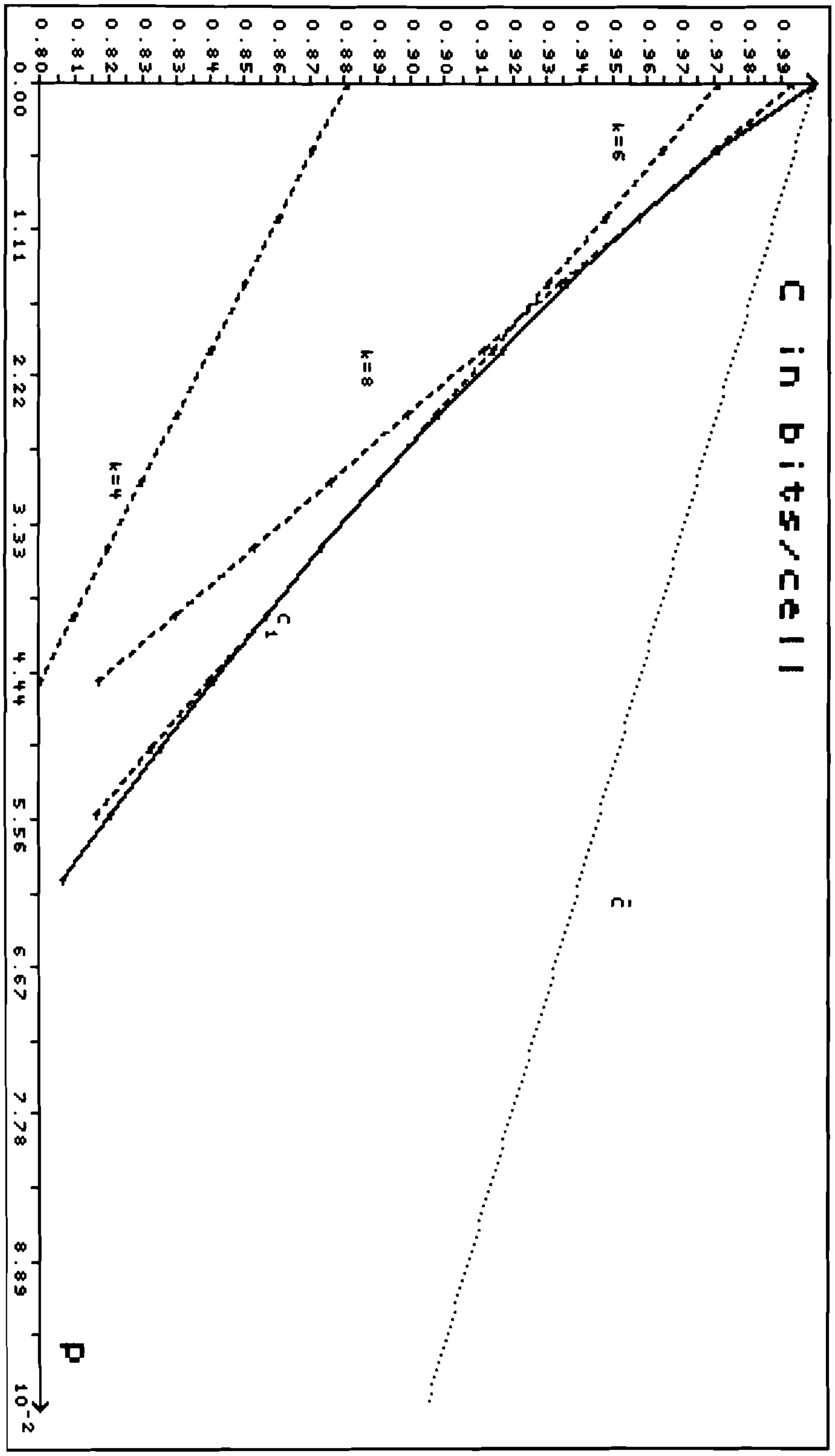


Figure 4.4 : SS-MRFC performance in case BDC^1 for $p \leq 0.1$.

4.3 WS-MRFC SCHEME FOR BDC^n

4.3.1 ERROR-CORRECTING MECHANISM

In case $n \geq 2$ the superchannel derived from BDC^n has $m = 2^n \geq 4$ superinputs and an equal number of outputs. With each superinput we associate a symbol in V_n, i.e. a binary vector of length n. As the channels are WS-symmetric the generalized WS-MRFC's of Becker [BEC73] are applied.

The general structure of the m - ary WS-MRFC system is identical to the SS-MRFC system, depicted in *figure 4.1*, except that each symbol is a binary vector of length n.

Instead of treating all species of errors equally, as in Section 4.2, we distinguish m - 1 different errors, corresponding to the m - 1 different n-tuples resulting from $\underline{j} \oplus \underline{i}$, $\forall\ \underline{i}, \underline{j} \in V_n$ and $\underline{i} \neq \underline{j}$. Considering these m - 1 different n-tuples as binary numbers we label them in order of increasing magnitude. An error is said to be of 'type c' if $\underline{j} \oplus \underline{i}$ has label c $(1 \leq c \leq m-1)$, where $\underline{j}$ is an input to the channel and $\underline{i}$ the corresponding output.

Let us denote the number of errors of type c by e_c, $1 \leq c \leq m-1$. With each error-type we associate a repetition factor k_c and an error fraction f_{ec}. The factors k_c are fixed for a particular code.

Again, the coding strategy requires that there be an unique l-tuple, called an information or message sequence, for each possible message u. The l-tuples, however, are restricted to have no subsequences of the form $\underline{i}\underline{j}^{k_c}$, with $\underline{i} \neq \underline{j}$ and where c is the label of the n-tuple $\underline{j} \oplus \underline{i}$. An enumerative coding scheme is assumed [BEC73].

If a WS-MRFC scheme is to correct $\underline{e} \triangleq (e_1, e_2, \dots, e_{m-1})$ errors, then a tail at least :

$$(\underline{k}.\underline{e}) + 1 \tag{4.10}$$

symbols long is appended to the message sequence by the channel encoder to form an appended message sequence of length N. Just like the message sequence the tail sequence does not contain subsequences $\underline{i}\underline{j}^{k_c}$ ($\underline{i} \neq \underline{j}$ and $\underline{j} \oplus \underline{i}$ is labeled by c). Neither does such a subsequence appear across the junction of the message part and tail part of the appended message sequence. By $\underline{k}$ we denote the vector $\underline{k} \triangleq (k_1, k_2, \dots, k_{m-1})$.

Now, symbol by symbol is stored according to the storage strategy, which is described by the associated superinput $\underline{\underline{X}}$. If in the course of the codeword formation an error occurs, let us say of type c, then the appended message sequence, remaining to be transmitted, is modified by deleting k_c symbols from the end and inserting k_c repetitions of the symbol stored incorrectly at the beginning. These k_c repetitions form the correcting sequence.
The channel decoder scans the received sequence from the right for the rightmost subsequence of the form $\underline{i}\underline{j}^{k_c}$ and collapse this into a $\underline{j}$ to form a new sequence. This procedure is continued until no further reduction is possible. Like in case n = 1, it is shown in Section 4.4,

that the MRFC strategy described above corrects all patterns of errors in which the associated error vector $\underline{e}'$ satisfies : $(\underline{k}.\underline{e}') < N - \ell$. This is proved only for appropriate choices of tail constructions. Finally, the precoder-inverse retrieves the message index u from the channel decoder output.

To illustrate what is described above, we give an example for n = 2. The storage strategies $\underline{\underline{X}}_1, \underline{\underline{X}}_2, \underline{\underline{X}}_3$ and $\underline{\underline{X}}_4$ are tabulated in Appendix B. Moreover, the numbering of the error-types is given.

Suppose $\ell = 6$, $\underline{k} = (3,3,2)$ and $\underline{e} = (1,0,1)$, then the blocklength of the code N = 12. Furthermore, let the message sequence be 00,10,10,11,00,01, the tail sequence 10,11,00,01,10,11 and the defect-pattern (see *TABLE 2.1*) $\underline{d} = (1,3,1,9,1,1,2,1,3,2,1,8)$.

Consequently, the codeword $\underline{x}$ = 00,$\hat{0}\hat{1}$,(10,$\hat{1}\hat{1}$,(10,10,10)),10,11,00,01,10.

The first symbol error is of type 3 and the second of type 1.

The decoding of the received word $\underline{y} = \underline{x}$ passes the following intermediate stages : 00,$\hat{0}\hat{1}$,(10,$\hat{1}\hat{1}$,(10,10,10)),10,11,00,01,10) $\longrightarrow$ 00,$\hat{0}\hat{1}$,(10,10),10,11,00,01,10 $\longrightarrow$ 00,10,10,11,00,01,10. Since $\ell = 6$ the message sequence is 00,10,10,11,00,01.

4.3.2 ASYMPTOTIC PERFORMANCE

According to the last section the obvious generalisation of the eq.'s (4.3) and (4.4) are :

$$R(\infty) = (1 - \sum_{c=1}^{m-1} k_c \cdot \ell_{ec}) \cdot {}^{m}\log \alpha_1 \qquad \text{bits/cell} \qquad (4.11)$$

and

$$f(\alpha) = \alpha^{k_{max}}.(1 - m.\alpha^{-1} + \sum_{c=1}^{m-1} \alpha^{-k_c}) = 0 \tag{4.12}$$

The parameter α_1 is the only real root with absolute value greater than one of eq. (4.12) and $k_{max} = \max_c k_c$. Using Lagrange's optimization method we obtain :

$$\lambda_{opt} = - \frac{\ell_{e_0}}{\ln(m)} \tag{4.13}$$

$$\alpha_{1opt} = m.\ell_{e_0} \tag{4.14}$$

$$k_{c_{opt}} = \frac{\ln(\ell_{e_0}) - \ln(\ell_{ec})}{\ln(\ell_{e_0}) + \ln(m)} \tag{4.15}$$

Where ℓ_{e_0} is defined by $\ell_{e_0} \triangleq 1 - \sum_{c=1}^{m-1} \ell_{ec}$.

Substituting the eq.'s (4.14) and (4.15) in eq. (4.11) yields :

$$R_{opt}(\infty) = 1 + \sum_{c=0}^{m-1} \ell_{ec}.{}^{m}\log \ell_{ec} . \tag{4.16}$$

which is identical to eq. (3.16). This means, that these WS-MRFC's are optimal in a sense that they achieve capacity.

In case n=2 we know a capacity-achieving storage strategy and consequently their transition probabilities (Appendix B). Hence,

$$\ell_{e_0} = (1 - p/2)^2$$
$$\ell_{e_1} = p^2/4$$
$$\ell_{e_2} = p^2/4$$
$$\ell_{e_3} = (1-p/2)^2 - (1-p)^2 \qquad (4.17)$$

Substituting these in eq. (4.16) we obtain :

$$R_{opt}(\infty) = C_2 \qquad (4.18)$$

The performance curves for some MRFC's, characterized by $\underline{k}$ results from eq. (4.11) by substituting eq. (4.17) :

$$R(\infty) = (1 - k_3.p + 1/4.(3k_3 - k_1 - k_2).p^2).{}^2\log(\sqrt{\alpha_1}), \qquad (4.19)$$

which are parabolas. They hit the capacity curve in one point for the optimal parameters in the eq.'s (4.14) and (4.15).

The optimal $\underline{k}_{opt}$ are illustrated in *figure* 4.5. Using the same argument as in the case of n=1, they are decreasing functions of p. *Figure* 4.6 depicts some performance curves for the k_c - facotrs to be integers. Again, in the area of practical importance ($p \leq 0.1$) the capacity C_2 is approached tight for all values of p (see also *figure* 4.7). It can be readily verified, that for integer values of k_c C_2 is not achieved. Furthermore, for some vectors $\underline{k}$ the efficiency claimed in eq. (4.11) cannot be proved. Hereto we refer to the next section.

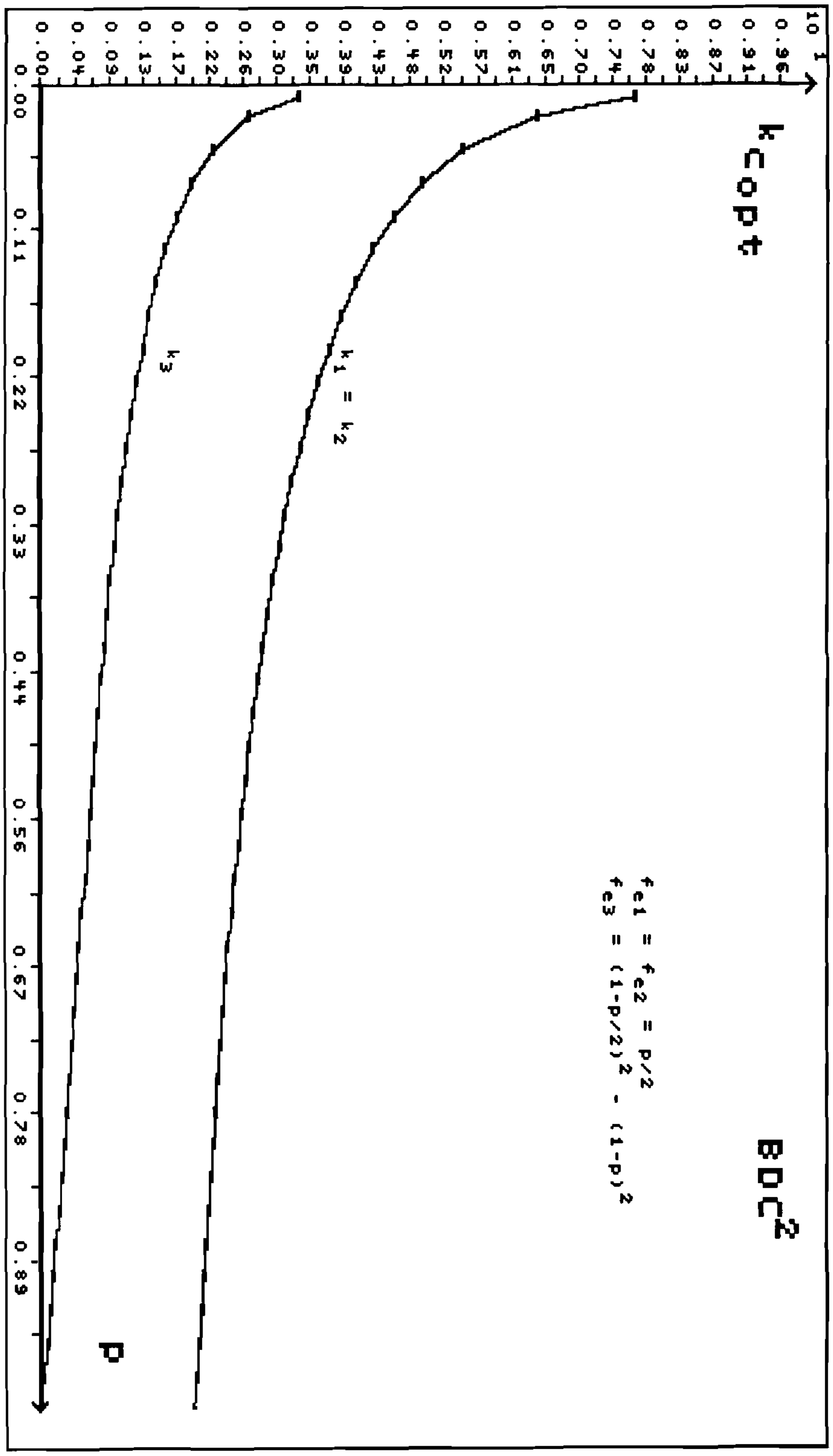


Figure 4.5 : The optimal repetition factors $k_{c_{opt}}$ as a function of p.

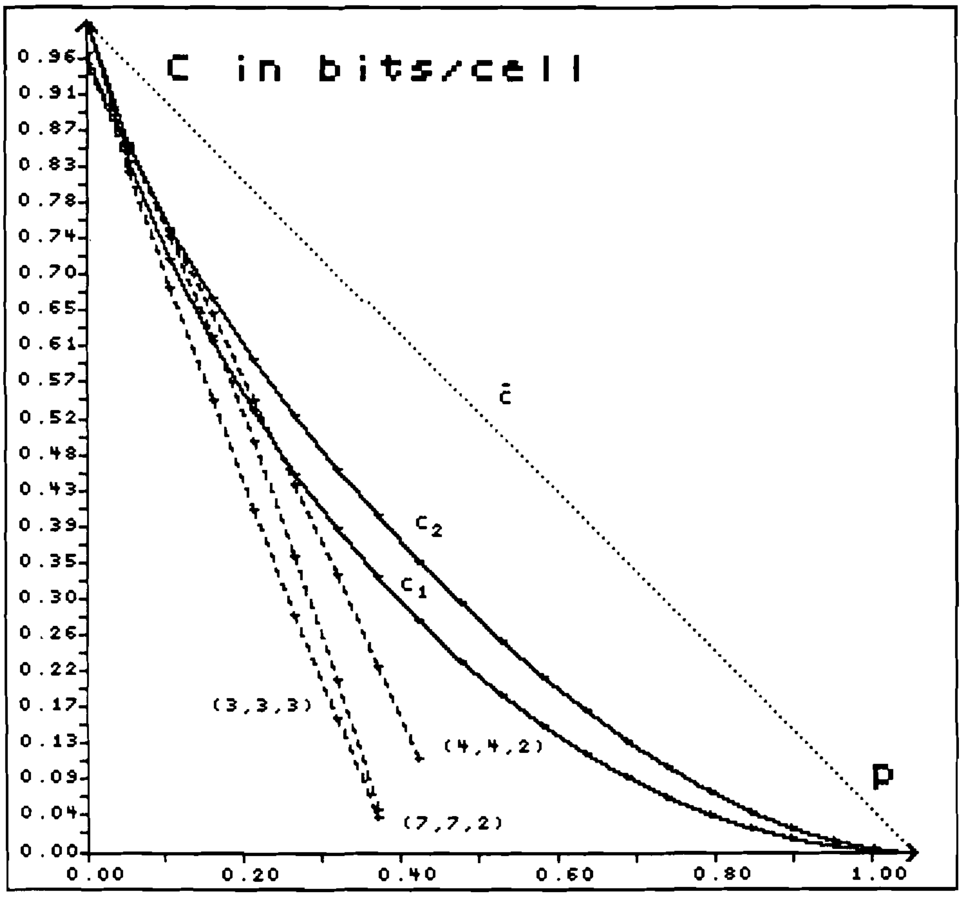


Figure 4.6 : WS-MRFC performance in case of BDC^2.

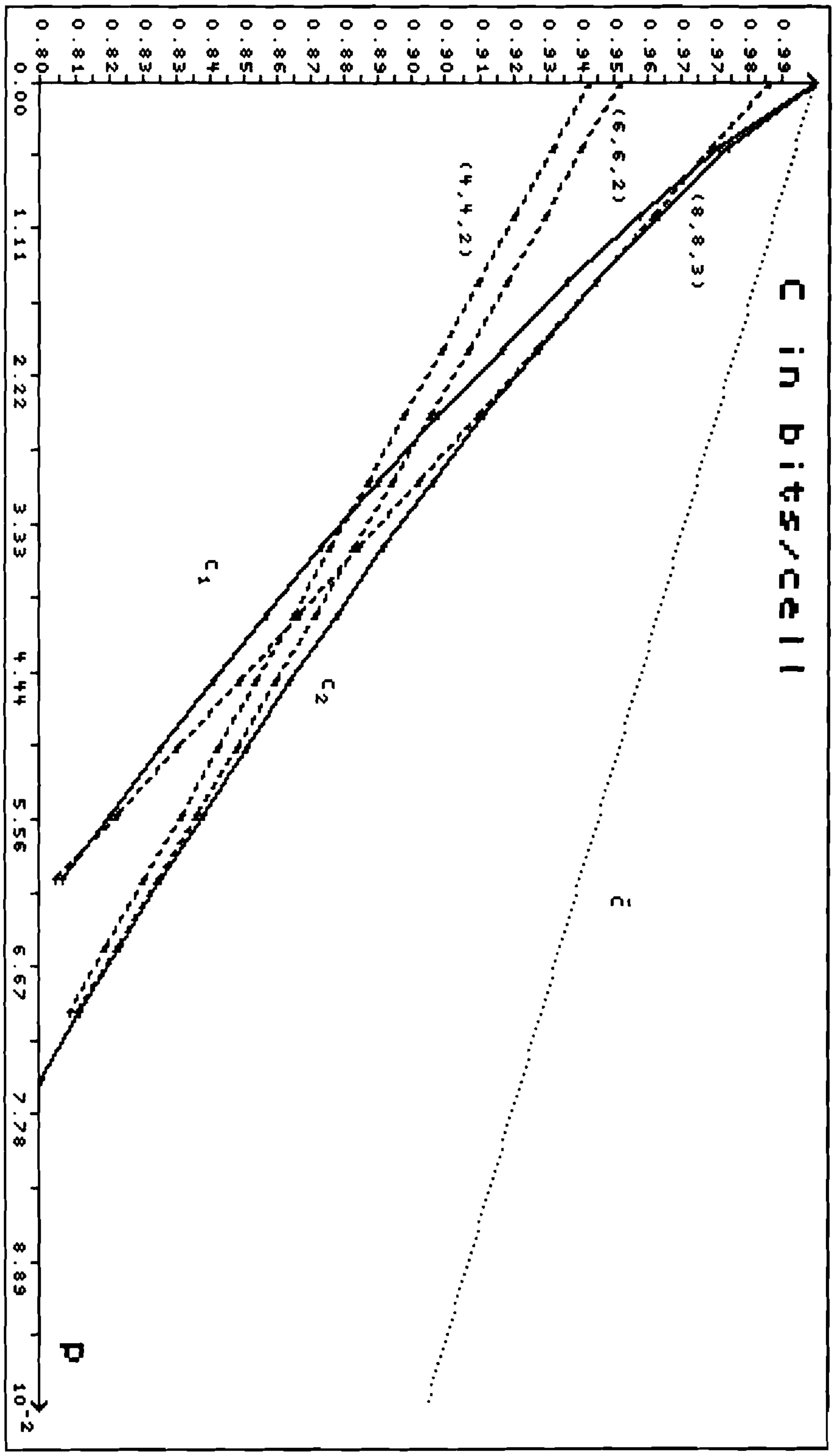


Figure 4.7 : WS-MRFC performance in case of BDC^2 for $p \leq 0.1$.

4.4 ERROR-CORRECTING CAPABILITY

In Section 4.2 it was claimed, that if the number of errors in the received code block $e' \leq e$, then the channel decoder (*figure 4.1*) reduces this sequence to a sequence in which the first $\ell = N - k.e$ symbols match the original message sequence. The WS-MRFC scheme in Section 4.3, on the other hand, claims to correct all patterns of errors in which the associated error vector $\underline{e}'$ satisfies : $(\underline{k}.\underline{e}') < N - \ell = (\underline{k}.\underline{e}) + 1$.
It was also mentioned by the way, that these claims can only be proved for particular tail sequences.
Both are now verified.

In [BEC73] Becker gives different proofs for the error correction capability of SS-*MRFC* and A-MRFC. The proof in case of WS-MRFC is analogously to the asymmetric case. As both the m-ary SS- and WS-symmetric channel are just special cases of the asymmetric channel, we only consider the most general A-MRFC scheme.
The alternative proof for the A-MRFC scheme now presented holds for three different tail definitions. In Section 4.4.3 we extract from these definitions analogue tail definitions in case of SS- and WS-MRFC schemes.

Before beginning the discussion of the error-correction power, we introduce some notational conventions and definitions.

4.4.1 NOTATIONAL CONVENTIONS AND DEFINITIONS

It is convenient to consider the m message symbols associated with the superinputs and the m outputs as binary numbers of length n. We assume both identical sets to be arranged in order of increasing magnitude, e.g. 000 is indexed by 1 and 111 by 8.
From now on ,we only use these indices to refer to message and output symbols.

Definition 4.1 :

$I_m = \{1,2,\ldots,m\}$, *i.e. the code alphabet which consists of the indices corresponding with the symbols in* V_n.

Definition 4.2 :

A_t *is the set of indices in* I_m, *which appear in the tail sequence.*

Unlike the SS- and WS-MRFC system the A-MRFC strategies distinguish error types, which depend on both the symbol transmitted and the symbol received.

Definition 4.3 :

Suppose we want to store $i \in I_m$ *and the corresponding output is* $j \neq i$, *then an error of type* (i,j) *occurred.*

The symbol $e_{i,j}$ is used to represent the number of transitions from i

to j in a particular codeword. Organized into an m x m matrix, we have the, so-called, 'error matrix' E.
Analogously to the error matrix E, we define the 'message error matrix' E_m and the 'tail error matrix' E_t. Obviously, $E = E_m + E_t$.
The number of errors e, recorded by E, is equal to the sum of the off-the-main-diagonal-terms of E. The terms e_m and e_t are related to the matrices E_m and E_t respectively, in a similar way.

With each error-type $e_{i,j}$ we associate a repetition factor $k_{j,i}$. The total set of repetition factors characterizes the subsequences, which are excluded from the basic code language words and which are used by the channel decoder for the correction of errors.
According to our intuition we define $k_{i,i} = 0$, for all $i \in I_m$. Collecting the integers $k_{i,j}$ into a matrix, we have the, so-called, 'characteristic matrix' $\mathcal{K}$.
Now, the tail sequence length can be expressed by :

$$N - \ell = tr(\mathcal{K}E) + 1 \qquad (4.20)$$

The subsequent lemma's, corollary and theorem prove the error capability of the A-MRFC scheme, assuming the tails are constructed according to one of the following definitions.

Definition 4.4 :

Let $A_t \subseteq I_m$ be such that $i \in A_t$ iff $k_{j,i} \geq N$ for all $j \in I_m$ and $j \neq i$. If $|A_t| \geq 1$, then any sequence $\underline{s} \in A_t^{N-\ell}$ is a valid tail.

Definition 4.5 :

$\underline{s} \in I_m^{N-\ell}$ *Is a valid tail for a message sequence, denoted by* $\bar{m}$, *if* $\bar{\underline{s}}$, *the sequence formed by prefixing* $\underline{s}$ *with the last symbol of* $\bar{m}$, *satisfies both the following :*

1) *i,j appears in* $\bar{\underline{s}}$ *only if both* $i \neq j$ *and* $k_{i,j} \geq 3$.
2) *i appears in* $\bar{\underline{s}}$ *only if both* $k_{j,i} \geq 3$ *and* $k_{i,j} \geq 2$ *for each* $j \neq i$.

Definition 4.6 :

$\underline{s} \in I_m^{N-\ell}$ *Is a valid tail for a message sequence, denoted by* $\bar{m}$, *if* $\bar{\underline{s}}$, *the sequence formed by prefixing* $\underline{s}$ *with the last symbol of* $\bar{m}$, *satisfies both the following :*

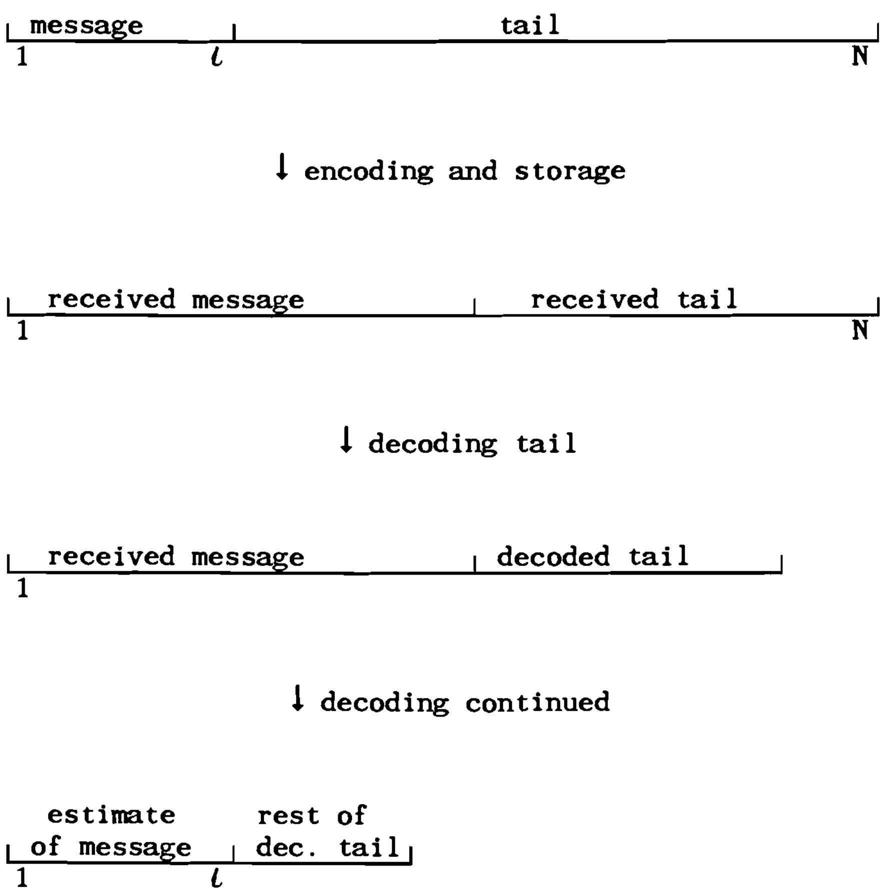
1) *i,j appears in* $\bar{\underline{s}}$ *only if both* $i \neq j$ *and* $k_{i,j} \geq 4$.
2) *i appears in* $\bar{\underline{s}}$ *only if both* $k_{j,i} \geq 2$ *and* $k_{i,j} \geq 2$ *for each* $j \neq i$.

4.4.2 PROOF OF THE ERROR-CORRECTING CAPABILITY OF THE A-MRFC SCHEME

We consider a A-MRFC scheme characterized by $\mathcal{K}$. The blocklength N and the message length ℓ satisfy : $N - \ell = tr(kE) + 1$, where E is denoted as the 'design error matrix' of the scheme. We assume the occurrence of an arbitrary error pattern with associated error matrix $E' = E'_m + E'_t$ and satisfying :

$$\mathrm{tr}(\mathcal{X}E') \leq \mathrm{tr}(\mathcal{X}E) \qquad (4.21)$$

Then, the received tail sequence is at least $\mathrm{tr}(\mathcal{X}E'_t) + 1$ symbols long. The sum of the off-the-main-diagonal-elements of E'_t is denoted by e'_t. Before we go on we give an overview of the different stages in the coding process.



Suppose the errors are such that the correcting sequence for the rightmost error is finished and received error-free. As the received

tail sequence length $\geq$ 1, all message symbols have been transmitted. Now, a simple inductive argument verifies, that all the errors are corrected from right to left.
As the correcting sequence of the rightmost error in the received message sequence is always error-free and completed, the only possibility the received message sequence is decoded incorrectly is the presence of a subsequence of the form $ij^{k_{i,j}}$ across the junction of the received message sequence and the decoded tail sequence.

The next lemma's end corollary are used to prove, that such a subsequence does not appear across the junction on the condition the tail sequence obeys at least one of the three *definitions* 4.4, 4.5 *and* 4.6.
In case the tail sequence is constructed according to *definition* 4.4, then e_t' must be equal to zero to satisfy eq. (4.21). From the argument mentioned above it is clear that the decoder always retrieves the message sequence correctly.
Therefore, we concentrate on the *definitions* 4.5 and 4.6.

Lemma 4.1 :

A received tail sequence, that is at least $tr(\mathcal{K}E_t')$ *symbols long and which has an associated error matrix* E_t', *contains at least one subsequence of the form* $\hat{x}x^{k_{\hat{x},x}}$, *if the corresponding decoded tail sequence starts with* $j^{k_{i,j}}$ *for any* $i,j \in A_m$ *and* $i \neq j$. *By* $\hat{x}$ *we denote the output symbol resulting from the occurrence of an error of type* $(x,\hat{x})$

Proof: 'by contradiction'

Assume *lemma 4.1* is not true, i.e. there is a received tail sequence with error matrix E_t' and length at least $tr(\mathcal{X}E_t')$, that reduces to a sequence starting with $j^{k_{i,j}}$ and which contains no subsequence of the form $\hat{x}x^{k_{\hat{x},x}}$. Constructing such a received tail sequence we have to verify the following starting sequences:

1) $j\hat{l}$
2) $jl\hat{r}$
3) $jlj\hat{r}$
4) $\hat{l}$
5) $l\hat{r}$
6) $lj\hat{r}$

Where $\hat{l}, \hat{r}$ denotes the occurrence of an error of type (l,j) and (r,j), respectively. To satisfy either one of the tail definitions $j \neq l$, $j \neq r$ and $l \neq r$.

In each of the 6 cases we give the longest possible configuration. It is shown in the following that the length of these constructed sequences is always less than $tr(\mathcal{X}E_t')$. Moreover, the length of the constructed sequences in case 2),3) and in case 5),6) are always less than the length of the constructed sequences in case 1) respectively 3). Consequently, this is also true for all possible starting sequences.

1) The longest possible configuration is :

$$j\ \underbrace{\hat{l}\ \hat{l}\ \ldots\ \hat{l}}_{k_{i,j}-2}\ \underbrace{\hat{l}\ l\ l\ \ldots\ l}_{k_{j,l}} \longleftarrow \text{Rest Tail} \longrightarrow$$

As no correcting sequence is completed (no occurrence of $\hat{x}x^{k_{\hat{x},x}}$ in the received tail sequence), there are only symbols 1 transmitted after the first error. The 'Rest Tail' contains the remaining errors in E_t'. Each of them is followed by as many symbols 1 as possible. Notice, that the first symbol in the Rest Tail is received incorrectly. Suppose e.g., that the first transmitted symbol of the Rest Tail (1) is received as a v, then $k_{v,1} - 1$ symbols 1 will follow before the next error occurs.
The difference between $tr(\mathcal{X}E_t')$ and the length of the configuration is given by :

$$(k_{i,j} - 2).k_{j,1} - 1 - (k_{i,j} - 2) \qquad (4.22)$$

In the subsequent cases we also give the longest possible configuration and an expression of the difference between $tr(\mathcal{X}E_t')$ and the length of the configuration.

$$j\ 1\ \underbrace{\hat{r}\ \hat{r}\ .\ .\ .\ \hat{r}}_{k_{1,j}}\ \underbrace{\hat{r}\ \hat{r}\ .\ .\ .\ \hat{r}}_{k_{i,j}-3}\ \underbrace{\hat{r}\ r\ r\ .\ .\ .\ r}_{k_{j,r}} \longleftarrow \text{Rest tail} \longrightarrow$$

The first $k_{1,j}$ errors are necessary to change the first 1 into j.

$$(k_{1,j} + k_{i,j} - 3).k_{j,r} - 2 - (k_{1,j} + k_{i,j} - 3) \qquad (4.23)$$

3)

$$j\ 1\ j\ \underbrace{\hat{r}\ \hat{r}\ .\ .\ .\ \hat{r}}_{k_{1,j}-1}\ \underbrace{\hat{r}\ \hat{r}\ .\ .\ .\ \hat{r}}_{k_{i,j}-3}\ \underbrace{\hat{r}\ r\ r\ .\ .\ .\ r}_{k_{j,r}} \longleftarrow \text{Rest Tail} \longrightarrow$$

$$(k_{1,j} - 1 + k_{i,j} - 3).k_{j,r} - 3 - (k_{1,j} - 1 + k_{i,j} + 3) \qquad (4.24)$$

4)

$$\underbrace{\hat{1}\ \hat{1}\ .\ .\ .\ \hat{1}}_{k_{i,j} - 1}\ \underbrace{\hat{1}\ 1\ 1\ .\ .\ .\ 1}_{k_{j,1}} \longleftarrow \text{Rest Tail} \longrightarrow$$

$$(k_{i,j} - 1).k_{j,1} - (k_{i,j} - 1) \qquad (4.25)$$

5)

$$1\ \underbrace{\hat{r}\ \hat{r}\ .\ .\ .\ \hat{r}}_{k_{1,j}}\ \underbrace{\hat{r}\ \hat{r}\ .\ .\ .\ \hat{r}}_{k_{i,j} - 2}\ \underbrace{\hat{r}\ r\ r\ .\ .\ .\ r}_{k_{j,r}} \longleftarrow \text{Rest Tail} \longrightarrow$$

$$(k_{1,j} + k_{i,j} - 2).k_{j,r} - 1 - (k_{1,j} + k_{i,j} - 2) \qquad (4.26)$$

6)

$$1\ j\ \underbrace{\hat{r}\ \hat{r}\ .\ .\ .\ \hat{r}}_{k_{1,j} - 1}\ \underbrace{\hat{r}\ \hat{r}\ .\ .\ .\ \hat{r}}_{k_{i,j} - 2}\ \underbrace{\hat{r}\ r\ r\ .\ .\ .\ r}_{k_{j,r}} \longleftarrow \text{Rest Tail} \longrightarrow$$

$$(k_{1,j} - 1 + k_{i,j} - 2).k_{j,r} - 2 - (k_{1,j} - 1 + k_{i,j} - 2) \qquad (4.27)$$

It is readily verified, that all the expressions in (4.22) - (4.27) are positive provided *definition* 4.5 or 4.6 is satisfied. Moreover, subtracting either (4.23) or (4.24) from (4.22) we obtain a negative result. Likewise the result of the subtracting of either (4.26) or (4.27) from (4.25) is also negative.

This implies, that received tail sequences, in which the occurrence of the first error is delayed with respect to the occurrence in case 1) and 4), cannot be longer than the configurations in 1) and 4).

Taking everything together, we obtain a contradiction to the assumption. Hence, the lemma is proved. |

Lemma 4.2 :

The channel decoder cannot reduce a received tail sequence with associated error matrix E'_t, *that is at least* $tr(\mathcal{K}E'_t)$ *symbols long, to a sequence which begins with* $j^{k_{i,j}}$ *for any* $i,j \in I_m$ *and* $i \neq j$.

Proof: 'by induction on e'_t'

1) In case $e'_t = 0$ the tail sequence is correctly received. As it is chosen according to either *definition* 4.5 or 4.6, no reduction results and in consequence the lemma holds.

2) 'induction hypothesis' : Assume the lemma holds for every tail sequence with e'_t errors that has length $\geq tr(\mathcal{K}E^0_t)$. By E^0_t we denote the corresponding tail error matrix.

   Then, in order to complete the proof, we have to show that the lemma holds for every tail sequence with error matrix E'_t that has length $\geq tr(\mathcal{K}E^1_t)$ and $e'_t + 1$ errors.

   The proof is by contradiction.

   Assume that *lemma* 4.2 does not hold for all tail sequences with $e'_t + 1$ errors. Consider such a received tail sequence for which *lemma* 4.1 does not hold. From *lemma* 4.1 we know that this sequence contains at least one subsequence of the form $\hat{x}x^{k_{\hat{x},x}}$.

Consider the rightmost one. After one decoding step we obtain a sequence with error matrix E_t^0, length at least $tr(\mathcal{K}E_t^0)$ and e_t' errors, that reduces to a sequence beginning with $j^{k_{i,j}}$. This contradicts the induction hypothesis and consequently completes the proof. |

Corollary 4.1 :

The channel decoder converts a received tail sequence with associated error matrix E_t', that is at least $tr(\mathcal{K}E_t') + 1$ symbols long, to a sequence which begins with the first transmitted tail symbol.

Proof: We consider the first tail symbol. Either this symbol is received correctly or in error.

A) The first tail symbol is received in error:

Assume that the correcting sequence of this first symbol contains an error pattern with error matrix E_{tc}', then this correcting sequence has a length of $tr(\mathcal{K}E_{tc}') + 1$. The remaining part of the received tail sequence consists of $tr(\mathcal{K}E_{tr}')$ symbols with error matrix $E_{tr}' = E_t' - E_{tc}'$.

B) The first tail symbol is correctly received:

The remaining part of the received tail sequence consists of $tr(\mathcal{K}E_t')$ symbols.

From *lemma* 4.2 we know, that in both case A) and B) the remaining part of the received tail sequence does not result in a sequence beginning with $j^{k_{i,j}}$. Hence, the first transmitted tail symbol is decoded correctly. |

Theorem 4.1 :

For arbitrary but fixed blocklength N, fixed message sequence length l and a fixed characteristic matrix $\mathcal{K}$ the A-MRFC *strategy corrects all patterns of errors in which the associated error matrix* E' *satisfies:*

$$tr(\mathcal{K}E') < N - l$$

Proof: The only way the channel decoder does not retain the message sequence correctly, is in case of a subsequence of the form $ij^{k_{i,j}}$ across the junction of the received message sequence and the decoded tail.

Notice, that the last symbol of the received message sequence is correct and unequal to the first transmitted tail symbol. Hence, from *corollary 4.1*, the last symbol of the received message sequence is unequal to the first symbol of the decoded tail. Consequently, an illegal subsequence across the junction can occur only when the message sequence ends in say i and the decoded tail begins with $j^{k_{i,j}}$ $(i \neq j)$. But, such a decoded tail can never occur according to *lemma* 4.2. |

It should be emphasized that the *definitions* 4.4, 4.5 *and* 4.6 are only sufficient conditions for the proof of *theorem 4.1*. The tail requirements may be relaxed slightly, especially for special classes of feedback codes, without altering the validity of *theorem 4.1*. The proof of *lemma 4.1*, however becomes considerably more complicated.

4.4.3 TAIL DEFINITIONS FOR SS-MRFC AND WS-MRFC SCHEMES

By this time we demonstrated the error correction power of the A-MRFC scheme for tail sequences chosen according to the *definitions 4.4, 4.5 and 4.6*, we are able to extract analogue definitions for the SS- and WS-MRFC scheme.

Let us start with the SS-MRFC scheme.

Definition 4.7 :

> $\underline{s} \in I_m^{N-\ell}$ *Is a valid tail for a message sequence, denoted by* $\overline{m}$, *if* $\overline{\underline{s}}$, *the sequence formed by prefixing* $\underline{s}$ *with the last symbol of* $\overline{m}$, *satisfies the following* :
>
> $$i,j \text{ appears in } \overline{\underline{s}} \text{ only if both } i \neq j \text{ and } k \geq 3.$$

Notice that in case $k = 3$ only *definition 4.6* is satisfied.
In case of Section 4.2 $I_m = \{0,1\}$. Hence, the tail is formed by an alternating binary sequence. The first tail symbol must be unequal to the last element of the message string.

In case of the WS - MRFC scheme we obtain the following two tail definitions:

Definition 4.8 :

> $\underline{s} \in I_m^{N-l}$ *Is a valid tail for a message sequence, denoted by* $\overline{m}$, *if* $\overline{\underline{s}}$, *the sequence formed by prefixing* $\underline{s}$ *with the last symbol of* $\overline{m}$, *satisfies the following :*
>
> *i,j appears in* $\overline{\underline{s}}$ *only if both* $i \neq j$ *and* $k_c \geq 3$ *for all* $1 \leq c \leq m-1$.

A tail satisfying *definition 4.8* may be formed by cycling through the elements of I_m beginning with the element ranking just above the last message sequence symbol.

Definition 4.9 :

> $\underline{s} \in I_m^{N-l}$ *Is a valid tail for a message sequence, denoted by* $\overline{m}$, *if* $\overline{\underline{s}}$, *the sequence formed by prefixing* $\underline{s}$ *with the last symbol of* $\overline{m}$, *satisfies the following :*
>
> *i,j appears in* $\overline{\underline{s}}$ *only if* $i \neq j$ *and* $k_{c_1} \geq 4$, *where* c_1 *corresponds with an error of type (j,i) and* $k_c \geq 2$ *for all* $1 \leq c \leq m-1$, $c \neq c_1$.

Suppose that $k_{c_1} \geq 4$, then any sequence in which each symbol in the sequence is equal to the result of the bit-by-bit modulo-2 addition of the previous symbol and the n-tuple with c_1, satisfies *definition* 4.9.

Notice in the example given in Section 4.3, that although the vector $\underline{k}$ does not satisfies one of the two *definitions* 4.8 and 4.9, the decoding is correct. This can be simply explained by the error-free reception of the transmitted tail sequence.

5 REFLECTIONS ON COMPLEXITY

5.1 ON THE COMPLEXITY OF THE STORAGE STRATEGY

In Chapter 3 we saw that a storage strategy, achieving capacity C_n, is fully described by a set of $m=2^n$ superinputs.
Moreover, we saw that because of the WS-symmetry (SS-symmetry in case n=1) only one superinput is necessary to (easily) reconstruct the others. W.l.o.g. we assumed, that only superinputs $\underline{\underline{X}}$ are considered, for which all the elements $\underline{x}_t$ masks the corresponding defect-pattern. Furthermore it was established that $\underline{x}_t$ is equal to the symbol, associated with capacity-achieving superinput $\underline{\underline{X}}$, if this symbol is compatible with the defect-pattern, indexed by t.
In this way, the elements $\underline{x}_t$ in $\underline{\underline{X}}$ which have to be determined, are those corresponding to defect-patterns t, which are not compatible with the symbol to be considered. Moreover, from these elements only the binary values corresponding with the non-defective cells are relevant.
It is readily verified, that the total number of relevant positions T_n is equal to

$$T_n = \sum_{f=1}^{n-1} (2^f - 1).\binom{n}{f} = 3^n - 2^{(n+1)} + 1 \qquad (5.1)$$

In general, the determination of these T_n binary values is tedious, but it is needed only once. In fact knowing these binary values, we store them in a memory so that they can be used in the storage process.

TABLE 5.1 gives T_n for $1 \leq n \leq 10$.

TABLE 5.1 : *T_n as a function of n.*

n	T_n	n	T_n
1	0	6	602
2	2	7	1932
3	12	8	6050
4	50	9	18660
5	180	10	57002

Notice, however, that storing these binary values the efficiency of the strategy is decreased.

Consider e.g. a memory consisting of Z cells. The storage strategy, which takes in each coding step n cells into account, achieves a rate R_n. A strategy, which takes in each step n+1 cells into consideration, is more efficient only if the following inequality holds :

$$\bar{C}_{n+1}.Z - T_{n+1} > R_n.Z - T_n \tag{5.2}$$

or

$$Z > \left\lceil \frac{T_{n+1} - T_n}{\bar{C}_{n+1} - R_n} \right| \tag{5.3}$$

In the eq.'s (5.2) and (5.3) the most favourable situation is considered. The strategy, taking n+1 cells into account, is assumed to achieve the upperbound $\overline{C}_{n+1}$.Furthermore the information that is needed to address the T_n essential bits is neglected. In *TABLE* 5.2 minimal values Z_{min} in case p = 0.1 for which eq. (5.3) holds, are given.

TABLE 5.2 : *The minimum size* Z_{min} *of a memory.*

n	Z_{min}	n	Z_{min}
1	52	6	84714
2	456	7	278056
3	2227	8	851452
4	8453	9	2016939
5	28630		

As a final remark, the decoding process does not need the storage strategy, described by $\underline{\underline{X}}$, to retrieve the original message symbols.

5.2 ON THE COMPLEXITY OF MRFC SCHEMES

From Chapter 4 we know, that the MRFC schemes, described by Becker [BEC73], can be applied to the superchannels in Chapter 3 with just a few adaptions.

Instead of integers we have to deal with binary vectors of length n. Consequently we use an alternative definition for WS-symmetry. Becker used the modulo-m subtract operation on integers to determine the error types. In this thesis on the contrary, we use the bit-by-bit modulo-2 add (or subtract) operation on binary n-tuples.

Except for these simple adaptions the precoder-, encoder-, decoder-, and precoder-inverse algorithms, given in [BEC73, p. 36 figure 2.5, p. 62 figure 4.1, p. 63 figure 4.2, p.37 figure 2.6] can be used.

It was concluded in [BEC73], that the major computational effort is in the operation of the precoder and precoder-inverse. Therefore, a less complex insertion-algorithm was suggested. The efficiency of this algorithm, however, is less than the ennumerative coding scheme.

Apart from the computational effort, the MRFC scheme needs $2^n - 1$ repetition values k_c. The maximum number of bits it takes to store these integers is upperbounded by : $(2^n - 1).\lceil {}^2\log k_{max} \rceil$, where $k_{max} = \max_c k_c$. In case $n \geq 10$, we may neglect this memory space with respect to T_n.

5.3 PERFORMANCE DEGRADATION DUE TO CONSTRAINED BLOCKLENGTH

The complexity of the coding process is a restraint on the blocklength. In the last two sections, we noticed that particularly a great n increases the amount of required memory space. The computational effort, required in the coding process however, increases particularly with growing blocklength N.

In the following we illustrate the effect of finite blocklength on the achievable rate of the MRFC scheme in case of BDC^1 and BDC^2 (Appendix B). The rate function is given by eq. (4.2). From Chapter 4 the information length ℓ is written in the form :

$$\ell = (1 - \sum_{c=1}^{m-1} k_c \cdot \ell_{ec}).N - 1 \qquad (5.4)$$

Furthermore, $M_\ell(k)$ is calculated by using the linear recurrence relation derived by Becker [BEC73, p.52 eq. (3.11)].

TABLE 5.3 : Achievable rate R(n) of MRFC schemes in case of BDC^1 and BDC^2

n = 1	
p = 0.05, k = 5	p = 0.15, k = 4
R(32) = 0.80786	R(32) = 0.59442
R(64) = 0.81814	R(64) = 0.59942
R(128) = 0.82329	R(128) = 0.60878
R(∞) = 0.83492	R(∞) = 0.63482
n = 2	
p = 0.05, $k_1 = k_2 = 5$, $k_3 = 2$	p = 0.15, $k_1 = k_2 = 4$, $k_3 = 2$
R(16) = 0.77728	R(16) = 0.59721
R(32) = 0.80331	R(32) = 0.62229
R(64) = 0.83113	R(64) = 0.63483
R(∞) = 0.85066	R(∞) = 0.64855

Notice that in case n=2 a blocklength N corresponds with 2N bits.

6 CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

In this thesis a new information-theoretical approach towards the BDC with side information at the transmitter was presented.
The approach originates from Schalkwijk in 1986.

It was assumed that the probability of occurrence of 0-defects and 1-defects are independent and identically distributed. This model proves to be suitable in case the imperfections (e.g. in the substrate) affect individual cells.
In this new approach it is also possible to take (permanent) imperfections into account that have a bursty character. The only effect of the bursty character of the defects is a change in the probabilities g_t, $t \in S$.

In case the number of defects f in n memory cells is fixed, the capacity $C_n(f)$ was determined for f=2 and $2 \leq n \leq 7$. These results are far superior to the best achievable rates known till now, which were obtained by using the additive coding scheme of Kusnetsov and Tsybakov. It was shown that the upperbound $\overline{C} = 1-f/n$ is achievable in case $f \in \{1,n-1,n\}$. For all other values of f it was proved that the upperbound is not achievable.
In case of a more general probabilistic model of a memory cell, the capacity C_n was determined for n={1,2,3}. Notice, that in case p=0.5 the capacity C_3 = 0.27443 bit/cell exceeds the value 0.27042, that was claimed by Schalkwijk [SCH86] in 1986.

Moreover an upperbound for arbitrary n was derived, which coincides with C_1 and C_2. This upperbound can be tightened by using the results of the fixed-number-of-defects model.

From Kusnetsov and Tsybakov we know that for asymptotic values of n the capacities $C_n(f)$ $(2 \leq f \leq n-1)$ and C_n approach $\overline{C} = 1 - p$ bit/cell arbitrary tight.

Apart from facilitating the calculations of the capacity C_n, the new approach tells us how we have to use the memory in order to actually achieve capacity. These storage strategies however cannot avoid errors which occur consequent on defects. Therefore, the class of m-ary MRFC was analized for n = 1,2.

Especially for small values of p the performance curves in case the repetition factors are integers, approach the capacity bound tight. Besides that, in case n=1 the performance lines and the capacity curve have a point in common. Instead of choosing repetition factors to be fixed integers we could switch between different integer values in order to improve the performance some more. Such a coding scheme however will have a great impact on the complexity of both the encoder and the decoder.

Futhermore, an alternative and more structured proof of the error-correction capability of the A-MRFC scheme was given. Unlike Becker's proof [BEC73] the proofs for the SS- and WS-MRFC scheme are structured along the same lines as the proof for the assymmetric scheme.

In the last chapter it was noticed that except for a few simple adaptions the encoder and decoder algorithms of Becker can be used. These algorithms have a low complexity.

Detached from the MRFC. the storage strategy once determined, has to be stored in the memory. Calculations show that for a particular memory size this loss in memory space is a restraint on the parameter n of the coding scheme. On the other hand the computational effort of the MRFC scheme increases particularly with growing blocklength N.
The performance degradation due to constrained blocklength was illustrated for some MRFC schemes.

Finally we remark that an issue that was not considered in this thesis is the penalty incurred in obtaining the defect information for the encoder.

6.2 RECOMMENDATIONS

An essential point in the analysis of this new approach is the storage strategy described by the capacity-achieving superinputs of the derived channel. Further research should be focused on optimal or if necessary suboptimal storage strategies that can be generated without the permanent use of memory space.

It was noticed in the last section that the new approach is also suitable for other kinds of (permanent) imperfections. In general however, the derived channels are asymmetric and in consequent harder to analize. Consider e.g. the situation that only 0-defects occur with probability p. The derived channel in case n=1 is a binary z-channel with crossover probability p. Two capacity-achieving superinputs are $\underline{X}_1 = (0,0)$ and $\underline{X}_2 = (1,0)$, where the first elememt in each superinput corresponds with a non-defective position. The capacity in case p=0.5 is equal to 0.32193 bit/cell, which agrees with [SCH86]. As the capacity-achieving input-distribution ($Q(\underline{X}_1) = 0.6$ and $Q(\underline{X}_2) = 0.4$) is also asymmetric precoding of the message sequence is necessary.

Apart from permanent defects random errors are also known to occur in semiconductor memory. Alpha-particles and thermal noise are the most important contributions to these errors. Further study must point out how these random errors can be incorporated in this new approach.

REFERENCES

[AHL86] R. Ahlswede,
"Arbitrarily Varying Channels with States Sequence Known to the Sender",
IEEE Trans. on Info. Theory, Vol. IT-32, No. 5, Sept. 1986, pp. 621-629.

[BAG85] C.P.M.J. Baggen,
"MDS Codes for the Correction of Stuck-at Defects",
Proceedings of the 6th Symp. on Info. Theory in the Benelux, Mierlo - the Netherlands, 23-24 May 1985, pp. 49-53.

[BEC73] D.W. Becker,
"Multiple - Repetition Feedback Coding",
Ph.D. - dissertation, University of California, San Diego, 1973.

[BEL77] I.B. Belov and A.M. Shashin,
"Codes that Correct Triple Defects in Memory",
Problemy Peredachi Informatsii, Vol. 13, No. 4, Oct.-Dec. 1977, pp. 62-65.

[BER68] E.R. Berlekamp,
"Block Coding for the Binary Symmetric Channel with Noiseless Delayless Feedback",
Error Correcting Codes, Wiley, 1968, pp. 61-85,

Reprinted in: Key Papers in the Development of Information Theory, D. Slepian ed., New York: IEEE, 1974, pp. 425-430.

[BUS84] P. Busschbach, "Constructive Methods to solve the Problems of: s-surjectivity, conflict resolution, coding in defective memories". Internal Report, Eindhoven University of Technology, Dec. 1984

[CHE85] C. Chen, "Linear Codes for masking Memory Defects", IEEE Trans. on Info. Theory, Vol. IT-31, No. 1, Jan. 1985, pp. 105-106.

[CHE86] C. Chen, "Error-Correcting Codes for Byte-Organized Memory Systems", IEEE Trans. on Info. Theory, Vol. IT-32, No. 2, March 1986, pp. 181-185.

[GAL68] R.G. Gallager, Information Theory and Reliable Communication, New York: Wiley, 1968.

[HEE81] C. Heegard, "Capacity and Coding for Computer Memory with Defects", Technical Report no. 45, Department of Statistics, Stanford University, May 1981.

[HEE83-1] C. Heegard and A. El Gamal,
"On the Capacity of Computer Memory with Defects",
<u>IEEE Trans. on Info. Theory</u>, Vol. IT-29, No. 5, Sept. 1983, pp. 731-739.

[HEE83-2] C. Heegard,
"Partitioned Linear Block Codes for Computer Memory with 'Stuck-at' Defects",
<u>IEEE Trans. on Info. Theory</u>, Vol. IT-29, No. 6, Nov. 1983, pp. 831-842.

[HOR63] M. Horstein,
"Sequential Transmission using Noiseless Feedback",
<u>IEEE Trans. on Info. Theory</u>, Vol. IT-9, July 1963, pp. 136-143.

[KUS74] A.V. Kusnetsov and B.S. Tsybakov,
"Coding in a Memory with Defective Cells",
<u>Problemy Peredachi Informatsii</u>, Vol. 10, No. 2, April-June 1974, pp. 52-60.

[KUS78] A.V. Kusnetsov, T. Kasami and S. Yamamura,
"An Error Correcting Scheme for Defective Memory",
<u>IEEE Trans. on Info. Theory</u>, Vol. IT-24, No. 6, Nov. 1978, pp. 712-718.

[LOS78] V.V. Losev, V.K. Konopel'ko and Yu.D. Daryakin,
"Double-and-Triple-Defect-Correcting Codes",
Problemy Peredachi Informatsii, Vol. 14, No. 4, Oct.-Dec. 1978, pp. 98-101.

[PUL85] C.L.M. van Pul,
"Computer Memories with Defective Cells",
Proceedings of the 6th Symp. on Info. Theory in the Benelux, Mierlo - the Netherlands, 23-24 May 1985, pp. 43-47.

[PUL87] C.L.M. van Pul,
"Some Distance Problems In Coding Theory",
Ph.D. - dissertation, Eindhoven University of Technology, Eindhoven, 1987.

[SCH71] J.P.M. Schalkwijk,
"A Class of Simple and Optimal Strategies for Block Coding on the Binary Symmetric Channel with Noiseless Feedback",
IEEE Trans. on Info. Theory, Vol IT-17, No. 3, May 1971, pp. 283-287.

[SCH73] J.P.M. Schalkwijk and K.A. Post,
"On the Error Probability for a Class of Binary Recursive Feedback Strategies",
IEEE Trans. on Info. Theory, Vol.IT-19, No. 4, July 1973, pp. 489-511.

[SCH82] J.P.M. Schalkwijk and A.J. Vinck,
"Coding Strategies for Some Deterministic Multi-user Channels",
Tijdschrift van het Nederlands Electronica- en Radiogenootschap, Vol. 47, No. 4, 1982, pp. 147-154.

[SCH86] J.P.M. Schalkwijk,
"On powers of the Defect Channel and their Equivalence to Noisy Channels with Feedback",
Proceedings of the 7th Symp. on Info. Theory in the Benelux, Noordwijkerhout - the Netherlands, May 22-23 1986, pp. 41-48.

[SHA56] C.E. Shannon,
"The Zero Error Capacity of a Noisy Channel",
IRE Trans. on Info. Theory, Vol.IT-2, 1956, pp. 8-19.

[SHA58] C.E. Shannon,
"Channels with Side Information at the Transmitter",
IBM J. Res. Develop, Vol. 2, Oct. 1958, pp. 289-293.
Reprinted in: Key Papers in the Development of Information Theory, D. Slepian, ed. New York: IEEE, 1974, pp. 339-372.

[TSY75-1] B.S. Tsybakov,
"Defects and Error Correction",
Problemy Peredachi Informatsii, Vol. 11, No. 3, July-Sept. 1975, pp. 21-30.

[TSY75-2] B.S. Tsybakov,
"Additive Group Codes for Defect Correction",
Problemy Peredachi Informatsii, Vol. 11, No. 1, Jan.-March 1975, pp. 111-113.

[VIN85] A.J. Vinck,
"Convolutional Codes and Defects",
Proceedings of the 6th Symp. on Info. Theory in the Benelux, Mierlo - the Netherlands, May 23-24 1985, pp. 55-61.

[VIN86] A.J. Vinck and G. Kerpen,
"Coding for Memories with Defects",
Proceedings of the REL-CON 1986,
Reliability Technology-Theory and Applications,
(Part 2 - Advanced Professional Seminars)
Elseviers Science Publishers BV (North-Holland) 1986.

[VIT79] A. Viterbi and J. Omura,
Principles of Digital Communication and Coding,
McGraw - Hill, 1979.

APPENDIX A: CAPACITY ACHIEVING SETS $S_2(\underline{\underline{X}})$ AND THEIR $a_2(\underline{y})$-DISTRIBUTIONS

n = 4

$S_2(\underline{\underline{X}})$	$a_2(\underline{y})$
0000	6
1110	6
0011	5
0101	4
1001	3

n = 5

$S_2(\underline{\underline{X}})$	$a_2(\underline{y})$
00000	10
11110	10
00111	8
11001	8
01010	2
10100	2

n = 6

$S_2(\underline{\underline{X}})$	$a_2(\underline{y})$
000000	15
001111	14
110011	13
111100	12
010101	3
101010	3

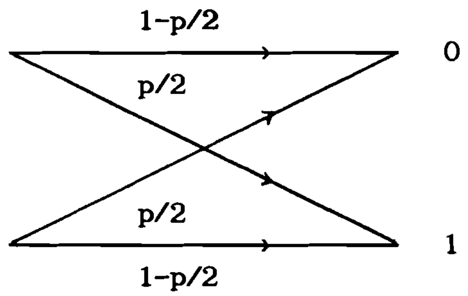
n = 7

$S_2(\underline{\underline{X}})$	$a_2(\underline{y})$
0000000	21
0011111	20
1100011	17
1101100	14
0110101	6
1011010	6

APPENDIX B: SETS OF CAPACITY ACHIEVING SUPERINPUTS AND THEIR TRANSITION PROBABILITIES IN CASE n = 1,2.

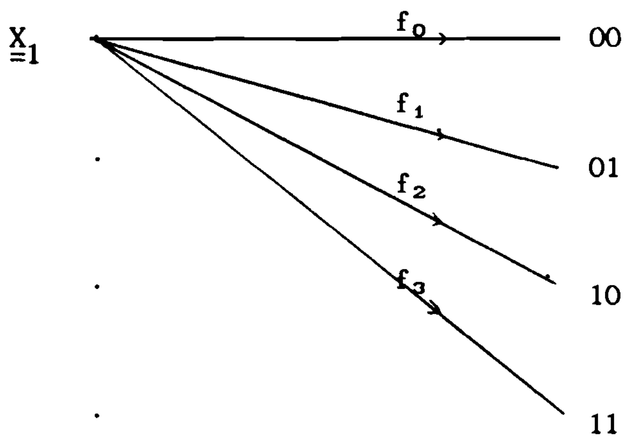
Both in case $n = 1$ and $n = 2$ we note the superinputs $\underline{X}$ respectively $\underline{\underline{X}}$ as columns instead of rows. For $n = 1$ the superchannel with capacity achieving superinputs is depicted below.

$$\underline{X}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \underline{X}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

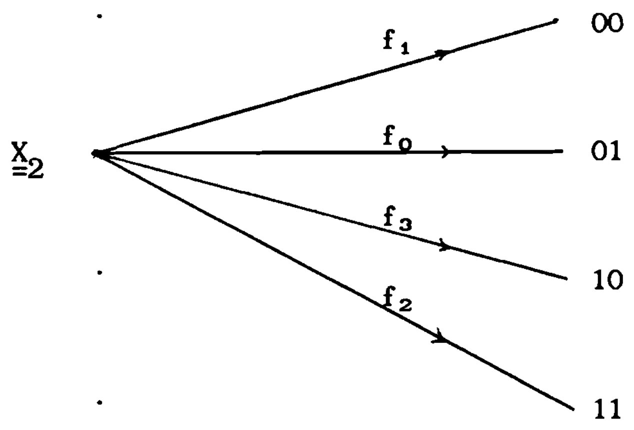


To get a better overall picture the scheme of the superchannel in case $n = 2$ is split in four parts. Each of them corresponding with one of the four superinputs.

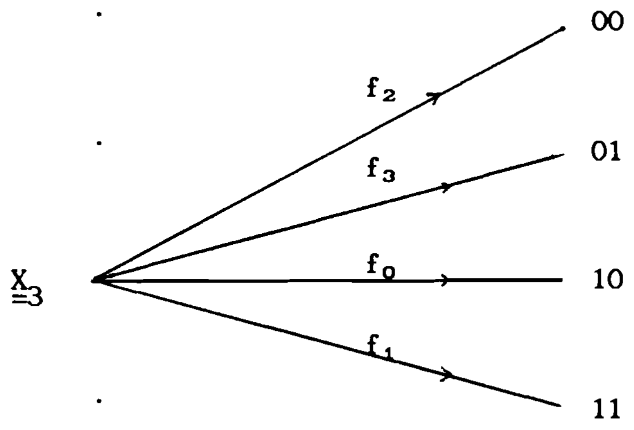
$$\underline{\underline{X}}_1 = \begin{bmatrix} 00 \\ 00 \\ 11 \\ 00 \\ 11 \\ 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$$



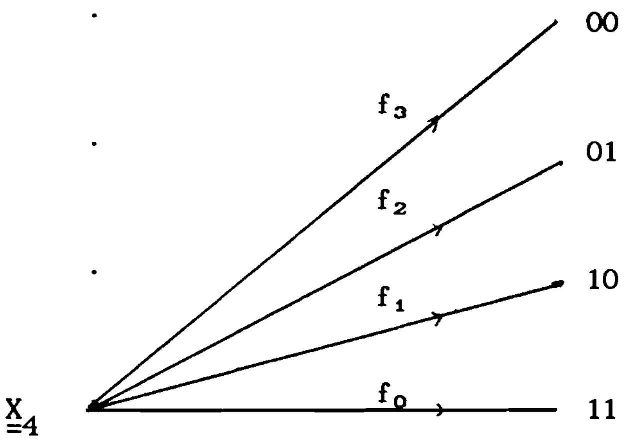
$$\underline{\underline{X}}_2 = \begin{bmatrix} 01 \\ 10 \\ 01 \\ 01 \\ 10 \\ 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$$



$$\underline{\underline{X}}_3 = \begin{bmatrix} 10 \\ 10 \\ 01 \\ 01 \\ 10 \\ 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$$



$$\underline{\underline{X}}_4 = \begin{bmatrix} 11 \\ 00 \\ 11 \\ 00 \\ 11 \\ 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$$



Where: $f_0 \triangleq (1 - p/2)^2$

$f_1 \triangleq p^2/4$

$f_2 \triangleq p^2/4$

$f_3 \triangleq (1 - p/2)^2 - (1 - p)^2$