Multiple-repetition coding for channels with feedback

Veugen, P.J.M.

DOI:
10.6100/IR495939

Published: 01/01/1997

Citation for published version (APA):
Multiple-repetition coding for channels with feedback

Thijs Veugen
Multiple-repetition coding
for
channels with feedback

PROEFSCHRIFT
ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van
de Rector Magnificus, prof.dr. M. Rem, voor
een commissie aangewezen door het College van
Dekanen in het openbaar te verdedigen op

maandag 23 juni 1997 om 16.00 uur

door

Thijs Veugen

geboren te Weert
Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. J.P.M. Schalkwijk
en
prof.dr. K.Š. Zigangirov
This thesis is concerned with a specific class of coding schemes for discrete memoryless channels with noiseless feedback, namely multiple-repetition feedback strategies. Multiple-repetition feedback strategies are easily implemented by low complexity encoders and decoders, and achieve high transmission rates combined with low decoding error probabilities. In Chapter 1 an overview is presented of known results on coding for discrete memoryless channels with noiseless feedback. Most known coding schemes are non-constructive, or have a high complexity. It is also shown how multiple-repetition feedback strategies originated.

Chapter 2 deals with multiple-repetition block coding strategies. A tail is necessarily appended to the message to optimize the error-correcting capabilities. The maximal error-correcting capability can only be guaranteed when the tail is suitably chosen. The repetition parameters determine whether a certain tail is suitable or not. A new tail construction is presented which broadens the formerly known class of allowable tail constructions.

Recursive multiple-repetition coding strategies are considered in Chapter 3. Existing methods for the BSC are generalized to DMC’s with arbitrary channel error probabilities. Two type of decoding errors are considered, and one of them is conjectured to determine the asymptotical error behaviour.

In Chapter 4 the transmission rate of multiple-repetition strategies is computed. It is shown that the transmission rate is independent of the type of coding, i.e. block coding or recursive coding, and that the transmission rate equals the channel capacity for certain channel error probabilities. For given repetition parameters, the channel error probabilities of the DMC can be computed for which the multiple-repetition feedback strategy achieves channel capacity. This result was only known for symmetric DMC’s, and is generalized to DMC’s with arbitrary channel error probabilities.

It is shown in Chapter 5 that for a given DMC with arbitrary channel error probabilities, the repetition parameters should be chosen close to the (absolute value of the) logarithm of the channel error probabilities to maximize the transmission rate. When the repetition parameters are chosen in this way, the transmission rate will approach the channel capacity up to a difference in the order of magnitude of the channel error probabilities.

Block coding strategies that are able to correct channel errors up to a certain maximal fraction are unsuitable for a BSC with high channel error probability. By modifying multiple-repetition block coding strategies as is done in Chapter 6, new strategies arise that perform well for high channel error probability. The new strategies require less than one bit feedback per transmission.

Several applications can be thought of where multiple-repetition feedback strategies could be used. As is illustrated in Chapter 7, the possibilities are not restricted to the actual transmission of information through noisy channels. One can also think of subjects as coding for defective memories, achieving equilibrium in an economic market, and stochastic estimation methods.

Finally, in Chapter 8 an overview is given of the papers that contain the main results on multiple-repetition feedback coding, including the new results by the author. Some suggestions for further research are mentioned also.
Contents

1 Development of repetition feedback coding 4
  1.1 Introduction ......................................................... 4
  1.2 Literature overview of coding for discrete memoryless channels with noiseless feedback ........................................ 7
    1.2.1 Interesting results without feedback ...................... 8
    1.2.2 Decision feedback .......................................... 12
    1.2.3 Information feedback .................................... 16
    1.2.4 Ulam’s game ................................................. 20
  1.3 Horstein .......................................................... 21
    1.3.1 Introduction .................................................... 21
    1.3.2 Encoding ..................................................... 22
    1.3.3 Decoding ..................................................... 23
    1.3.4 Performance .................................................. 24
    1.3.5 Finite message sequence .................................. 24
    1.3.6 Complexity ................................................... 24
    1.3.7 Error exponent ............................................. 25
    1.3.8 Arbitrary transmission rate ................................ 26
    1.3.9 Conclusions .................................................. 28
  1.4 Multiple-repetition strategies 28
    1.4.1 Schalkwijk .................................................. 28
    1.4.2 Schalkwijk and Post ....................................... 34
    1.4.3 Becker ....................................................... 38
    1.4.4 Veugen ....................................................... 39
  1.5 Conclusions ...................................................... 41

2 Block coding 42
  2.1 Binary symmetric block coding .................................. 42
    2.1.1 Error correcting capabilities ............................ 43
    2.1.2 Proof ....................................................... 44
  2.2 Block coding for arbitrary channels ............................ 46
    2.2.1 Error correcting capabilities ................................ 48
    2.2.2 Proof of Tail Condition 2 ................................ 49
  2.3 Conclusions ...................................................... 54
3 Recursive coding
3.1 Binary symmetric recursive coding
3.1.1 Fixed decoding delay
3.1.2 Variable decoding delay
3.2 Recursive coding for arbitrary channels
3.2.1 Comparison with the binary symmetric estimation algorithm
3.2.2 Asymptotic error exponent
3.3 Conclusions

4 Achieving channel capacity
4.1 The transmission rate
4.2 When is capacity achieved?
4.3 Proof of Theorem 4.1
4.4 Examples
4.4.1 Binary symmetric channel
4.4.2 Z-channel
4.4.3 A "super" channel
4.5 Conclusions

5 Choosing the parameters
5.1 Binary symmetric channel
5.2 Uniform channels
5.3 Arbitrary channels
5.3.1 Repetition parameters
5.3.2 Symbol precoding distribution
5.3.3 Example
5.4 Conclusions

6 Variations and extensions
6.1 Generalized scheme
6.1.1 Description
6.1.2 Block error exponent
6.2 Alternative schemes
6.2.1 Majority decision
6.2.2 Collective
6.2.3 Consecutive
6.3 Conclusions

7 Practical aspects and possible applications
7.1 Further research
7.2 Precoding
7.2.1 Enumerative coding
7.2.2 Median approximation
7.2.3 Regular grammar
7.2.4 Prefix-synchronized codes
7.2.5 Dummy reversal
7.2.6 Sliding block code
7.3 Practical applications
7.3.1 Semi-conductor memories ........................................... 108
7.3.2 Economic markets ................................................. 108
7.3.3 Channels with delayed feedback ................................. 110
7.3.4 Memories with known defects .................................... 110
7.3.5 Duplex channels .................................................. 113
7.3.6 Statistical experiments ........................................... 115
7.3.7 Other applications ................................................ 116
7.4 Conclusions .......................................................... 117

8 Literature and open problems .................................. 118
8.1 Literature ............................................................ 118
8.2 Open problems ...................................................... 119
8.3 Conclusions .......................................................... 122

Bibliography ............................................................... 123

A Asymptotic error exponents ........................................ 131
A.1 Asymptotic exponent of $p_n^a$ .................................. 131
A.2 Asymptotic exponent of the alternative estimation error 132

B Binary precoding rate for increasing repetition parameters 134

C Decreasing capacities .................................................. 136

Acknowledgements ....................................................... 137
Notation ........................................................................ 138
Index ............................................................................ 141

Samenvatting .................................................................. 144

Curriculum Vitae ........................................................... 145
Chapter 1

Development of repetition feedback coding

In this Chapter an overview is presented of the known results on coding for discrete memoryless channels, especially coding for discrete memoryless channels with a noiseless feedback link. We briefly explain the results. The reader who is interested in a particular result is referred to the corresponding paper or to one of the interesting textbooks [4, 110, 23, 35, 48]. After having given this general background we describe Horstein's scheme [45] and show how the multiple-repetition feedback coding strategies originated from it. The rest of this thesis elaborates on the class of multiple-repetition feedback coding strategies.

1.1 Introduction

The communication model for transmitting information over noisy channels that is used by information theorists was invented by Shannon in 1948. In part II of [81] Shannon introduces the discrete channel with noise and the notion of channel capacity, and in part IV of [82] Shannon introduces the continuous channel. We are only interested in discrete memoryless channels.

Definition 1.1 A discrete memoryless channel (DMC) is a channel with a finite input alphabet $\mathcal{X}$, a finite output alphabet $\mathcal{Y}$ and channel probabilities $p_{xy} = \Pr\{\text{output} = y \mid \text{input} = x\}$ that only depend on $x$ and $y$ ($x \in \mathcal{X}, y \in \mathcal{Y}$).

A well known example of a DMC is the binary symmetric channel (BSC) as depicted in Figure 1.1, where the probability of a channel error is $p$. The capacity $C$ of a channel is

![Figure 1.1: The binary symmetric channel](image-url)
the maximum amount of information that can be transmitted over the channel in one use. Shannon showed that besides this operational interpretation of channel capacity an explicit formula in information theoretic terms can be given:

\[ C = \max_{Q(x), x \in \mathcal{X}} I(X; Y). \] (1.1)

In words, \( C \) is the largest average mutual information, maximized over all input probability assignments. For a BSC the capacity is \( C = 1 - h(p) \), where \( h \) is the binary entropy function.

The capacity of a DMC can be achieved by block coding. Suppose we have a set \( \mathcal{M} \) consisting of \( M \) messages. Each message \( m \in \mathcal{M} \) that has to be transmitted is encoded to a sequence of \( N \) channel input symbols, i.e. a code word. The number \( N \) is called the block length. The encoding function is denoted by \( f_N : \mathcal{M} \rightarrow \mathcal{X}^N \). The set \( \{ f_N(m) \mid m \in \mathcal{M} \} \) is called the code. A transmitted code word \( X = f_N(m) \) is received as a sequence of \( N \) channel output symbols \( Y \). The decoder then assigns a guess to the received sequence according to a decoding function \( g_N : \mathcal{Y}^N \rightarrow \mathcal{M} \). The average probability of error is defined by

\[ P_e^{(N)} = \frac{1}{M} \sum_{m \in \mathcal{M}} \Pr\{g_N(Y) \neq m \mid X = f_N(m)\}. \] (1.2)

This denotes the probability of a wrong decoder guess when the messages are uniformly chosen. The rate of a code is defined by

\[ R = \frac{\log M}{N}, \] (1.3)

which equals the amount of information that is sent over the channel per transmission (i.e. per channel use). Note that throughout this thesis we take the logarithm to the base \( |\mathcal{X}| \).

**Definition 1.2** A rate \( R \) is said to be achievable if there exists a sequence of codes with increasing block length of rate (at least) \( R \) such that \( P_e^{(N)} \) tends to zero as \( N \to \infty \).

This leads to the statement of one of the most important theorems of information theory.

**Theorem 1.1 (Shannon's channel coding theorem)** All rates \( R < C \) are achievable. Conversely, if rate \( R \) is achievable, then \( R \leq C \).

In proving the achievability of rates below capacity, Shannon used the revolutionary idea of random coding. Shannon calculated the average of \( P_e^{(N)} \) over a random choice of codes and used that to show the existence of at least one good code. Driven by Theorem 1.1 scientists went looking for good codes which led to the development of the theory of error correcting codes [8].

In 1955 Elias [29] discovered that the capacity of a channel can also be achieved by convolutional codes. As opposed to block coding, the code word \( f_N(m) \) does not only depend on the message \( m \), but also on a fixed number of previous messages, i.e. the encoder has a memory. The idea of convolutional coding is illustrated by a simple example depicted in Figure 1.2. We assume we have an infinite sequence of binary messages (\( \mathcal{M} = \{0, 1\} \)) and that the shift register initially contains three zeros. Each message, or source bit, is shifted in the register from the left and moves after each encoding step one place to the right. In each encoding step two channel input bits are generated (in comparison with a block code one could say that the block length is two). The first channel input bit is the modulo two sum of the three bits in the shift register. The second channel input bit is the modulo two
sum of the first and the last bit in the shift register. Since for every source bit there are two channel input bits, the code's bitrate is 1/2. Since the encoder has to remember two source bits in addition to the current source bit, the code is said to have memory two. Each source bit stays in the shift register for three time units and thus one source bit influences $3 \cdot 2 = 6$ channel input bits. This number is called the constraint length. The constraint length of a convolutional code plays a similar role to that of the block length of a block code. Several methods have been devised for decoding convolutional codes. The interested reader is referred to [57].

Thus far coding schemes were considered such that the probability of erroneous decoding tends to zero for increasing block or constraint length. The idea of complete error free decoding (i.e. zero probability of erroneous decoding) led Shannon [86] in 1956 to the introduction of zero error capacity. In determining the zero error capacity $C_0$ of a DMC it is important to know which input symbols are adjacent. Two input symbols are called adjacent when there is an output symbol that can be caused (with positive probability) by both input symbols. It is not difficult to see that $C_0 = 0$ if all input symbols are adjacent. Shannon also considered the DMC with complete feedback (DMCF) which means that there exists a return channel sending back from the receiving point to the transmitting point, without error and without delay, the symbols actually received.

**Theorem 1.2 (Shannon)** The capacity of a DMCF is equal to the ordinary capacity $C$ (without feedback).

While the capacity of a DMC is not increased by feedback, the zero error capacity may be increased by feedback. Although it is difficult to determine $C_0$ for the general DMC, Shannon was able to determine the zero error capacity $C_{of}$ of a DMCF.

**Theorem 1.3 (Shannon)** The zero error capacity $C_{of}$ of a DMCF is zero if all pairs of input symbols are adjacent. Otherwise $C_{of} = \log P_0$ where

$$P_0 = \min_{Q(x) \in \mathcal{X}} \max_{y \in \mathcal{Y}} \sum_{x \in X_y} Q(x),$$

(1.4)

$Q(x)$ being a probability assigned to input symbol $x$ ($\sum_{x \in \mathcal{X}} Q(x) = 1$) and $X_y$ the set of input symbols which can cause output symbol $y$ with probability greater than zero.
Shannon's channel coding theorem states that data can be transmitted over the channel in appropriately coded form at any rate less than channel capacity with arbitrarily small error probability. However, in general the error probability can be made small only by making the coding constraint length (or block length) large. This, in turn, introduces complexity into the encoder and decoder. Thus, if one wishes to employ coding on a particular channel, it is of interest to know not only the capacity but also how quickly the error probability can be made to approach zero with increasing constraint length. Elias [29] was the first to show that the best coding schemes exhibit a decrease in error probability that is exponential in the block length. Therefore it makes sense to define the error exponent $E$.

**Definition 1.3** Let $P_e(N)$ be the (average) probability of decoding error for a particular coding scheme with fixed (bit)rate and given block length (or constraint length) $N$. Then the error exponent $E$ is defined by

$$E = \lim_{N \to \infty} \frac{-\log P_e(N)}{N}.$$  

(1.5)

Note that for all coding schemes that are considered in this thesis, the limit in Equation 1.5 does exist. It follows from Equation 1.5 that for increasing block length $P_e(N) \approx (|\mathcal{X}|)^{-E \cdot N}$. When the transmission rate is fixed, one would like to have a coding scheme that yields the largest error exponent. The error exponent $E$ as a function of the transmission rate $R$ is often called the reliability function. Several bounds were derived for the maximal achievable error exponent as we will see in the next Section.

### 1.2 Literature overview of coding for discrete memoryless channels with noiseless feedback

We give a brief overview of the most important papers up to now in the area of coding for discrete memoryless channels with a noiseless feedback link. The reader who is interested in results on coding for continuous channels with feedback is referred to a survey paper by Schalkwijk [68]. A limited overview of the area of channel coding is also found in a report by Darlington, Felsen, and Siegel [24, p. 1374], and a survey paper by Viterbi [108]. Before we discuss the particular results some general remarks on coding schemes for DMC's with feedback are given, which are verified for the BSC by Weldon [109] in his Ph.D. thesis of 1963.

As mentioned in Theorem 1.2 the capacity of a DMC is not enlarged by feedback. However, it is useful to study DMC's with feedback since the complexity of the coding schemes can be reduced and larger error exponents can be achieved. Also, by use of the feedback link the receiver can inform the transmitter of the status of the decoding process which enables a variable length coding scheme: when many errors occur on the channel the receiver postpones its decision and the transmitter sends additional information.

As an example, suppose we have a BSC and two messages: 0 and 1. Assume we would like to send the message 0. We could use a block coding scheme with block length $N$ and code $\{0^N, 1^N\}$ and transmit $0^N$. The decoder makes a majority decision on the received sequence to decide what message was transmitted (say the receiver takes a random guess when the number of received zeros equals the number of received ones). We could also use a variable length coding scheme where the transmitter keeps sending a zero until a fixed number of zeros is received. Another option would be to send zeros until the a posteriori
error probability, which is the probability that the receiver makes an erroneous decision given the received sequence, drops below a certain threshold. Although variable length coding schemes in general achieve higher error exponents than fixed length coding schemes, the disadvantage of variable length coding schemes is that it is not known in advance how long the transmission of a message is going to take. An intermediate solution would be to introduce a maximum number of allowed transmissions, so that the receiver is forced to make a decision when the transmission of a message takes too much time.

Another property that can be used to distinguish between different coding schemes is the finiteness of the information stream. E.g. for the convolutional code in Figure 1.2 we assumed an infinite sequence of binary messages whereas for block coding the information stream consisted of one message selected from a finite set. Although a convolutional code can be terminated (which is of course necessary for practical applications), the assumption of an infinite information stream makes a theoretical analysis more convenient. Coding schemes that assume an infinite information stream are called sequential coding schemes or recurrent coding schemes. Due to the nature of such schemes the decoding is also done sequentially, which means that the received symbols are decoded one by one with some delay. The delay is the difference between the moment at which a symbol is received and the time instant at which it is decoded. The delay in sequential coding systems plays a similar role as the block length in block coding systems. Similar to the example above, the delay can be fixed, variable or bounded.

We also mention that the variability of the constraint length of a coding scheme is an important issue. Remember that the constraint length is the number of channel input symbols (or actually transmitted symbols) that is influenced by one source (or information) symbol. In general, coding schemes that have a fixed constraint length attain lower error exponents than coding schemes where the constraint length is a random variable (and usually depends on the channel noise). This remark agrees with the former statement about variable length coding schemes which dealt with the total length of a coding system (i.e. the number of channel symbols for all source symbols).

Coding schemes for DMC’s with feedback can also be classified by the kind of feedback link. Thus far we assumed that all received symbols are fed back to the transmitter via a noiseless and delayless feedback link. In that case the transmitter has full knowledge of the received sequence and consequently of all channel errors that occurred during transmission of the message. This type of feedback is called information feedback. Another type of feedback that is often used in practice (see e.g. [80, 15]) is decision feedback. In decision feedback systems the receiver sends one bit back to the transmitter after a fixed amount of transmissions to inform the transmitter about the status of the decoding process. Usually, the receiver either asks for a repetition of the last block, or acknowledges the receipt of the last block. That is why these systems are also called repeat request systems or automatic repeat request (ARQ) systems.

The results that are going to be discussed are classified according to the criteria mentioned above. Most results are derived by some sort of random coding argument. There are only a few constructive coding schemes.

1.2.1 Interesting results without feedback

Forney [34] showed that the error exponents of convolutional codes and block codes are related by a graphical construction, called the concatenation construction. Suppose we have a convolutional code that achieves an error exponent $e(r)$ at rate $r$. By terminating the
convolutional code we obtain a block code that can achieve an error exponent $E(R)$ at rate $R$. Forney showed that

$$E(R) = \max_r (1 - R/r) e(r).$$  \hfill (1.6)

An example of the concatenation construction for a BSC is depicted in Figure 1.3. The exponent $E(R)$ is the random coding bound (see Theorem 1.4) for block coding, and the exponent $e(r)$ is the random coding bound [34] for convolutional coding. From the random coding argument, it follows that there exist codes that indeed achieve these bounds.

The construction of $E(R)$ from $e(r)$ is described as follows. Suppose we have given the curve $e(r)$. Draw all straight lines from $(0, e(r))$ to $(r, 0)$ for $r$ ranging between 0 and 1. Every point on the resulting family of straight lines gives a realizable combination of $E$ and $R$, so $E(R)$ is the least upper bound to all the straight lines.

The construction of $e(r)$ from $E(R)$ is called the inverse concatenation construction and is described as follows. Suppose we have given the curve $E(R)$. Draw all tangents to $E(R)$ and complete the rectangles of which the tangents are the diagonals. Finally, connect the points at the upper right corners of the rectangles. This results in the curve $e(r)$.

Also depicted in Figure 1.3 are the critical rate $R_{\text{crit}}$ and the computation cutoff rate $r_{\text{comp}}$. In general, for sequential coding schemes, for rates less than $r_{\text{comp}}$, the average number of decoding computations does not grow exponentially with the constraint length, but algebraically. In other words, it is difficult to find good sequential coding schemes for rates above $r_{\text{comp}}$. The same phenomenon is seen for block coding schemes. The convolutional rate $r_{\text{comp}}$ relates by the concatenation construction to a block rate $R_{\text{crit}}$. In general it is difficult to obtain good block coding schemes for rates above $R_{\text{crit}}$. The concept of critical rate and computation cutoff rate will be further addressed in the next paragraphs. It also follows from Figure 1.3 that truncation of a convolutional code causes a performance loss in terms of error exponent.
Block coding

The bounds on error exponents that are presented in this paragraph are also found in paragraph 2.5 of Csiszar and Körner [23], together with some results on constant composition codes.

In 1955 Elias [29] was the first who presented bounds on the maximal achievable error exponent $E(R)$. Elias only considered the BSC with channel error probability $p$. A lower bound was based on a random coding argument.

**Theorem 1.4 (Elias’ random coding bound)** Let $\rho$, $0 \leq \rho \leq 1/2$, be such that $R = 1 - h(\rho)$. Let $R_{crit} = 1 - h(\sqrt{p}/(\sqrt{p} + \sqrt{1 - p}))$. Then $E(R) \geq E_{r}(R)$, where

$$E_{r}(R) = R - C + (p - \rho) \log \frac{p}{1-p}$$

(1.7) if $R \geq R_{crit}$

$$R_{crit} - R + E_{r}(R_{crit})$$

(1.8) if $R < R_{crit}$.

Elias also proved a famous upper bound.

**Theorem 1.5 (Elias’ sphere packing bound)** Let $\rho$, $0 \leq \rho \leq 1/2$, be such that $R = 1 - h(\rho)$. Then $E(R) \leq E_{sp}(R)$, where

$$E_{sp}(R) = R - C + (p - \rho) \log \frac{p}{1-p}.$$  

(1.9)

So the best achievable error exponent for block codes on a BSC is known for rates above the critical rate. Note that $E_{sp}(R)$ equals the binary informational divergence $D(p || p)$. Elias showed that these bounds also hold when restricted to the class of linear codes.

Two years later Shannon [87] gave asymptotic bounds for block codes with a fixed distribution of mutual information per letter.

In Chapter nine of his book [31] of 1961 Fano presented bounds for fixed composition block codes for DMC. By optimizing the parameters Fano was able to generalize Elias' bounds for the BSC to arbitrary DMC’s. Fano also considered block codes with average composition constraints.

In 1965 Gallager [36] derived the same bounds as Fano, but in a simpler way with an easy presentation.

**Theorem 1.6 (Gallager)** Let $E(R)$ be the maximal achievable error exponent at rate $R$. Then $E_{r}(R) \leq E(R) \leq E_{sp}(R)$, where

$$E_{r}(R) = \max_{0 \leq \rho \leq 1} [-\rho \cdot R + \max_{Q} E_{0}(\rho, Q)],$$

(1.10)

$$E_{sp}(R) = \sup_{0 < \rho < \infty} [-\rho \cdot R + \max_{Q} E_{0}(\rho, Q)],$$

(1.11)

and

$$E_{0}(\rho, Q) = - \log \sum_{y \in Y} \left( \sum_{x \in X} Q(x) \cdot p_{xy}^{1/[1+\rho]} \right)^{1+\rho}.$$  

(1.12)

Gallager showed that the random coding bound $E_{r}(R)$ is positive, continuous and convex downward. The critical rate is defined as the infimum of $R$ values for which the slope of $E_{sp}(R)$ is not less than $-1$. For rates above $R_{crit}$ we then have $E_{r}(R) = E_{sp}(R)$, which determines the best achievable error exponent for that range. Note that for a BSC we have $R_{crit} = 1 - h(\sqrt{p}/(\sqrt{p} + \sqrt{1 - p}))$. Besides considering very noisy channels and parallel channels, Gallager also presented an improvement of the random coding bound for low rates. The idea was to expurgate those code words for which the error probability is high.
Theorem 1.7 (Gallager's expurgated bound) There exist codes that achieve error exponent $E_{ex}$ given by

$$E_{ex}(R) = \sup_{p \geq 1, Q} \left\{ -p \cdot R + E_x(p, Q) \right\},$$

where

$$E_x(p, Q) = -p \log \left( \sum_{x, x' \in X} Q(x)Q(x') \left[ \sum_{y \in Y} \sqrt{p_{xy}p_{x'y}} \right]^{1/p} \right).$$

Shannon, Gallager, and Berlekamp [84, 85] rigorously prove Fano's sphere packing bound. They also present a new bound, the straight line bound, which is an improvement of the sphere packing bound for low rates.

Theorem 1.8 (Shannon, Gallager, and Berlekamp) Let $R \geq 0$. Then $E(\lambda R) \leq E_{sl}(\lambda R)$ for $0 \leq \lambda \leq 1$, where

$$E_{sl}(\lambda R) = (1 - \lambda)E_{ex}(0) + \lambda E_{sp}(R).$$

Note that $E_{ex}(0)$ is Gallager's expurgated bound where $R$ tends to 0. The supremum in Equation 1.13 is then obtained for $\rho = 1$. Due to the straight line bound, the value $E'(0)$ was determined.

In 1978 Berlekamp, McEliece, and van Tilborg [9] proved that the general decoding problem for linear codes, as well as the general problem of finding the weights of a linear code, is NP-complete. This suggests that a polynomial time decoding algorithm does not exist. In other words, the decoding of linear codes is difficult.

Convolutional coding

In 1963 Reiffen [63] concluded that the computation cutoff rate for sequential decoding equals the random coding bound for block codes at zero rate.

$$r_{comp} = E_{r}(0)$$

Note that the maximum in Equation 1.10 when $R = 0$ is attained for $\rho = 1$. For a BSC, $r_{comp} = -\log(\frac{1}{2} + \sqrt{p(1-p)})$. Reiffen also introduced the concept of very noisy channels.

Definition 1.4 A DMC is called very noisy when, given a channel input probability distribution,

$$\frac{\text{Pr}\{Y = y\} - p_{xy}}{\text{Pr}\{Y = y\}} = \epsilon_{xy} \ll 1$$

for all $x \in X$ and $y \in Y$.

Reiffen showed that for a very noisy DMC, $r_{comp} = C/2$, and that an input probability distribution that achieves $C$ also maximizes $r_{comp}$ and vice versa.

In 1967 Viterbi [106] invented a decoding algorithm which was later called the Viterbi algorithm. This algorithm is particularly effective for short constraint lengths and has found numerous applications. Viterbi gave an analogue for convolutional codes of the sphere packing bound for block codes which is exactly the inverse concatenation construction of $E_{sp}$. Similarly for the straight line bound $E_{x}$. Viterbi showed that the convolutional sphere packing bound was optimal for rates above $r_{comp}$ and that the Viterbi algorithm attains this bound. For very noisy channels Viterbi determined the optimal error exponent.
Theorem 1.9 (Viterbi) For a very noisy DMC the best achievable error exponent $e(r)$ by convolutional codes is

$$e(r) = \begin{cases} 
C/2, & \text{if } 0 \leq r \leq C/2 \\
C - r, & \text{if } C/2 \leq r < C 
\end{cases}$$  \hspace{1cm} (1.18) \hspace{1cm} (1.19)

1.2.2 Decision feedback

Since erasure decoding is important when decision feedback is available, we first explain some decoding methods. They are depicted in Figure 1.4. After receiving a message the decoder has to decide which message was sent. Usually each received sequence is decoded to one particular message, but other methods are possible. Three different methods are distinguished by considering the space of observations. The first one is maximum likelihood decoding.
decoding: the receiver chooses the a posteriori most likely message. The second one is erasure decoding: each possible message has a disjoint decision region, but when the received message falls outside all decision regions, no decision is made. When decision feedback is available, this provides the possibility of repeating the message. The third method is list decoding: the decision regions overlap and the decoder generates a list of alternatives corresponding to the regions in whose intersection the received sequence lies.

Weldon [109] considered decision feedback strategies for the BSC. He considers linear codes of fixed, bounded and variable length combined with fixed or variable information content. From a bound for fixed length erasure decoding for the BSC without feedback, Weldon derives an achievable error exponent for codes for the BSC with decision feedback by letting the transmitter repeat non-decoded words.

**Theorem 1.10 (Weldon)** When considering linear codes for the BSC with decision feedback, the error exponent

\[ E_w(R) = C - R \]  

(1.20)

can be achieved.

Since in practice channel errors tend to occur in bursts, Weldon’s [109, p. 87] following remark is important.

For burst-error channels, that is, channels with finite memory, detect-repeat coding schemes seem inherently better than variable-length schemes since in the former case we discard most words containing bursts rather than attempt to add enough redundancy to correct them.

Weldon [109, p. 89] also notes that it is important to determine the weight distribution of a linear code in order to obtain better bounds on the error probability, but, as mentioned before, this was shown to be infeasible in 1978 [9].

In 1970 Bluestein [14] presents a variable length sequential coding scheme. Using a random coding argument Bluestein showed that the scheme achieves the same exponent as was derived by means of random block coding by Forney [33] two years earlier. However, Bluestein uses the feedback channel more heavily than Forney did and therefore Bluestein’s scheme cannot be considered as decision feedback.

**Block coding**

In 1968 Forney [33] derives exponential bounds on the error probability for erasure and list decoding. The bound for erasure decoding can also be used for block coding on DMC’s with decision feedback.

**Theorem 1.11 (Forney)** When using block codes for DMC’s with decision feedback, the error exponent

\[ E_f(R) = \max_{\rho \geq 1, Q} \left[ -\rho \cdot R + E_{0f}(\rho, Q) \right] \]  

(1.21)

is achievable, where

\[ E_{0f}(\rho, Q) = \sum_{x \in X, y \in Y} Q(x)p_{xy} \left[ \log p_{xy} - \log \left( \sum_{x' \in X} Q(x')p_{x'y}^{1/\rho} \right) \right]. \]  

(1.22)
A similar result was obtained by Csiszár and Körner [23, p. 201]. Viterbi and Gallager had shown earlier in unpublished work that \( E_f(R) \geq C - R \), so Forney’s exponent applied to the BSC is better than Weldon’s exponent. The exponent \( E_f(R) \) approaches 0 as \( R \) approaches capacity with a slope of \(-1\), in contrast to the error exponent for DMC’s without feedback, whose slope is generally zero at capacity. In other words, near capacity the achievement of low error probabilities is dramatically simplified by decision feedback. For very noisy channels the formulas become very simple and give a good insight for the relative magnitudes of the exponents.

**Theorem 1.12 (Forney)** For very noisy DMC’s, \( R_{crit} = C/4 \), and the exponents \( E_f \), \( E_{sp} \), and \( E_r \) are given by

\[
E_f(R) = 2C - 2\sqrt{RC} \tag{1.23}
\]

\[
E_{sp}(R) = (\sqrt{C} - \sqrt{R})^2 \tag{1.24}
\]

\[
E_r(R) = \frac{C}{2} - R, \text{ if } R \leq \frac{C}{4} \tag{1.25}
\]

\[
(\sqrt{C} - \sqrt{R})^2, \text{ if } R \geq \frac{C}{4}. \tag{1.26}
\]

So with decision feedback it is even possible to exceed the sphere packing bound. In particular, for both very noisy and totally symmetric DMC’s, Forney shows that

\[
E_f(R) = E_{sp}(R) + C - R. \tag{1.27}
\]

In 1969 Viterbi [107] derives bounds for erasure and list decoding for very noisy DMC’s without feedback and is able to determine the optimal error exponent for very noisy DMC’s with and without decision feedback.

**Theorem 1.13 (Viterbi)** Suppose we have a very noisy DMC. Let \( E(R) \) and \( E_f(R) \) be the best achievable error exponent for block coding without and with decision feedback respectively. Then

\[
E(R) = \begin{cases} 
\frac{C}{2} - R, \text{ if } 0 \leq R < \frac{C}{4} \\
(\sqrt{C} - \sqrt{R})^2, \text{ if } \frac{C}{4} \leq R < C 
\end{cases} \tag{1.28}
\]

\[
E_f(R) = 2C - 2\sqrt{RC}. \tag{1.29}
\]

It follows that the random coding bound is optimal for very noisy channels as well as Forney’s bound for very noisy channels with decision feedback.

For a long time Forney’s exponent \( E_f \) was thought to be optimal for arbitrary DMC’s with decision feedback, until recently Telatar and Gallager [91] contested this statement. They show that their bound is still not tight except at zero rate. From their results follows that their exist coding schemes that achieve \( P_e = 0 \) for the binary Z-channel, which is depicted in Figure 1.5, with decision feedback. This seems to imply that the zero error capacity \( C_{0f} \) of the Z-channel is equal to the ordinary capacity, which would contradict Shannon’s Theorem 1.3 that states that \( C_{0f} = 0 \) for the Z-channel. However, when using error bounds for erasure decoding without feedback to obtain bounds for decision feedback codes, it is assumed that a code word is repeated in case of an erasure, but this could cause an infinitely number of repetitions. In other words, Telatar and Gallager (as well as Weldon, Forney, and Viterbi) allow unbounded variable length coding in contrast to Shannon who only considered fixed length coding.
Convolutional coding

In 1980 Yamamoto and Itoh [112] present a convolutional repeat request scheme with Viterbi decoding. The scheme achieves a reliability that is twice the reliability that was achieved by Viterbi [106] without feedback. Viterbi’s reliability is obtained by the inverse concatenation construction of the random coding bound $E_c(R)$ (see Equation 1.10) for block codes.

Kudryashov in 1984 [52] shows by a random coding argument that convolutional schemes exist with a higher reliability than Yamamoto and Itoh [112]. Kudryashov uses a new skewed decoding rule which enables him to go beyond Forney’s block coding bound (see Equation 1.21). Kudryashov proposes to use his decoding rule in Yamamoto and Itoh’s scheme to obtain an error exponent

$$e_k(R) = E_f(R) + R,$$

(1.31)

where $E_f$ is Forney’s block coding bound.

One year later Kudryashov [53] presents a scheme that consists of an inner code and an outer code. The inner code is a block code and the outer code is a convolutional code that uses repeat request. This scheme has less reliability than [52] but it is more efficient in terms of encoding and decoding complexity.

Recently Kudryashov [54] was able to improve on the result in [52] by introducing a new multiplicative decision rule. The block code that results when Kudryashov’s convolutional code is terminated achieves Forney’s exponent (see Equation 1.21). The exponent $e_f(R)$ of the convolutional code is exactly the result of the inverse concatenation construction of the block code exponent.

Hashimoto [41] compares the different results of Forney [33], Yamamoto and Itoh [112], and Kudryashov [52]. Hashimoto is not aware of Kudryashov’s latest improvement [54], and states that because of the concatenation construction that relates block codes and convolutional codes, it is natural to expect that there exists a convolutional scheme that attains $e_f(R)$. However, Hashimoto states that it is very difficult to apply and analyze Forney’s decision rule in the case of convolutional codes. The same problem arises when analyzing Kudryashov’s scheme [52]. Yamamoto and Itoh’s scheme [112] uses a likelihood ratio test, which is generally believed to be suboptimal, but gives a much simpler retransmission scheme. Hashimoto shows that by modifying Kudryashov’s scheme, the exponent $e_f(R)$ can be achieved and that the reliability of Yamamoto and Itoh’s likelihood ratio test is actually better than stated before, and is close to $e_f(R)$ for low rates.
1.2.3 Information feedback

In 1958 Dobrušin [27] proved, independently from Shannon, that a feedback channel does not increase the capacity of a DMC, regardless of the capacity of the feedback channel. When all received symbols are transmitted back to the transmitter without delay and noise, the transmitter obtains the maximum of useful information. Dobrušin showed that for some simple channels with memory, feedback can increase its capacity. Dobrušin [27, p. 374] says:

The use of feedback does not permit the transmission of a message which cannot be transmitted without using feedback; however, feedback can simplify the method of coding signals, which is used for transmission.

Weldon [109] considered not only linear codes for the BSC with decision feedback, but also with information feedback. Weldon obtains an achievable exponent for truncated variable length, fixed information content codes that is, as expected, higher than its analogue for decision feedback (see Theorem 1.10). According to Weldon [109, p. 88], the feedback enables the transmitter to match the transmission rate quite closely to the channel conditions and to inform the receiver of the particular transmission rate being employed.

Block coding

In [67] Savage says that Shannon showed that the sphere packing bound also holds for DMC’s with feedback that are uniform at the input (for definition see [31, p. 126-127]), and that both Shannon and Berlekamp conjecture that the sphere packing bound holds for block coding on an arbitrary DMCF. The sphere packing bound does not hold for variable length coding schemes, even not in the case of decision feedback (see e.g. Theorem 1.12). Savage [67, p. 968] makes the following interesting remark:

The block-coding strategies, however, are interesting since any improvement in coding reliability observed with them can be ascribed directly to the effect of feedback on the choice of code words representing messages. This is not true of the variable-length strategies since, in this case, some unknown fraction of the improvement in reliability is attributable to the variation of the code length with the level of channel noise.

Savage presents a scheme that consists of two stages. In the first stage a block code is used with list decoding. In the second stage another block code is used to resolve the list. This scheme requires that the complete list of code words is sent back to the transmitter over the feedback channel after the first stage, which makes it unsuitable for many practical applications. The error exponent that is achieved by Savage goes beyond the random coding bound.


Definition 1.5 Given a rate R, correctable error fraction $f$, $0 \leq f \leq 1/2$ is said to be achievable if for arbitrary long block length $N$, there exists a block code with rate $R$ that can correct all channel errors in one block, provided the total number of channel errors does not exceed $f \cdot N$. 

16
Berlekamp's region of achievable pairs $(f, R)$ is depicted in Figure 1.6. It is bounded by the Volume bound $R = 1 - h(f)$, which is a generalization of the Hamming bound [38] without feedback, and by the Tangent bound, a straight line through $(1/3, 0)$ that is tangent to the Volume bound.

We compare Berlekamp's notion of achievability with that of Shannon (see Definition 1.2) for a BSC with channel error probability $p$. Given a rate $R$ and a sequence of codes that achieves correctable error fraction $f$, the average error probability of this sequence of codes tends to zero if and only if $p < f$. So rate $R$ is also achievable in the Shannon sense. The opposite is however not true. It follows that Berlekamp's notion of achievability is more restricting, which explains why Berlekamp's achievable region is contained in the capacity region of the BSC which is bounded by $C = 1 - h(p)$.

Berlekamp presents constructive strategies that achieve all straight lines starting in $(1/k, 0)$ and that are tangent to the Volume bound, for each $k \geq 3$.

Zigangirov [115] considers block coding for the BSC. Using the ideas of [114], where Zigangirov describes variable length coding schemes, Zigangirov obtains an achievable error exponent. Zigangirov also determines the critical rate for the BSC with feedback.

**Theorem 1.14 (Zigangirov)** Consider a BSC with channel error probability $p$. Let $\rho$, $0 \leq \rho \leq 1/2$ and $R$ be related by $R = 1 - h(\rho)$. Define the function

$$f(z, \alpha, \lambda) = p \left( \frac{1 + (1 - 2\alpha)z}{2\alpha} \right)^\lambda + (1 - p) \left( \frac{1 - (1 - 2\alpha)z}{2(1 - \alpha)} \right)^\lambda.$$  

(1.32)

Then the error exponent $E_z(R)$ is achievable by block coding, where

$$E_z(R) = \max_{\lambda > 0, 0 < \alpha < 1/2} \left[ -\lambda \cdot R - \log \max\{f(0, \alpha, \lambda), f(1, \alpha, \lambda)\} \right]$$  

(1.33)

Zigangirov notes that by choosing $\alpha = \rho$ and $\lambda = \lambda(\rho) = -1 + \log \frac{1-\rho}{\rho} / \log \frac{1-\rho}{\rho}$, the right hand side of Equation 1.33 is maximized for $R \geq R_{\text{crit}}^f$ and equals $D(\rho \| p)$, the binary
informational divergence which also equals the binary sphere packing bound. The critical rate $R_{\text{crit}}$ for the BSC with feedback is determined by solving $f(0, p, \lambda(\rho)) = f(1, p, \lambda(\rho))$. It is larger than the critical rate for the BSC without feedback. Furthermore, by suitably choosing $\alpha$ and $\lambda$ as $R$ tends to zero, Zigangirov obtains

$$E_z(0) = -\log(p^{1/3}(1-p)^{2/3} + (1-p)^{1/3}p^{2/3}),$$

which was shown to be optimal in [10]. This result also leads to an improvement of the straight line bound (see Theorem 1.8) for the BSC with feedback by replacing $E_{cz}(0)$ by $E_z(0)$.

In 1971 Ahlswede [2] presents constructive block codes for arbitrary DMC’s with feedback based on an iterative procedure. The codes achieve channel capacity but are not very practical.

Djačkov [26] is able to generalize the results of Zigangirov [115] to the class of symmetrical DMC’s and the DMC’s with binary input.

In 1976 Zigangirov [116] shows that Berlekamp’s Volume bound (see Figure 1.6) is achievable. Note that Berlekamp only showed the achievability of the tangent points.

Two years later Zigangirov [118] gives a rigorous proof (as opposed to Berlekamp [10]) that no block transmission method can go beyond $E_z(0)$ (see Equation 1.34).

In 1979 Yamamoto and Itoh [111] present a scheme for DMC’s with feedback that is based on the Schalkwijk-Barron scheme [71] for the additive white gaussian channel. The scheme works in two different modes. During the message mode a message is sent to the receiver. Due to the feedback link the transmitter knows the decoded message. Subsequently, during the control mode, the transmitter tells the receiver whether the decoded message has to be retransmitted or accepted. The error exponent achieved by Yamamoto and Itoh is:

$$E_{y}(R) = \max_{x, x' \in \mathcal{X}} (1 - \frac{R}{C}) \sum_{y \in \mathcal{Y}} p_{xy} \log \frac{p_{xy}}{p_{x'y}}.$$

Note that the summation in Equation 1.35 is actually the informational divergence between input row $x$ and input row $x'$ of the channel probability matrix. It follows that $E_{y}(R) = \infty$ when there is a channel transition that cannot occur. The exponent of Yamamoto and Itoh exceeds Forney’s decision feedback exponent (see Equation 1.21). The same remark applies here as for most decision feedback block coding schemes, namely that because of the possibly infinite number of retransmissions we are actually dealing with a variable length scheme instead of a block coding scheme. Having realized that, we must assign the discovery of Equation 1.35 to Burnašev [16], who presented a variable length coding scheme achieving that same error exponent already in 1976. In fact, when restricted to the BSC, the error exponent can be found in Burnašev’s Ph.D. thesis of 1974, and in his paper [19] together with Zigangirov in 1975 (see also subsection 7.3.6 of this thesis). Burnašev [16, p. 251-252] shows that the exponent of Equation 1.35 is optimal assuming a Markov instant of decision-making.

Dobrusin [28] and Haroutunian [40] are able to prove the sphere packing bound for symmetrical DMC’s with feedback. Since the random coding bound is equal to the sphere packing bound for rates above the critical rate (without feedback), this determines the optimal exponent for symmetrical DMC’s with feedback at rates above the critical rate. It is not clear whether the sphere packing bound also holds for asymmetrical DMC’s with feedback. An attempt by Ševerdjaev [96] in 1982 failed.
In 1988 the best known achievable error exponent for the BSC was Zigangirov’s $E_2(R)$ (see Equation 1.33). Burnaev [17] is able to improve on this result by a modification. For the exact value of Burnaev’s exponent $E_b(R)$ we refer to the paper, as the formula is complicated and gives little insight. The critical rate derived by Burnaev is equal to Zigangirov’s $R_{crit}$.

Bounded coding delay

In 1971 Fang [30] presents a bounded variable length convolutional coding scheme that achieves an error exponent that equals the channel capacity for all rates less than capacity. This improves Forney’s bound for higher rates.

In 1979 Kudryashov [51] presents a bounded variable length coding scheme with random coding. The scheme uses a large memory. Kudryashov derives asymptotic bounds for the error exponent while fixing the quotient $a$ of the average coding delay divided by the maximal coding delay. By taking the limit of $a \to 0$ this leads to error exponents that are achievable by (unbounded) variable coding delay.

**Theorem 1.15 (Kudryashov)** When considering (unbounded) variable length coding schemes for DMC’s with feedback, the error exponent

$$E_k(R) = \max_{x \in \mathcal{X}, Q \in \mathcal{Q}(\mathcal{X}, \mathcal{Y})} \sum_{y \in \mathcal{Y}} p_{xy} \log(p_{xy} / \sum_{x' \in \mathcal{X}} Q(x') p_{x'y})$$

is achievable, where $Q$ is an arbitrary input probability distribution and $I_Q(X; Y) = \sum_{x \in \mathcal{X}} Q(x) I(X = x; Y)$, the mutual information given input distribution $Q$.

Note that the sum in Equation 1.36 is again an informational divergence. Kudryashov’s exponent is the best known achievable exponent. For the BSC the error exponent reduces to

$$E_k(R) = D(\gamma \| p)$$

$$R = h(\gamma) - h(p),$$

where $1/2 \leq \gamma \leq 1 - p$. Kudryashov notes that the restriction on coding delay is not equivalent to the restriction on peak transmitter power that is generally accepted in investigating continuous channels with feedback. The latter would be equivalent to a restriction on the maximum number of channel symbols expended in transmitting one source symbol (which was introduced as constraint length in this thesis).

In 1991 Han and Sato [39] consider uniformly upper bounded variable length codes for DMC’s with feedback.

**Theorem 1.16 (Han and Sato)** Consider a DMC. Let $C_f^V$ and $C^V_{0f}$ be the capacity and zero error capacity respectively for the class of bounded variable length codes. Then

$$C_f^V = C_f$$

and

$$C^V_{0f} = C, \text{ if } C_0 > 0$$

$$0, \text { if } C_0 = 0.$$
Han and Sato remark that Equation 1.39 remains valid when considering the class of unbounded variable length codes, and also when considering maximal error probability instead of average error probability. For unbounded variable length codes, Equation 1.40 holds under a condition weaker than $C_0 > 0$.

Han and Sato [39, p. 660] observe some duality between source coding and channel coding:

In source coding at the rate equal to the entropy, the fixed-length code is not capable of zero-error compression, whereas, the variable-length code such as Huffman codes can attain the zero-error compression at the same entropy rate. This is just the same situation as we have so far studied for channel coding with variable and fixed length codes.

**Variable coding delay**

In 1968 Zigangirov [114] considers a variable delay coding scheme for the BSC which is similar to Horstein's scheme [45] which is described in detail in Section 1.3.

In 1976 Ulrey [95] considers converses of the channel feedback capacity theorem (see Theorem 1.2). Shannon [86] proved the weak converse for block coding, which means that $P_e^{(N)}$ cannot tend to zero when $R \geq C$. The strong converse means that for any sequence of codes with rate at least capacity, $P_e^{(N)}$ tends to one. The strong converse for block coding on DMC's with feedback was proved independently by Kemperman [49] and Kesten (unpublished). Kemperman's proof is also published in [110, p. 51, 95]. Ulrey shows that the weak converse also holds in the case of sequential coding, and gives a counterexample for the strong converse in the case of sequential coding.

**1.2.4 Ulam's game**

A different approach to the coding problem for DMC's with feedback is Ulam's game [94, p. 281]. This relation is for example described by Ahlswede and Wegener in Chapter nine of [4]. Ulam's game is a game with two players, named Paul and Carole. Carole thinks of an integer $m$ from one to $M$. Paul has $N$ questions with which to determine $m$. The questions must be of the form "Is $m \in A$?", where $A \subseteq \{1, \ldots, M\}$. Paul may use previous answers before deciding his next question. Carole is permitted to lie but she may lie at most $e$ times through the entire course of the game. Paul wins if at the end of the $N$ questions there is a unique possible value for $m$. Otherwise, Carole wins.

For $N = 20$ Ulam's game is also known as Twenty questions with a liar. Usually, $e$ is fixed and small, and $M$ and $N$ are large. Then necessary and sufficient conditions are investigated for Paul to win. The relation with the coding problem for a DMCF becomes clear when considering Carole as a transmitter that is trying to send a message $m$ to the receiver Paul in $N$ transmissions. Due to the feedback link, each time the next question is known to both transmitter and receiver. Their purpose is to find a coding scheme that is able to correct $e$ channel errors.

Since for a DMC, each channel error occurs with a fixed probability, a more suitable version of Ulam's game in that sense would be that Carole lies with a fixed probability $p$, instead of upper bounding the number of lies. In that case Paul has to determine $m$ with an acceptable reliability.
Fixed number of lies

Rivest, Meyer, Kleitman and Spencer [64] show in 1977 that (for increasing \( M \) and \( e \)) \( \log M + e \cdot \log \log M + O(e) \) questions are sufficient for Paul to determine \( m \) and that for any questioning scheme at least \( \log M + e \cdot \log \log M + O(1) \) questions are needed.

Together with Winklmann [65] they show three years later that the optimal number of questions is \( \log M + e \cdot \log \log M + O(e \log e) \), when the only allowed questions are comparisons, i.e. each \( A = \{1, \ldots, a\} \), for some \( a \in \{1, \ldots, M\} \). They also show the relation with a similar problem, of finding the minimum root of a set of increasing functions.

One of the conclusions of Dhagat, Gács and Winkler [25] is that Paul is unable to win when \( e/N \geq 1/3 \). This coincides with Berlekamp’s Tangent bound [11] which is depicted in Figure 1.6.

Spencer [90] explains the names Carole and Paul: Paul may be considered the great questioner - Paul Erdős, and Carole may be thought of as her acronym - Oracle. For \( e \) fixed, and \( M, N \) sufficiently large, Spencer gives necessary and sufficient (asymptotic) conditions for Paul to win.

Hill and Karim [42] determine the optimal number of questions for the case \( M = 2^{20} \) and \( e = 3, 4 \).

Recently, Lawler and Sarkissian [56] presented a near-optimal heuristic algorithm for Ulam’s game, which unfortunately requires a large amount of memory.

Random number of lies

In 1989 Pelc [62] considers a variant of Ulam’s game where only comparison questions are allowed and Carole lies with probability \( p \). For \( p < 1/3 \), Pelc gives an \( O(\log M) \) algorithm for Paul to win. For \( 1/3 \leq p < 1/2 \), the algorithm requires \( O(\log^2 M) \) questions.

Recently, Feige, Raghavan, Peleg and Upfal [32] considered the same variant of Ulam’s game as Pelc [62]. Suppose Paul has to determine \( m \) with reliability \( 1-Q \), then they show that \( O(M/Q) \) questions are sufficient. They also consider parallel algorithms.

1.3 Horstein

One of the early results on coding for DMC’s with feedback was obtained by Horstein [45] in 1963. Horstein presents a sequential coding scheme for the BSC with information feedback that attains high reliability. Since several coding schemes [114, 72, 69, 117, 115, 116, 17] that were constructed later use Horstein’s idea, it is described in detail. We follow Horstein’s text quite closely.

1.3.1 Introduction

Horstein’s scheme uses a variable constraint length \( N \). The expected value of \( N \) is denoted by \( \bar{N} \). A probability of error \( P_e \), as well as a transmission rate \( R \), is prescribed and is related to an average constraint length \( \bar{N}(P_e, R) \). The reliability is then defined by

\[
E_h(R) = \lim_{P_e \rightarrow 0} \frac{-\log P_e}{\bar{N}(P_e, R)}.
\]

Horstein gives an interesting interpretation of \( E_h(R) \), of which the validity was proved by Hofstetter [43]:

\[
21
\]
It can be interpreted as the limiting magnitude, as \( P_e \to 0 \) (and, consequently, as \( N \to \infty \)) of the average information provided per transmission about the incorrect value of a bit. This follows from the fact that the initial probability of the incorrect value is \( 1/2 \) and its probability when the bit is decoded (correctly) is \( P_e \). Since the information provided about a symbol \( u \) by a sequence \( v \) is \( I(u; v) = \log(\Pr\{u \mid v\}/\Pr\{u\}) \), the total information provided about the incorrect value is \( \log 2P_e \), and the limiting magnitude of the average information provided per transmission is

\[
\lim_{P_e \to 0} \left| \log 2P_e \right|/\tilde{N}(P_e, R) = E_h(R).
\]

This interpretation will provide us an easy way to compute the reliability in subsections 1.3.7 and 1.3.8.

### 1.3.2 Encoding

Horstein assumes that the message to be transmitted is an infinite binary sequence of independently and uniformly chosen symbols. If a binary point is placed to the left of the message sequence, the message can be regarded as an (infinite) binary fraction, i.e. a point in the interval \([0, 1]\). A sequence of finite length \( n \) is represented by the subinterval of length \( 2^{-n} \) whose left end-point corresponds to the fraction represented by the sequence. The receiver has no a priori knowledge of the message sequence. Therefore, its initial probability distribution of the location of the point representing the message, which Horstein calls the message point, is uniform over the interval \((0, 1)\). This distribution, which is modified by each received symbol, is referred to as the receiver's distribution.

In Figure 1.7 the encoding procedure is explained. The unit interval is divided in half, as shown in Figure 1.7(a). If the message point lies in the lower half, 0 is the transmitted symbol; otherwise 1 is sent. In the example of Figure 1.7, 0 is transmitted, and also received. The new receiver's probability density is depicted in Figure 1.7(b). Note that \( p \) is the channel error probability of the BSC, and \( q = 1 - p \). For example, the new probability that the
message point is in the interval $(0, 0.5)$, is

\[
\begin{align*}
\Pr\{\text{message point in } (0, 0.5) \mid \text{zero received}\} &= \frac{\Pr\{\text{zero transmitted and received}\}}{\Pr\{\text{zero received}\}} \\
&= \frac{0.5 \cdot q}{0.5 \cdot q + 0.5 \cdot p} \\
&= q, 
\end{align*}
\]

as is shown in Figure 1.7(b). Due to the reception of a zero, the interval $(0, 0.5)$ has been stretched uniformly by a factor $2q$, and the interval $(0.5, 1)$ has been compressed uniformly by a factor $2p$. Consequently, the message points in the interval $(0, 0.5)$ have become more reliable than the message points in the interval $(0.5, 1)$. Note that due to the feedback link, the transmitter is aware of the new receiver’s distribution. To determine the next transmission, the transmitter divides the new interval $(0, 1)$ at its midpoint $M$ (see Figure 1.7(c)) and sends 0 or 1 according to whether the message point is now in the lower or upper half of the interval. In the example of Figure 1.7, the new transmitted symbol is a one. After the second symbol is received, the receiver’s distribution consists of three subintervals. The transmission process is repeated continually as described above. Each transmitted symbol is chosen according to the position of the message point with respect to the midpoint of the interval. The number of subintervals in the receiver’s distribution is always one more than the total number of received symbols. The receiver’s distribution is completely specified by the new positions of those points that have previously been midpoints of the interval $(0, 1)$ (see Figure 1.7(c)).

By the process just described, any subinterval containing the message point is gradually expanded so that its length approaches unity. Any subinterval which does not contain the message point gradually vanishes. However, the situation is different when the considered subinterval contains the midpoint. Then a correct reception does not necessarily lead to an expansion of the subinterval. This is only the case when the message point lies in the larger of the two parts into which the subinterval is divided by the midpoint.

### 1.3.3 Decoding

The receiver becomes increasingly certain about successive bits as additional symbols are received. After each transmission, the lengths of the intervals $(0, 0.5)$ and $(0.5, 1)$ represent the probabilities of a 0 and a 1, respectively, in the first position of the message sequence. A decision about the first bit is made as soon as the length of either of the intervals $(0, 0.5)$ or $(0.5, 1)$ exceeds $1 - P_e$, where $P_e$ is the specified probability of error. Suppose the interval $(0, 0.5)$ reaches this length first. It is immediately expanded to a length of unity and the interval $(0.5, 1)$ is discarded, corresponding to the fact that the receiver’s determination of the first bit at this time is final. The second bit is then determined as soon as the length of either of the intervals $(0, 0.25)$ or $(0.25, 0.5)$ exceeds $1 - P_e$, and so on. The interval that remains after each receiver’s decision is called the transmission interval.

If the receiver should happen to decode a bit incorrectly, the sender immediately changes the bit in question to agree with the altered message sequence. The new message point will then lie in the transmission interval. In this way the occurrence of a decoding error is prevented from disrupting the transmission process.
1.3.4 Performance

The probability that all of the first \( n \) bits will be decoded correctly is at least \((1 - P_e)^n\). We are not entitled to state, however, that in a decoded block of \( n \) bits, the average number that will be in error is \( P_e n \). The reason depends on the fact that each time a decoding decision is made and one of the two possible subintervals is discarded, all remaining probabilities are conditional on the bit that has been chosen. We are therefore guaranteed that the probability of error in decoding a bit is no worse than \( P_e \) only when all preceding decoding decisions have been correct. However, after an error has occurred, \( P_e \) will once again be an accurate measure of the probability of error after an additional number of transmissions on the order of magnitude of the average constraint length. The fact that the actual probability of error is difficult to determine, is a serious drawback of Horstein's scheme.

Because of the uniform probability measure maintained on the transmission interval and the division of the interval into two equal parts, the symbols 0 and 1 are equally likely in the eyes of the receiver on each transmission. Consequently, according to Horstein, each transmitted symbol carries one bit of information about the message sequence. The average information received is just the capacity of the BSC, \( 1 - h(p) \) bit per transmitted symbol. The drawback in this reasoning by Horstein is that it is not absolutely clear that all transmissions are independent, which is a necessary condition.

1.3.5 Finite message sequence

We now take into account the fact that the number of message bits available to the sender at any one time is necessarily finite. Thus with each transmission the sender has in mind an interval, which Horstein calls the message interval, rather than a point. We could use the same encoding procedure as described in subsection 1.3.2, but a problem arises when the midpoint is contained in the message interval. In that case, Horstein chooses the symbol to be transmitted that corresponds to the larger of the two parts into which the midpoint divides the message interval. In trying to prevent this from occurring, Horstein keeps the message interval sufficiently small. The symbol \( d_0, d_0 \ll 1 \), is used to denote the maximum allowed length of the message interval. Whenever the length of the message interval exceeds \( d_0 \), the next message bit is introduced to halve its length.

The constraint length is defined as the number of transmissions required to expand the interval which coincides with the message interval when the bit is introduced, from (slightly less than) \( d_0 \) to \( 1 - P_e \). This initial message interval is referred to as bit interval. The bit interval always includes the message interval. For convenience of the reader, all distinguished intervals and points are recapitulated in Figure 1.8.

1.3.6 Complexity

After each transmission, the receiver's distribution is recomputed by calculating the new position of every former midpoint that has not been discarded by a decoding decision. Since each transmission introduces a new midpoint, the number of former midpoints that must be relocated (and, consequently, the amount of computation required to recompute the receiver's distribution) increases linearly with the constraint length.

Horstein does not mention the problem of numerical precision. Since the midpoints that have to be stored require an increasing accuracy, the memory increases and the computations require more significant numbers.
1.3.7 Error exponent

In calculating the asymptotic behavior of $\bar{N}$, Horstein assumes that no decoding errors are made during the time interval defined by $N$. Furthermore, since the asymptotic behavior dominates the entire process as $P_e \to 0$, Horstein only considers the situation where the length of the bit interval is nearly unity, but less than $1 - P_e$. The length of the bit interval represents the probability that the receiver would assign to the correct value of the bit in question, which is called bit number one. The complement of the bit interval is referred to as the error interval.

The calculation of $E_h$ involves an average over all possible (infinite) message sequences whose first bit agrees with bit number one, as well as an average with respect to the channel noise. Horstein takes the position of an observer who knows the identity of bit number one, but who has no knowledge of any of the following bits. The observer’s probability distribution for the location of the message point is uniform over the bit interval and zero over the error interval. As the length of the error interval approaches zero, the probability that the message point and the error interval lie in the same half of the transmission interval on any transmission approaches $1/2$. The error interval is expanded by a factor $2q$ as the result of a transmission if the message point and the error interval lie in the same half of the transmission interval and the transmitted symbol is correctly received, or if they lie in opposite halves and the received symbol is incorrect. The error interval is therefore expanded with probability $(1/2) \cdot q + (1/2) \cdot p = 1/2$. The other half of the time it is compressed by a factor $2p$. The average information received about the error interval is

$$I = \frac{1}{2} \log 2q + \frac{1}{2} \log 2p = \frac{1}{2} \log 4pq.$$  \hspace{1cm} (1.47)

In subsection 1.3.1 was shown that the error exponent is the limiting magnitude of the average information per transmission about the incorrect value of a bit, so since $4pq < 1$ for all values of $p$,

$$E_h(C) = |I| = \frac{1}{2} \log \frac{1}{4pq}. \hspace{1cm} (1.48)$$

A rigorous proof of Equation 1.48 by Horstein is found in [44].
1.3.8 Arbitrary transmission rate

It is possible to modify Horstein’s scheme so that it can be operated at any rate less than channel capacity by choosing the a priori probabilities of the input symbols to the channel appropriately. For that purpose, prior to each transmission, the transmission interval is divided into a subinterval of size $a$ and a subinterval of size $1-a$, for some $a$, $1/2 \leq a \leq 1$. A one is sent when the message point lies in the subinterval of size $a$, and zero otherwise.

Probably for implementing reasons, Horstein introduces cut positions at $(1-a)/3$ and $(2a+1)/3$, as depicted in Figure 1.9(a). Such a transmission scheme would result in a stationary point at $1/3$. Points initially to one side of $1/3$ would never cross to the other side, regardless of the symbols received and decoding would soon become impossible. Therefore, Horstein uses a modification where on the odd transmissions, the transmission interval is subdivided as in Figure 1.9(a), and on the even transmissions, the transmission interval is subdivided as in Figure 1.9(b). Consequently, the number of cut positions that must be relocated following each transmission has been doubled. Since we see no reason for such a complicated modification, we simply divide the transmission interval by one cut position at position $a$.

Let $\pi$ denote the channel output distribution, then

$$\pi_0 = (1-a)q + ap$$

(1.49)  

$$\pi_1 = aq + (1-a)p.$$  

(1.50)

According to Horstein, the transmission rate equals the average mutual information between the input and the output of the channel, so

$$R(a) = I(X;Y)$$

(1.51)

$$= H(Y) - H(Y \mid X)$$

(1.52)

$$= h(\pi_1) - h(p).$$

(1.53)

The error exponent is computed similarly as in subsection 1.3.7. Consider the limiting situation as the length of the bit interval approaches unity. Suppose the error interval lies in the subinterval of size $1-a$. Then the error interval is expanded by a factor $q/\pi_0$ when
a zero is received, and compressed by a factor $p/\pi_1$ when a one is received. The average information provided about the error interval is then

$$I^0(a) = \pi_0 \log \frac{q}{\pi_0} + \pi_1 \log \frac{p}{\pi_1}. \quad (1.54)$$

The corresponding error exponent is

$$E_h^0(a) = |I_0(a)| = D(\pi_1||p), \quad (1.55)$$

although Horstein was not aware of this elegant formulation.

Now suppose the error interval lies in the subinterval of size $a$. Then the error interval is expanded by a factor $q/\pi_1$ when a one is received, and compressed by a factor $p/\pi_0$ when a zero is received. The average information provided about the error interval is then

$$I^1(a) = \pi_1 \log \frac{q}{\pi_1} + \pi_0 \log \frac{p}{\pi_0}. \quad (1.56)$$

The corresponding error exponent is

$$E_h^1(a) = |I_1(a)| = D(\pi_0||p). \quad (1.57)$$

Since $\pi_1 \geq \pi_0 \geq p$ it is clear that $E_h^0(a) \geq E_h^1(a)$. For rates below capacity even strict inequality holds. The error exponents are depicted in Figure 1.10 for $p = 0.1$. It is interesting to note that $E_h^0$ is exactly the exponent that is achieved by Kudryashov [51], as is shown in Equation 1.37.

Unfortunately, in his analysis Horstein only considers the case where the error interval lies in the subinterval of size $1 - a$. This mistake is also observed by Schalkwijk and Post [72, p. 505].
1.3.9 Conclusions

The entire system, operating at channel capacity with $P_e = 10^{-15}$, was simulated by Horstein on an IBM 709 computer. The simulation results seem to coincide with Equation 1.48. E.g. for $p = 0.1$, $\bar{N} = 85.2$, so the error exponent is 0.546 where the limiting value is 0.737.

Although Horstein's scheme has its elegance, and good performance was achieved, a theoretical analysis seems complicated. Furthermore, since after each transmission the receiver's distribution has to be recomputed, the scheme suffers from a great computational complexity.

In 1968 Zigangirov [114] considers a modification of Horstein's scheme. The difference is that Zigangirov chooses the parameter $a$ randomly after each transmission and that Zigangirov does not discard a subinterval after a decoder's decision. By investigating special difference equations Zigangirov obtains the same error exponent at capacity as Horstein. Zigangirov also considers a modification for a finite number of (binary) messages with $a = 1/2$, which are decoded simultaneously. In that case Zigangirov obtains an error exponent of $D(q|p)$ independent of the (average) transmission rate, which corresponds with Horstein's exponent $E^0_n$ at zero rate. In contrast with multiple-repetition coding, Zigangirov does not solve Horstein's problem of computational complexity.

1.4 Multiple-repetition strategies

In the previous Sections several schemes were discussed for coding on DMC's with feedback. Most of the results were based on random coding arguments, and few constructive schemes were presented. The schemes that actually were constructive, suffered from problems of memory or computational complexity, which limited their practical applicability. In this context we quote Weldon [109, p. 89]:

One of the aspects of the general problem of improving communication reliability by using feedback which appears to be a very fruitful area for investigation is the problem of finding codes which can take advantage of the availability of the feedback channel so as to reduce the complexity of the equipment needed to implement them. These codes need not be optimum in any sense; they need only be "good" and relatively simple to implement in order to be very useful.

1.4.1 Schalkwijk

Since Horstein's scheme is constructive and achieves high reliability, it is a suitable starting point. In 1971 Schalkwijk [69] was able to derive simple block coding strategies from Horstein's scheme that are easily implementable and are optimal in the sense that they achieve Berlekamp's bounds [11] (see Figure 1.6).

Median paths

Schalkwijk investigated the behavior of the midpoints at capacity ($a = 1/2$), which Schalkwijk called medians. The path that arises when the midpoints of consecutive transmissions are concatenated, was called the median path. In general, these paths are irregular, but Schalkwijk discovered that for certain special cases the median paths are very regular. The median paths for $p = 0.1$ are depicted in Figure 1.11. An example of regular median paths
Figure 1.11: Median paths for $p = 0.1$
Figure 1.12: Median paths for $p = (3 - \sqrt{5})/4$
is shown in Figure 1.12, where \( p = (3 - \sqrt{5})/4 \).

As shown in Figure 1.7, the median moves up when a one is received, and moves down when a zero is received. The half that corresponds with the received symbol becomes more reliable, and the receiver's density distribution is multiplied by \( 2q \) at that half. The other half of the receiver's density distribution is multiplied by \( 2p \) and becomes less reliable. In his search for regular median paths, Schalkwijk looked for paths where the median returns to its initial value after a few transmissions. The simplest case would be after receipt of 01 or 10. E.g. after receipt of 01, the receiver's density in the interval \((0.5, 1)\) is multiplied by \( 2p \cdot 2q \). For the median to return to its initial value 0.5 after receipt of 01, this would necessary imply \( 2p \cdot 2q = 1 \), which has as only solution \( p = 1/2 \). Since the capacity of a BSC with \( p = 1/2 \) is 0, this case is not interesting. The next simplest case would be that the median returns to its initial value after receipt of 011 or 100 (010 is not possible because the median is below its initial value after receipt of 01 since \( 2p \cdot 2q < 1 \)). E.g. after receipt of 011, the receiver's density in the interval \((0.5, 1)\) is multiplied by \( 2p \cdot 2q \cdot 2q \), so the median returns to its initial value when

\[
2p \cdot 2q \cdot 2q = 1. \tag{1.58}
\]

The solution of Equation 1.58 is \( p = (3 - \sqrt{5})/4 \). The median paths for \( p = (3 - \sqrt{5})/4 \) are depicted in Figure 1.12.

\[\text{e-error trees}\]

By taking a closer look at this tree, we see that by shifting the tree three positions to the right, the tree is actually a subtree of its own. If we would delete all edges from the original tree that are also edges in its subtree, a tree remains that Schalkwijk calls the zero-error tree. The tree that results when the zero-error tree is shifted three positions to the right, is called the one-error tree. Similarly, for every integer \( e \), the \( e \)-error tree is defined as the \( 3e \) units delayed version of the zero-error tree. Note that the original tree is the (disjunct) union of all \( e \)-error trees (\( e \geq 0 \)). From the picture in Figure 1.12 it is not clear how each path goes through the tree because some points have more than one incoming edge. This ambiguity is solved by stating that each path in the \( e \)-error tree jumps to the \((e + 1)\)-error tree if and only if it ends with 0111 or 1000. As long as these subsequences do not occur in the received sequence the path stays in the \( e \)-error tree. As an example the median path that corresponds with the received sequence 1001110 is accented in Figure 1.12 by *'s. This path jumps from the zero-error tree to the one-error tree after having received the fourth one. Note that this path leads to the same median as the path corresponding to the received sequence 1010.

\[\text{Idea}\]

As was shown in subsection 1.3.2, the median always moves towards the message point after a correct reception. Since after receipt of e.g. 1000, the median moves below its initial value whereas it moved up after the first transmission, this necessarily implies that there has occurred a channel error. This explains Schalkwijk's term "\( e \)-error tree", because each jump of the median path from the \( e \)-error tree to the \((e + 1)\)-error tree necessarily implies that there has occurred a channel error. In general, when a median path is in the \( e \)-error tree at a certain moment, this implies that the number of channel errors that occurred so far is at least \( e \). It also follows that all error free median paths are contained in the zero-error...
tree, and therefore no correctly received sequence can contain the subsequences 0111 and 1000. Note that this does not imply that the message sequence is not allowed to contain subsequences 0111 and 1000.

Consider the transmitter when a channel error occurred. As an example we take the path 10 which is accentuated in Figure 1.12 by *'s. Suppose at this moment the transmitter decides to move up the median because the message point is above the momentary median, but the transmitted 1 is received as 0. This is noticed by the transmitter, who still wants to move up the median by sending 1s (which we assume are received correctly) until the median exceeds the median that resulted after the second transmission. Eventually, after 100111 is received, the median is at the same level as it would be after correct reception of 101, the only difference between these two sequences being that 101 corresponds to a path in the zero-error tree and 100111 corresponds to a path ending in the one-error tree. It is concluded that, not only, when a median path goes through the e-error tree, necessarily at least e errors have occurred, but also, when e errors have occurred, the path eventually ends up in the e-error tree.

This observation led Schalkwijk to the idea of an error correcting mechanism that does not involve median paths. Namely, each time the transmitter notes that a 1 is received as 0, the transmitter repeats the 1 three times and regards the error as “corrected”. Similarly, as a 0 is received as 1, the transmitter sends three more 0s to correct the error. The receiver is able to actually correct the channel errors by replacing from right to left each subsequence 0111 in the received sequence by 1 and each subsequence 1000 by 0. Note that this error correcting mechanism works recursively, i.e. an error in the “correcting sequence” is corrected in the same way.

Performance

We count the number of medians at depth N in the zero-error tree. Consider an arbitrary message point. Suppose that during N transmissions no channel errors occur. Since after each transmission the median moves towards the message point, the receiver’s density close to the message point is each time multiplied by 2q. The receiver’s density close to the message point after N transmissions is therefore $(2q)^N$. The part of the interval $(0, 1)$ to that side of the previous median at depth $N - 1$, to which the last step is directed, can be readily seen to have posterior probability q (see Figure 1.7(c)). The probability of the part of the interval $(0, 1)$ on either side of the current median by definition has a posterior probability 1/2. Hence, the step from the median at depth $N - 1$ to the current median covers an interval of probability $q - 1/2$. Consequently, the size of this final step is $(q - 1/2)/(2q)^N$. The median at depth $N$ in the zero-error tree that corresponds with the received sequence $r = r_1 \ldots r_N$ equals

$$m(r) = \frac{1}{2} + \sum_{n=1}^{N} \left(2r_n - 1\right)(q - \frac{1}{2})/(2q)^n. \quad (1.59)$$

Let $M_N$ be the number of medians at depth N in the zero-error tree. From Equation 1.59 follows that for large N the distance between two consecutive medians at depth N is of the order $(2q)^{-N}$, so

$$M_N \approx (2q)^N. \quad (1.60)$$

Suppose we would like to correct e errors, then the number of different messages that can be transmitted is equal to the number of endpoints in the e-error tree, which is $M_{N-3e}$,
so the transmission rate equals

\[ R \approx \frac{\log(2q)^{N-3e}}{N} \]
\[ = (1 - \frac{3e}{N}) \log(2q). \]  

(1.61)  
(1.62)

It follows that Schalkwijk's strategy achieves the correctable error fraction \( f = e/N \) (see Definition 1.5) with rate \( R = (1 - 3f) \log(2q) \). For \( f = p \) is obtained

\[ R = (1 - 3p) \log(2q) \]
\[ = (1 - p) \log(2q) + p \log(2q)^{-2} \]
\[ = q \log(2q) + p \log(2p) \]
\[ = 1 - h(p), \]

which equals the volume bound. Note that Equation 1.65 follows from Equation 1.64 using Equation 1.58. It is concluded that Schalkwijk's strategy is optimal in the sense that it achieves maximal correctable error fraction.

Precoding

When no transmission errors occur, the median path stays in the zero-error tree. The paths in the zero-error tree of depth \( N \) correspond with the binary sequences of length \( N \) that do not contain subsequences 0111 and 1000. From Equation 1.59 follows that given a message point, the transmitted sequence \( t = t_1 \ldots t_N \) that would result if no transmission errors occur, is computed recursively by approximating the message point by \( m(t_1 \ldots t_n) \) as closely as possible by choice of \( t_n \) for \( n = 1, \ldots, N \). When \( t \) is determined, the message sequence has been precoded, and the complete transmission system has become independent of the median paths: all occurring errors are corrected by Schalkwijk's error correcting mechanism. In other words, the receiver's distribution is not relevant any more, and the computational problem with the receiver's distribution which was the bottleneck in Horstein's scheme, has been overcome. The resulting transmission scheme is easily implementable, and optimal: the goal as mentioned by Weldon in the beginning of this Section has been achieved.

Generalization

The strategy described above was derived by Schalkwijk from the simplest regular median tree, where the median returns to its initial value after going one step up and two steps down. It is easily generalized by considering median trees where the median returns to its initial value after going one step up and \( k-1 \) steps down, for arbitrary \( k \geq 3 \). This occurs when the channel error probability equals \( p = p_k \), the solution \( p < 1/2 \) of

\[ (2p)(2q)^{k-1} = 1. \]  

(1.67)

The subsequences that are used for the error correction then are 01\(^k\) and 10\(^k\). The correctable error fraction \( f \) agrees with rate \( R \) such that

\[ R = (1 - kf) \log(2q_k), \]

(1.68)

where \( q_k = 1 - p_k \). The rate versus correctable error fraction is a straight line that starts in \((f, R) = (1/k, 0)\) and hits the volume bound when \( f = p_k \).
1.4.2 Schalkwijk and Post

In 1973 Schalkwijk and Post [72, 73] showed that Schalkwijk’s idea of correcting errors is not only suitable for block coding, but can also be used in sequential schemes for the BSC with fixed or variable delay.

Quantile paths

Schalkwijk [69] showed that the paths, formed by consecutive midpoints in Horstein’s scheme, are sometimes very regular when \( a = 0.5 \). Schalkwijk and Post also consider the case \( 1/2 < a \leq 1 \). Since the midpoints in this case do not divide the transmission interval into two equal parts, they are no longer called medians, but quantiles. Note that Schalkwijk and Post let \( a \) denote the probability that a zero is transmitted, as opposed to Horstein who sends a zero with probability \( 1-a \). Therefore, in the paper of Schalkwijk and Post, \( 0 \leq a < 1/2 \).

The regular quantile paths for \( 0 \leq a < 1/2 \) are asymmetrical. In general, the quantile returns to its initial value after going 1 step up and \( k_0 - 1 \) steps down, as well as after going 1 step down and \( k_1 - 1 \) steps up. Schalkwijk and Post [72, p. 500] compute the values of \( a \) and \( p \) for which these regular quantile paths are obtained. Let \( x_i = q/\pi_i \), where \( \pi_i \) is the probability that \( i \) is received (\( i = 0, 1 \)). Then \( x_0 \) and \( x_1 \) have to satisfy

\[
\left( \sum_{i=0}^{k_0-2} x_0^{-i} \right)^{k_1-2} = x_0^{k_0-2} \tag{1.69}
\]

\[
\left( \sum_{i=0}^{k_1-2} x_1^{-i} \right)^{k_0-2} = x_1^{k_1-2}. \tag{1.70}
\]

From the solutions \( x_0 \) and \( x_1 \) of above Equations, the values of \( a \) and \( p \) are easily computed. Take for example \( k_0 = 3 \) and \( k_1 = 4 \), then \( x_0 = 2.1479 \) and \( x_1 = 1.46557 \), so \( a = 0.37281 \) and \( p = 0.12884 \). The resulting quantile paths are depicted in Figure 1.13.

Recursive coding

Schalkwijk and Post derive a sequential coding scheme for the case when the quantile tree is regular, i.e. the quantile returns to its initial value after going 1 step up and \( k_0 - 1 \) steps down, as well as after going 1 step down and \( k_1 - 1 \) steps up. Note that the coding scheme can also be used in case the quantile tree is not regular, since the scheme is actually independent of Horstein’s quantile paths, but the performance analysis of Schalkwijk and Post is only valid in case the quantile paths are regular. It is assumed that the message (sequence) to be transmitted corresponds with an (infinite) path in the zero-error quantile tree: the message does not contain subsequences \( 10^{k_0} \) and \( 01^{k_1} \). The encoder uses Schalkwijk’s error correcting mechanism: when a 1 is received as a 0, \( k_1 \) extra 1s are transmitted, and when a 0 is received as a 1, another \( k_0 \) 0s are sent. The decoder is composed of a code word estimator and an inverse encoder. The code word estimator tries to reconstruct the transmitted sequence, and from the estimate obtained by the code word estimator, the inverse encoder produces the (probable) message sequence.

Let \( a_n \) be the quantile after the \( n^{th} \) transmission (\( a_0 = a \)). The quantile \( a_n \) is equal to Horstein’s midpoint after \( n \) transmissions. Since the transmitter sends a 1 when the message point is above the quantile: the quantile has to move up, and 0 otherwise, Schalkwijk and
Figure 1.13: Quantile paths for $a = 0.37281$ and $p = 0.12884$
Post had the idea of comparing $a_{n-1}$ and $a_{n+D}$ for sufficiently long $D$ to decide on the $n^{th}$ transmitted bit. When $a_{n-1} < a_{n+D}$, the code word estimator assumes a 1 was transmitted, and 0 otherwise. The quantile $a_{n+D}$ is used as a (reliable) estimate of the message point. The number $D$ is the decoding delay. In order to compare $a_{n-1}$ and $a_{n+D}$, a first order approximation of the quantile is used in the state diagram of Figure 1.14. Suppose the code word estimator wants to reconstruct the $n^{th}$ transmitted symbol. Before receiving the $n^{th}$ symbol, the code word estimator starts in state 0. Whenever a 1 is received, the estimator goes up along the arrows, and goes down after receiving a 0. Each state is a first order approximation of the current quantile. From the positive states, the estimator moves up $k_0 - 1$ steps after receiving a 1 and goes down one step after each 0. When in the negative states, the estimator moves down $k_1 - 1$ steps after receiving a 0 and goes down one step after each 1. It is easily seen that the regularity of the quantile paths is maintained in the state diagram. After $D$ symbols are received, the code word estimator is ready to give a reliable estimate of the $n^{th}$ transmitted symbol. When the code word estimator is in a positive state in the state diagram after transmission $n + D$, 1 will be a good estimate for the $n^{th}$ transmitted symbol. Otherwise, the code word estimator assumes that a 0 was transmitted at the $n^{th}$ transmission.

The inverse encoder compares the received sequence with the estimated transmitted sequence given by the code word estimator and obtains the error location estimates. Working from left to right through the estimated transmitted sequence, the inverse encoder deletes $k_0$ not previously deleted zeros on the right-hand side of each (estimated) 0 → 1 error, and it deletes $k_1$ not previously deleted ones on the right-hand side of each (estimated) 1 → 0 error. In this way, if the code word estimator has not made a mistake, correction bits are discarded and zero-error path symbols are retained. As an example, assume that $k_0 = 3$, and the estimated transmitted sequence is 1000000001. The notation 0 is used to denote a 0 that is received as a 1. For the estimated error at time 3, the inverse encoder deletes the zeros at position 4, 5, and 6, and for the second estimated error at time 5, the inverse encoder deletes the zeros at position 7, 8, and 9. The remaining sequence is then 1001, and the errors at time 3 and 5 have been “corrected”.

Results

We assume that $a$ and $p$ are fixed such that the quantile paths are regular.

Schalkwijk and Post consider two different types of code word estimators. The first one uses a fixed delay $D$, and the decision for each transmitted symbol is made after exactly $D$ transmissions. The second one uses a variable delay $D$. The variable delay code word estimator decides for a 1 whenever a certain positive threshold state is achieved, and similarly for a 0. These threshold states are fixed, but the number of transmissions needed to reach a threshold differs per estimation. The thresholds are chosen such that a prescribed error probability is assured. The variable delay code word estimator is very similar to Horstein’s decoding method.

Although it is not absolutely clear that the described encoding method gives the same transmission rate as Horstein, Schalkwijk and Post [72, p. 505] state that

$$R = h(\pi_1) - h(p).$$  \hspace{1cm} (1.71)

By using advanced function theoretic arguments, Schalkwijk and Post [72, p. 503] are able to determine the error exponent of the code word estimator in case of fixed delay, but
Figure 1.14: State diagram for code word estimator
are not aware of an elegant presentation by means of informational divergences:

\[ E'_p = \min \left\{ D \left( \frac{1}{k_0} \| \pi_1 \right), D \left( \frac{1}{k_1} \| \pi_0 \right) \right\}. \]  

(1.72)

Note that the exponent \( D \left( \frac{1}{k_0} \| \pi_1 \right) \) is the error exponent when a 0 is transmitted, and \( D \left( \frac{1}{k_1} \| \pi_0 \right) \) corresponds with the transmission of a 1.

Similarly, Schalkwijk and Post [72, p. 505] determine the reliability of the code word estimator for the variable delay coding scheme, again without using this notation:

\[ E'_p = D(\pi_1 \| p) \]  

(1.73)

\[ E'_s = D(\pi_0 \| p). \]  

(1.74)

The exponent \( E'_p \) is equal to Horstein's \( E'_q \) (see Equation 1.55) and corresponds with the transmission of a 1. The other exponent, that corresponds with the transmission of a 0, was overlooked by Horstein.

We would like to draw the attention of the interested reader, that is going to study the paper of Schalkwijk and Post, to a relation between the variables used which is not mentioned by Schalkwijk and Post, namely that \( w_0 = 1/x \) and \( \tilde{w}_0 = 1/y \).

Conclusion

Schalkwijk and Post were able to rigorously prove Horstein's results, including the distribution of the coding delay, in case the quantile paths are regular. An easy left-to-right (in contrast with Schalkwijk's right-to-left block decoding) sequential decoding algorithm was presented by Schalkwijk and Post. Even in the case of fixed coding delay, the error exponent is positive at capacity. Note that the constraint length however remains variable, or in the words of Schalkwijk and Post: the time instants at which information (as opposed to correction) digits are sent are determined by the received data.

In 1977 Zigangirov [117] considers a variation with a bounded constraint length of the fixed coding delay scheme of Schalkwijk and Post. Zigangirov uses the right-to-left substitution decoding method and an encoding buffer. The length of the encoding buffer is equal to the coding delay. Zigangirov claims that an error exponent can be achieved whose exponent vs. rate curve hits the \textit{recurrent sphere packing bound}. The recurrent sphere packing bound is the inverse concatenation construction (see Figure 1.3) of the sphere packing bound (see Equation 1.9), and is shown by Viterbi [106] to determine the optimal exponent for convolutional codes (without feedback). However, Zigangirov's proof does not hold because a mistake was made. More precisely, Zigangirov erroneously assumes that an information symbol is decoded correctly when the number of channel errors that occur during transmission of the first \( D + 1 \) symbols does not exceed \( D/k \). A simple counterexample for \( D = k = 3 \) is the transmitted sequence 0110 (remember that 1 denotes the transmission of a 0 that is received as a 1) which can not be distinguished by the receiver from the transmitted sequence 0111. Both sequences are decoded to 1, but the first one should be decoded to 0. We will run into the same problem in later Chapters of this thesis, which will necessarily complicate the proofs.

1.4.3 Becker

In 1973 Becker [7], at that time a Ph.D. student of Schalkwijk, wrote a dissertation about multiple-repetition feedback coding. Becker describes a generalization of Schalkwijk's [69]
block coding schemes for the BSC to arbitrary DMC's with feedback. Just as in the rest of this thesis, Becker considers multiple-repetition feedback strategies in an algebraic way without concern about quantile or median paths. Although it is useful to keep the idea of regular median paths in mind, since the repetition strategies are derived from them, an actual scheme that involves computation of the receiver's distribution according to Horstein necessarily would increase the computational and memory complexity.

Becker's idea is to use repetition parameters $k_{xy} \ (x \in \mathcal{X}, y \in \mathcal{Y})$, such that whenever an input symbol $x$ is erroneously received as a $y$, the symbol $x$ is repeated $k_{xy}$ times by the transmitter. Consequently, all message sequences have to be precoded: subsequences $yx^{k_{xy}}$ are not allowed.

Becker considers three types of DMC's with feedback with increasing generality. However, for all DMC's considered by Becker, the number of inputs and outputs are equal. The strict-sense symmetric channels are DMC's where the channel probabilities $p_{xy}$ are constant for all $x$ and $y$ such that $x \neq y$. The second type of DMC's are wide-sense symmetric channels, where the channel probabilities $p_{xy}$ depend on the modulo $|x|$ difference between $y$ and $x$ (the input and output alphabets consist of the integers from 0 upto $|\mathcal{X}| - 1$). Becker's motivation for considering this class of DMC's separately, is that a wide-sense symmetric DMC arises when discrete communication is performed on the additive Gaussian noise channel. The most general type of DMC's with feedback considered by Becker is the class of asymmetric DMC's, where there are no restrictions on the channel probabilities.

Becker develops block coding schemes, where each block of length $N$ consists of $L$ information symbols (as opposed to repetition symbols) and a tail of length $N - L$. Becker shows that when the tails are suitably chosen, each block is decoded correctly (the $L$ information symbols are retrieved) whenever

$$\sum_{x \neq y} k_{xy} e_{xy} < N - L,$$  \hspace{1cm} (1.75)

where $e_{xy}$ denotes the number of $x \rightarrow y$ errors in the block. For each type of DMCF, suitable tails are obtained. Furthermore, the transmission rate of the system is determined for each type of DMCF, and shown to be asymptotically optimal in certain cases (for asymmetric DMC's the optimality is conjectured but not proven).

Becker proposes a solution to the problem of precoding by developing a precoder and an inverse precoder.

Today we would say that Becker's precoding algorithms are based on the idea of enumerative (de)coding, but in 1973 this idea was unknown. In the cases of strict-sense symmetric and wide-sense symmetric repetition coding, the computational and memory complexity of the (inverse) precoder are acceptable, but in the asymmetric case the symbol distribution of the precoded sequences has to be set tight to a certain value to obtain a maximal transmission rate. When the symbol distribution is taken into account at the (inverse) precoder, their complexity dominates the complexity of the entire coding system. When a slightly less than optimal transmission rate is sufficient for a certain application, this problem can be avoided. More about precoding is found in Section 7.2 of this thesis.

1.4.4 Veugen

In this thesis Veugen will present several new results on multiple-repetition feedback strategies. The class of DMC's that is considered is larger than Becker's class of asymmetric DMC's in the sense that the output alphabet is allowed to exceed the input alphabet. Note
that it is not necessary to consider the case with more input symbols than output symbols because Shannon [88] showed that it is possible then to eliminate a suitably chosen input symbol without affecting the capacity of the channel. However, due to the idea that an error must be corrected by repeatedly sending the correct input symbol, multiple-repetition feedback strategies can not be efficiently used on all DMC’s. An example of a DMC that is unsuitable for multiple-repetition feedback strategies is depicted in Figure 1.15. It is assumed in Figure 1.15 that $p_0 > p_1 > p_2 > p_3$. A channel like in Figure 1.15 arises when the output of a continuous channel is quantized in four levels, although more levels are similarly modeled. So when for example the output symbol $\frac{1}{4}$ is received, it is more likely that a 0 was transmitted than a 1. A multiple-repetition strategy would not use this extra information and simply consider $\frac{1}{4}$ as an erasure: both transmitter and receiver know that $\frac{1}{4}$ can not be the symbol that was transmitted, so a transmission error must have occurred and that erroneous transmission will be ignored. In other words, the use of multiple-repetition strategies implies hard decision decoding (see also the paragraph “Soft decision” in Section 8.2). Note that because the receiver knows that a transmission error has occurred when $\frac{1}{4}$ is received, only one repetition will be sufficient to “correct” that error. Consequently, in this thesis the repetition numbers $k_{xy}$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y} \setminus \mathcal{X}$ will be equal to 1.

Another consequence of using multiple-repetition strategies is that each input symbol $x \in \mathcal{X}$ is assumed to have exactly one “correct” output symbol $y \in \mathcal{Y}$: receiving another output symbol $y \neq x$ is considered as a transmission error (or as an erasure when $y \notin \mathcal{X}$). So in order to efficiently use a multiple-repetition strategy, the labeling of the input and output symbols should be chosen such that $\mathcal{X} \subseteq \mathcal{Y}$ and $p_{xx} \geq p_{xy}$ for each $x, y \in \mathcal{X}$. For most DMC’s in practice it is possible to find such a labeling.

The theoretical results that will be presented in the following Chapters, are valid for all DMC’s with feedback, but the reader should bare in mind that $k_{xy} = 1$ is assumed for $x \in \mathcal{X}$ and $y \in \mathcal{Y} \setminus \mathcal{X}$, and that consequently all outputs $y \in \mathcal{Y} \setminus \mathcal{X}$ are considered as erasures.
1.5 Conclusions

The availability of a feedback link does not increase the capacity of a DMC. However, the complexity of coding schemes can be significantly reduced which is of great practical importance. But also in theory, it opens new possibilities for the continuing search for good codes that do exist according to Shannon's basic theorem of information theory (see theorem 1.1).

Unfortunately, few constructive coding schemes are known for DMC's with feedback. Furthermore, most constructive schemes suffer from problems of computational or memory complexity. One of the most attractive constructive coding schemes is Horstein's scheme [45] for the BSC with feedback. The complexity problems of Horstein's scheme were overcome by Schalkwijk [69] (block coding), and Schalkwijk and Post [72, 73] (recursive coding). Becker was able to generalize Schalkwijk's block coding schemes to multiple-repetition feedback strategies for DMC's with feedback with equal input and output alphabets.

It is concluded that multiple-repetition feedback strategies form an important class of easy implementable and efficient coding schemes that can be used on many DMC's with feedback. Therefore, it is worthwhile to study the properties of multiple-repetition feedback strategies. In each of the following Chapters a particular aspect of multiple-repetition feedback strategies will be considered, and new results will be presented. Each Chapter starts with a quick overview of the contents of that Chapter, and ends with some conclusions. In Section 8.1 a complete overview can be found of all new results together with their references.
Chapter 2

Block coding

In subsection 1.4.1 it was shown how Schalkwijk [69] derived repetition feedback block coding schemes for the BSC from regular median paths in Horstein’s scheme [45]. As mentioned in subsection 1.4.3, Becker [7] generalized Schalkwijk’s idea to DMC’s with equal number of input and output symbols using an algebraic point of view. Becker’s generalization introduced a new problem: how should the tail of a block be chosen to guarantee the error correcting capabilities of the code? Before the general tail problem is examined, we consider the simple case of binary symmetric block coding. Eventually, a new tail condition is presented that extends Becker’s results.

2.1 Binary symmetric block coding

In this Section we restrict ourselves to binary (\( \mathcal{X} = \mathcal{Y} = \{0, 1\} \)) symmetric repetition feedback strategies with repetition parameter \( k \). When \( k = 2 \) there are only \( 2L \) binary sequences of length \( L \) without forbidden subsequences, (namely the sequences \( 0^i1010... \) and \( 1^i0101... \) for \( 1 \leq i \leq L \)) so the amount of information per precoded symbol, which equals \( \log(2L)/L \), approaches 0 for increasing length. It is easily seen that the number of precoded sequences of length \( L \) increases exponentially with \( L \) when \( k \geq 3 \) (see e.g. Equation 1.60). Therefore we assume \( k \geq 3 \). Let \( N \) be the block length. In order to transmit information in blocks of length \( N \), the transmitter lets each block consist of \( L \) (precoded) information symbols completed with a tail of length \( N - L \). The tail is chosen prior to the transmission of the block, and determines the error correcting capability of the strategy (see subsection 2.1.1). The number \( L \) is fixed and known to both transmitter and receiver. It is assumed that a precoded sequence \( p \) of length \( L \) is given. For more details on precoding see Section 7.2. The sequence \( p = p_1 \ldots p_L \) does not contain subsequences \( 01^k \) or \( 10^k \).

When a channel error occurs during the transmission of a block, the transmitter repeats the last (erroneously received) symbol \( k \) times. This also leads to a shortening at the end of the (remaining) tail with \( k \) symbols. Note that this error correcting mechanism works recursively, i.e. an error in the “correcting sequence” is corrected in the same way.

The receiver decodes the received block by scanning the block from right to left and replacing each subsequence \( 01^k \) or \( 10^k \) by 1 or 0 respectively. The substitutions can not be done from left to right because the errors have to be corrected in the reverse order as they occurred. The receiver obtains an estimate \( \hat{p} \) for the precoded sequence \( p \), by discarding all but the first \( L \) symbols of the decoded sequence.
An overview of the system is depicted in Figure 2.1, where the code word $c$ is the precoded sequence $p$ concatenated with a suitably chosen tail. The decoded sequence $d$ is the remaining part of the received sequence after all forbidden subsequences have been substituted. Usually $D \geq L$, so the $D - L$ last symbols of the decoded sequence can be deleted to obtain the estimated precoded sequence $\hat{p}$. When $D < L$ it is impossible for the receiver to recover the message $m$ so a decoding error is declared. An explicit example is illustrated in Figure 2.2, where $k = 3$, $L = 4$ and $N = 10$. Note that throughout this thesis pictures of repetition strategies are used where a white square denotes a zero, a black square a one, and a combined square a channel error. In non-binary examples a white square denotes a zero, a grey square a one, and a black square a two, so that in all cases the numerical value of a square increases with its darkness.

### 2.1.1 Error correcting capabilities

The tail should be chosen in such a way that the decoding of the information symbols is not affected. For the binary symmetric case, a tail consisting of alternating symbols such that the first tail symbol differs from the last information symbol seems to be a good choice. An alternating tail minimizes the probability that a forbidden subsequence in the received tail could come about which could affect the decoding of the information symbols.

Let $e$ be the number of channel errors that occurred during the transmission of a block. Since each channel error causes a shortening of the tail with $k$ symbols, it is necessary that
the tail length is at least $ke$ in order to correct all errors. The example in Figure 2.3, where $k = 3$, $L = 3$, $N = 6$ and $e = 1$ shows that a block length of $L + ke$ is not sufficient. In the next subsection it is proved that when an alternating tail is used, all error patterns in a block are corrected as long as

$$L + ke < N,$$  \hspace{1cm} (2.1)

which was also proved slightly differently by Becker [7, pp. 16-22].

2.1.2 Proof

In this subsection we assume that the transmitter and the receiver encode and decode the information according to the repetition feedback block coding principles as described above. The transmitted and received sequences are denoted by $\hat{t} = t_1 \ldots t_N$ and $\hat{r} = r_1 \ldots r_N$ respectively. The precoded sequence of information symbols is denoted by $\hat{p} = p_1 \ldots p_L$. The code word $\hat{c}$ is the precoded sequence concatenated with an alternating tail such that the first tail symbol differs from $p_L$. The receiver’s estimate of the precoded sequence is denoted by $\hat{\hat{p}} = \hat{p}_1 \ldots \hat{p}_L$.

The case $L = 1$ is considered first. It turns out that for $L = 1$ even $L + ke = N$ (equality in Equation 2.1) is sufficient to guarantee correct decoding.

**Lemma 2.1** Suppose $L = 1$. If $ke < N$, then $\hat{\hat{p}} = \hat{p}$.

**Proof:** Note that the code word $\hat{c}$ is a sequence consisting of alternating symbols. Suppose $\hat{ke} < N$. The proof is by natural induction to $e$. If $e = 0$, then $\hat{r}$ is error-free and equal to $\hat{c}$, so $\hat{\hat{p}} = \hat{p}$.

Suppose that ($e > 0$, and that) no forbidden subsequence occurs in $\hat{r}$. Then $\hat{p}_1 = r_1$, and furthermore, between two consecutive transmission errors in $\hat{r}$ are at most $k - 1$ correctly received symbols. Assume $\hat{\hat{p}} \neq \hat{p}$, i.e. $r_1 \neq t_1$, then the last error occurred not later than
transmission $k(e - 1) + 1$. Therefore, there are at least $N - (k(e - 1) + 1) \geq k$ correctly received symbols after the last error, as depicted in Figure 2.6 for the general case. This means that the last error is corrected, and a forbidden subsequence occurs in $r$, which is a contradiction. It follows that $\hat{p} = p$.

We are left with the case: $(e > 0$, and $) a forbidden subsequence occurs in $r$. Let $r_n \ldots r_{n+k}$ be the rightmost forbidden subsequence of $r$. We will show that the induction argument can be applied to the induced instant with parameters $N' = N - k$, $\hat{p}' = \hat{p}$, $c' = c_1 \ldots c_N'$, $t' = t'_1 \ldots t'_N = t_1 \ldots t_{n-1}t_{n+k} \ldots t_N$, $r'_n = r'_1 \ldots r'_N = r_1 \ldots r_{n-1}r_{n+k} \ldots r_N$, and $\hat{p}' = \hat{p}$. The induced instant arises when the receiver substitutes $r_n \ldots r_{n+k}$ by $r_{n+k}$, which is the first step in the decoding process. When $\hat{p}' = \hat{p}'$ it is clear that $\hat{p} = p$. To apply the induction argument, the number of channel errors in the induced instant has to be less than $e$. Three cases are distinguished:

\begin{itemize}
  \item $r_n \neq t_n$. In this case the error at transmission $n$ is followed by $k$ error-correcting symbols, so substitution of $r_n \ldots r_{n+k}$ by $r_{n+k}$ just corrects this error and reduces the number of channel errors by one.
  
  \item $r_n = t_n$ and $r_{n+1} = t_{n+1}$. Since the $n$th transmission is error-free, and $t_{n+1} \neq t_n$, the $(n+1)$th transmission is an error-free transmission of a code word symbol. Since $c$ is alternating, $t_{n+2} \neq t_{n+1}$. Furthermore, because $t_n \neq t_{n+1}$, and the input alphabet is binary, $t_{n+2} = t_n$. Therefore, transmissions $n + 2$ until $n + k$ are all erroneous transmissions of symbol $t_n$. With the substitution of $r_n \ldots r_{n+k}$ by $r_{n+k}$, the channel errors at transmissions $n + 2$ to $n + k$ are eliminated, but a new channel error is introduced at transmission $n$, so the number of channel errors is reduced by $k - 2 \geq 1$. Note that in this case during transmissions $n + 3$ to $N$ the transmitter (unsuccessfully) tried to correct the channel error that occurred at transmission $n + 2$ (by repeatedly transmitting symbol $t_{n+2}$).

  \item $r_n = t_n$ and $r_{n+1} \neq t_{n+1}$. In this case $t_n = t_{n+1}$, and since $c$ is alternating, the $n$th transmission must therefore be an error-free transmission of an error-correcting symbol. Consequently, transmissions $n + 1$ until $n + k$ are all erroneous transmissions of symbol $t_n$. The substitution eliminates the channel errors at transmissions $n + 1$ to $n + k$,
but introduces a new one at transmission $n$, so it reduces the number of channel errors by $k - 1 \geq 1$. Note that in this case during transmissions $n + 2$ to $N$ the transmitter (unsuccessfully) tried to correct the channel error that occurred at transmission $n + 1$ (by repeatedly transmitting symbol $t_{n+1}$).

It remains to show that in each case $t'_n$ and $r'_n$ could be sequences of a repetition strategy evolving from code word $c'$. For the first case this is obvious. For the last two cases it is clear up-to transmission $n - 1$. In both cases $t'_n = t_n$ and $r'_n \neq t_n$, so since no forbidden subsequences occur in $r'_n \ldots r'_{N'}$, transmissions $n + 1$ to $N'$ can be seen as an unsucceeded attempt to correct the error that occurred at the $n$th transmission (remember that in both cases $t_{n+k} = t_{n+k+1} = \ldots t_N = t_n$). Therefore the induction argument can be applied.

The unpracticality of the $k = 2$ strategies is emphasized by a counterexample of Lemma 2.1 for $k = 2$ and arbitrary block length: $c_1 = 1$, $t = 1010 \ldots 101$, and $r = 1010 \ldots 100$ which reduces to $\hat{p} = 0$.

With the result of Lemma 2.1, the proof for the case of arbitrary number of information symbols is easy.

**Theorem 2.1** If $L + ke < N$, then $\hat{p} = p$.

**Proof:** Suppose $L + ke < N$. Let $e_p$ and $e_t$ be the number of errors that occurred during the transmission of respectively, the information part $p$, and the (remaining) tail ($e = e_p + e_t$). Each transmission error in the information part $p$ leads to $k$ extra repetition symbols, and a reduction of the tail by $k$. So it takes $L + ke_p$ transmissions to transmit $p$ such that all occurring errors are corrected, and there are $N'' = N - (L + ke_p) > ke_t$ transmissions left to transmit the remaining tail. Since transmissions $L + ke_p + 1$ upto $N$ can be seen as an attempt to transmit the first tail symbol, it follows from Lemma 2.1 that $r_{L+ke_p+1} \ldots r_N$ is decoded to a sequence $\hat{d}$ that starts with the first tail symbol. The situation is illustrated in Figure 2.7 where $\hat{d}$ denotes the decoded sequence that remains after replacing the forbidden subsequences in $r$.

Furthermore, the $(L + ke_p)^{th}$ transmission completes the sending of $p$, so it must be an error-free transmission. Therefore, Lemma 2.1 can again be applied to deduce that $r_{L+ke_p} \ldots r_N$ is decoded to a sequence that starts with $p_L$. It also follows that $r_{L+ke_p} \ldots r_N$ is decoded to $p_L\hat{d}$, because $\delta_1 \neq p_L$. Because $r_1 \ldots r_{L+ke_p}$ is decoded to $p_L$, and $r_{L+ke_p} \ldots r_N$ is decoded to $p_L\hat{d}$, it is concluded that $r_1 \ldots r_N$ is decoded to $p_L\hat{d}$, since $\delta_1 \neq p_L$. Therefore, $\hat{p} = p$.

Since the reliability of a decoded symbol decreases when going from left to right through the decoded sequence, one might think that a block is decoded correctly as long as the last information symbol is decoded correctly. This statement is contested by the following counterexample: $p = 011$, $t = 011000$, $r = 011101$, so $\hat{p} = 101$.

### 2.2 Block coding for arbitrary channels

In this Section multiple-repetition feedback block coding for arbitrary discrete memoryless channels is considered and the results of the preceding Section are generalized. Although the theoretical results are valid for all DMC's, the reader is reminded that for some type of channels multiple-repetition strategies are less suitable (see subsection 1.4.4). For larger
alphabet cardinalities the number of possible tail constructions grows and it is not obvious
how a suitable tail should be chosen. Especially when the repetition numbers are small, a
forbidden subsequence could easily arise in the received tail. Several Tail Conditions are
considered that were presented by Becker [7, p. 59-60]. Inspired by the idea of an alternating
tail, a new Tail Condition is presented and its validity is proved.

Consider a discrete memoryless channel with input alphabet $X$ and output alphabet $Y$.
Similar to binary symmetric block coding, a block consists of a fixed number $L$ of information
symbols completed with a tail of length $N-L$. As opposed to binary symmetric block coding,
the repetition parameter is not necessarily fixed for general block coding. When an $x \rightarrow y$
error occurs, the transmitter repeats the symbol $x$ $k_{xy}$ times ($x \in X, x \neq y, y \in Y$), and
shortens the end of the (remaining) tail with $k_{xy}$ symbols. Note that this error correcting
mechanism works recursively, i.e. an error in the “correcting sequence” is corrected in the
same way. Given a message $m \in M$, the precoder constructs an $X$-ary sequence (i.e.
a sequence consisting of elements of $X$) $p$ of length $L$ that does not contain forbidden
subsequences $yx^{k_{xy}}$. For asymmetric channels it is useful for the precoder to fix the symbol
precoding distribution in order to increase the transmission rate (see Section 4.1). The
receiver scans the received block from right to left and replaces each forbidden subsequence
$yx^{k_{xy}}$ by $x$. After that the first $L$ remaining symbols form an estimate for the (precoded)
message.

An overview of a multiple-repetition block coding scheme is depicted in Figure 2.1, and
an explicit example is shown in Figure 2.4, where a white square denotes a zero, a grey square
denotes a one, a black square denotes a two, and a combined square denotes a channel error.
The parameters in the example are $X = Y = \{0, 1, 2\}$, $L = 4$, $N = 10$, and all repetition
parameters are equal to three.
2.2.1 Error correcting capabilities

Let \( e_{xy} \) be the number of \( x \rightarrow y \) channel errors that occurred during the transmission of a block. Since an \( x \rightarrow y \) channel error causes a shortening of the tail with \( k_{xy} \) symbols, it is necessary that the tail length is at least \( \sum_{xy} k_{xy} e_{xy} \) in order to correct all errors. In Figure 2.3 it is shown that equality is not sufficient, even in the binary symmetric case. Therefore, the best possible result would be that all error patterns in a block are corrected when

\[
L + \sum_{xy} k_{xy} e_{xy} < N \tag{2.2}
\]

Obviously, this depends on the tail that is used. We present four different Tail Conditions. If a tail is used that satisfies one of these four Tail Conditions, all error patterns in a block are corrected when Equation 2.2 holds. The Tail Conditions pose restrictions on the tail, but as in the binary symmetric case, the tail depends also on the last information symbol \( p_L \). Therefore the \( X \)-ary sequence \( c_L \ldots c_N \), which equals the tail preceded by the last information symbol, is used in the description of the Tail Conditions.

**Tail Condition 1** The sequence \( c_L \ldots c_N \) satisfies both:

1. The sequence \( c_L \ldots c_N \) is not a forbidden sequence and \( c_L \neq c_{L+1} \).
2. For each \( L < n \leq N \), \( k_{c_{ny}} \geq N - L \) for each \( y \neq c_n \), \( y \in \mathcal{Y} \).

It is obvious that a tail satisfying Tail Condition 1 is received error-free when Equation 2.2 holds, because any transmission error that occurs while transmitting the tail would violate Equation 2.2. Since the first tail symbol is not equal to the last information symbol, the received tail can therefore impossibly introduce an extra forbidden subsequence, so the received message will be decoded correctly. This Tail Condition is unsuitable for larger tail lengths. Tail Condition 1 was also mentioned by Becker [7, p. 95], but he forgot the first requirement and had \( N \) in stead of \( N - L \) which unnecessarily weakens the Tail Condition.

**Tail Condition 2** The sequence \( c_L \ldots c_N \) satisfies both:

1. The sequence \( c_L \ldots c_N \) is a sequence consisting of alternating symbols \( c_L \) and \( c_{L+1} \) such that \( c_L \neq c_{L+1} \) (so \( c_L \ldots c_N = c_L c_{L+1} c_L \ldots \)), \( k_{c_L c_{L+1}} \geq 3 \), and \( k_{c_{L+1} c_L} \geq 3 \).
2. For each \( L \leq n \leq N \), both \( k_{c_{ny}} \geq 2 \) and \( k_{yc_n} \geq 2 \) for each \( y \neq c_n \), \( y \in \mathcal{Y} \).

This Tail Condition can be seen as a generalization of the alternating tail in the binary symmetric case. It’s correctness is proved in the next subsection.

**Tail Condition 3** The sequence \( c_L \ldots c_N \) satisfies both:

1. Each subsequence \( x_1 x_2 \) that appears in \( c_L \ldots c_N \) satisfies both \( x_1 \neq x_2 \) and \( k_{x_2 x_1} \geq 3 \).
2. For each \( L \leq n \leq N \), both \( k_{c_{ny}} \geq 3 \) and \( k_{yc_n} \geq 2 \) for each \( y \neq c_n \), \( y \in \mathcal{Y} \).

If the tail satisfies Tail Condition 3, then a tail can be constructed that satisfies Tail Condition 2, namely the tail \( c_L c_{L+1} c_{L+1} \ldots \). A counterexample for the reverse implication is: \( \mathcal{X} = \mathcal{Y} = \{0, 1, 2\} \), \( k_{01} = k_{10} = 3 \), \( k_{02} = k_{20} = k_{21} = k_{12} = 2 \), in which case an alternating \( 0 \rightarrow 1 \) tail satisfies Tail Condition 2, while no tail can be constructed that satisfies Tail Condition 3. Tail Condition 3 was also mentioned by Becker [7, p. 59] but he didn’t require the necessary demand \( k_{yx} \geq 2 \), which actually falsifies his proof [7, pp. 65-73].
Tail Condition 4 The sequence $c_L \ldots c_N$ satisfies both:

1. Each subsequence $x_1x_2$ that appears in $c_L \ldots c_N$ satisfies both $x_1 \neq x_2$ and $k_{x_2x_1} \geq 4$.

2. For each $L \leq n \leq N$, both $k_{c_ny} \geq 2$ and $k_{yc_n} \geq 2$ for each $y \neq c_n$, $y \in \mathcal{Y}$.

Becker [7, p. 59-60] also mentioned Tail Condition 4 and proved its correctness [7, pp. 65-73]. It is possible that a tail can be constructed that satisfies Tail Condition 4, while no tail (of length at least 2) can be constructed that satisfies Tail Condition 2. For example: $X = Y = \{0, 1, 2\}$, $k_{01} = k_{02} = k_{21} = 4$, $k_{01} = k_{12} = k_{20} = 2$, where the tail $012012 \ldots$ satisfies Tail Condition 4, but no alternating $s_2-s_1$ tail exists such that $k_{s_1s_2} \geq 3$ and $k_{s_2s_1} \geq 3$. It is also possible that a tail can be constructed that satisfies Tail Condition 2, while no tail (of length at least 2) can be constructed that satisfies Tail Condition 4. For example: $X = Y = \{0, 1, 2\}$, $k_{01} = k_{10} = 3$, $k_{20} = k_{21} = k_{12} = 2$, where an alternating $0 - 1$ tail satisfies Tail Condition 2, but no two symbols $x$ and $y$ satisfy $k_{xy} \geq 4$. As mentioned before, for these repetition parameters also no tail (of length at least 2) exists that satisfies Tail Condition 3.

It follows that Tail Condition 2 really broadens Becker’s class of allowable tail constructions and thereby rejects Becker’s Conjecture [7, p. 73] that his class would be complete. In an attempt to further relax the Tail Conditions, one example comes to mind:

Tail Condition 5 The sequence $c_L \ldots c_N$ satisfies both:

1. Each subsequence $x_1x_2$ that appears in $c_L \ldots c_N$ satisfies both $x_1 \neq x_2$ and $k_{x_2x_1} \geq 3$.

2. For each $L \leq n \leq N$, both $k_{c_ny} \geq 2$ and $k_{yc_n} \geq 2$ for each $y \neq c_n$, $y \in \mathcal{Y}$.

Unfortunately, a counterexample for the correctness of Tail Condition 5 is given in Figure 2.5, where $X = Y = \{0, 1, 2\}$, $k_{20} = 2$, $k_{01} = k_{10} = k_{21} = k_{12} = k_{02} = 3$, $N = 16$, $L = 3$, and $\sum_{xy} k_{xy} e_{xy} = k_{20} \cdot 6 = 12$. It seems difficult to further relax the Tail Conditions, but we didn't look for a completeness proof since the proof for the correctness of a specific Tail Condition is already complicated, as can be seen in the following subsection.

2.2.2 Proof of Tail Condition 2

In this subsection the correctness of Tail Condition 2 is proved. We assume that the transmitter and the receiver encode and decode the information according to the repetition feedback block coding principles as described before. The transmitted and received sequences are denoted by $t = t_1 \ldots t_N$ and $r = r_1 \ldots r_N$ respectively. The decoded sequence is denoted by $d$. The decoded sequence is the result of the receiver's decoding of the received sequence such that no forbidden subsequence occurs in $d$. The precoded sequence of information symbols is denoted by $p = p_1 \ldots p_L$. The code word $c$ is the precoded sequence concatenated with the tail. The decoder's estimate of the precoded sequence is denoted by $\hat{p} = \hat{p}_1 \ldots \hat{p}_L$. We assume that a tail is used that satisfies Tail Condition 2.

We try to follow the proof for the binary symmetric case as in subsection 2.1.2, but as was indicated at the end of subsection 1.4.2, the proofs become necessarily more complicated since forbidden subsequences in the received tail can arise in various ways. When proving the case $L = 1$, as in Lemma 2.1, one subcase has to be considered separately in Lemma 2.2. The situation described in Lemma 2.2 is comparable to the example in Figure 2.3: the transmission of the block seems to progress smoothly, but at the end a bunch of equal
Figure 2.5: Counterexample for further relaxed tail
channel errors occur which could affect previously correctly received information symbols during decoding. In the proof of Lemma 2.2 we use the notation $a \div b$ for non-negative integer $a$ and positive integer $b$, to denote the downwards rounded result of $a/b$.

**Lemma 2.2** Let $L = 1$. Suppose that all transmission errors are corrected, i.e. all necessary error-correcting symbols are received. Let $y$ be an arbitrary (output) symbol that occurs $Y$ times in the (decoded) sequence $d$ of length $D$. Let $\Psi$ be a binary variable, that is equal to 1, if $Y > 0$ and $y \neq r_N$, and 0, otherwise. If $k \leq D - Y + \Psi$, then the sequence $ry^k$ is decoded to a sequence that begins with $p_1$.

**Proof:** Note that $d$ is a sequence consisting of alternating symbols. Suppose $k \leq D - Y + \Psi$. The proof is by natural induction on $D$. If $D = 1$, then $d_1 = p_1 = r_N$ and $\Psi = 0$. Since $k_{\Psi_1} \geq 2$ when $y \neq p_1$, the result of the decoding of $ry$ is equal to $p_1y$.

Suppose $D > 1$. Let $y = r_N$, then $y$ is the last symbol of $d$, and $t_N = y$. Let $n, 1 < n \leq N$ be such that $t_n = \ldots = t_N = y$, and $t_{n-1} \neq y$. This means that the sending of symbol $y$ started at transmission $n$, and all occurring errors were corrected after transmission $N$. Note that this implies that the two alternating symbols of sequence $d$ are $y$ and $r_{n-1}$. Suppose that $r_1 \ldots r_{N}y^k$ is decoded to $y^{k'}\gamma$, where $k' \geq 0$ is maximally chosen, for some $y', y' \in \mathcal{Y}$, and some sequence $\gamma$. Suppose that $r_1 \ldots r_{n-1}$ is decoded to $d'$. Note that $d = d'\gamma$. Let $Y'$ be the number of times that $y'$ occurs in $d'$. Let $\Psi'$ be equal to 1, if $Y' > 0$ and $y' \neq r_{n-1}$, and 0, otherwise. We would like to show that $k' \leq D' - Y' + \Psi'$, where $D'$ is the length of the sequence $d'$, so that the induction hypothesis can be applied. Since $\gamma$ does not influence the decoding of $r_1 \ldots r_{n-1}y^{k'}$, this will complete the proof.

Suppose $y = y'$, then $r_1 \ldots r_{N}y^k$ is decoded to $y^{k'+1}$, and $\Psi = 0$. If $D' = 1$, then $y = 1$ and $k \leq 1$, so $r_1 \ldots r_{N}y^k$ is decoded to $p_1y^{k'+1}$ because $k_{\Psi_1} \geq 3 > k + 1$. If $D' > 1$, then $Y' > 0$ and $\Psi' = 1$, so $k' = k + 1 \leq D - Y + \Psi + 1 = D' - (Y' + 1) + 1 = D' - Y' + \Psi'$. Therefore we assume w.l.o.g. $y \neq y'$.

Suppose $y = y'$, then $k' = 1$, $k < k_{\Psi_1}$, and $\gamma = y^k$. Because $y' \neq r_{n-1}$, it follows that $D' - Y' \geq 1 = k'$, so we assume w.l.o.g. $y' \neq y$, and consequently $\Psi' = 0$.

Let $e_\psi$ be the number of transmissions $\nu$, $n \leq \nu < N$, such that $r_\nu = \psi$ ($\psi \in \mathcal{Y}$). Each erroneous transmission $y \rightarrow \psi$ is followed by exactly $k_{\psi} - 1$ correct transmissions of symbol $y$. Transmission $N$ then completes all correction sequences. Therefore $e_\phi = \sum_{\psi \in \mathcal{Y}, \psi \neq \phi} e_{\phi'(k_{\psi'} - 1)}$.

Assume the worst case: $r_\nu = y'$ for $n \leq \nu < w$, and $r_\nu = y$ for $w \leq \nu \leq N$, for some $w \in \{n+1, \ldots, N\}$. Then $e_\phi = N - w$, $e_{y'} = w - n$, and $e_\phi = e_{y'}(k_{y'} - 1)$. The number of received $y$'s that are absorbed by $y^k$ during decoding of $r_1 \ldots r_{N}y^k$ is at most $(k - 1) \div (k_{y} - 1)$. So the number $R$ of (remaining) $y$'s satisfies $R = \max\{0, e_\phi + 1 - (k - 1) \div (k_{y} - 1)\}$. The number of received $y'$-s that is absorbed by the (remaining) $\psi$'s is equal to $(R - 1) \div (k_{\psi'} - 1)$. Three cases are considered:

$R > 0$. In this case $k' = e_{y'} - (R - 1) \div (k_{\psi'} - 1)$, so $(k_{y} - 1)(k_{\psi'} - 1)k' \leq (k_{y} - 1)(e_{y'} + 1 - R)$.

Since $R = e_{y'} + 1 - (k - 1) \div (k_{y} - 1)$, it follows that $(k_{y} - 1)(k_{\psi'} - 1)k' \leq k - k_{y} - 1$.

$y' \neq y$ and $R = 0$. Here $k' \leq e_{y'}$. Since $e_{y'} + 1 \leq (k - 1) \div (k_{y} - 1)$, it follows that $(k_{y} - 1)(k_{\psi'} - 1)k' \leq (k_{y} - 1)e_{y'} \leq k - k_{y} - 1$.

$y' = y$ and $R = 0$. We have $k' = e_{y'} + k - (e_{y'} + 1)(k_{y} - 1)$. Since $e_{y'} \leq e_{\phi}$, it follows that $k' \leq k - k_{y} - 1$. 

51
To complete the induction step two cases are considered:

\( y = r_{n-1} \). In this case \( Y > 0, \Psi = 1, \) and \( k_{yy} \geq 3. \)

Suppose \( y' = y, \) then \( k_{yy'} \geq 3 \) and \( Y' = Y. \) If \( R > 0 \) then \( 4k' \leq k - 1. \) If \( R = 0 \) then \( k' \leq k - 2. \) So surely \( k' \leq k - 2. \) So if \( y' = y, \) then \( k' \leq D - Y + \Psi - 2 = D' - Y' + \Psi'. \)

Suppose \( y' \neq y, \) then \( Y' = 0 \) because we assumed \( y' \neq \emptyset. \) If \( R > 0 \) then \( 2k' \leq k - 1. \) If \( R = 0 \) then \( 2k' \leq k - 3. \) So surely \( k' \leq k - 2 < D' - Y' + \Psi'. \)

\( y \neq r_{n-1}. \) Since we assumed that \( y \neq \emptyset, \) we have \( Y = \Psi = 0. \)

Suppose \( y' = y, \) then \( Y' = 0. \) So \( k' \leq k - 1 \leq D - Y + \Psi - 1 = D' - Y' + \Psi'. \)

Suppose \( y' = r_{n-1}, \) then \( Y' = D \div 2 \) and \( k_{yy'} \geq 3. \) If \( R > 0 \) then \( 2k' \leq k - 1. \) If \( R = 0 \) then \( 2k' \leq k - 2. \) So certainly \( 2k' \leq k - 1 \leq D - 1. \) If \( D \mod 2 = 0, \) then \( 2k' \leq D - 2, \) because \( k' \) is an integer, so \( k' \leq \frac{1}{2}D - 1 = D' - Y' + \Psi'. \) If \( D \mod 2 = 1, \) then \( Y' = D'/2 \) so \( k' \leq D'/2 = D' - Y' + \Psi'. \)

Finally suppose \( y' \neq y \) and \( y' \neq r_{n-1}, \) then \( Y' = 0. \) So \( k' \leq k - 1 \leq D - 1 = D' - Y' + \Psi'. \)

In each case \( k' \leq D' - Y' + \Psi'. \)

The next lemma is a generalization of Lemma 2.1 for the binary symmetric case. The result of Lemma 2.2 is used in the proof. As in the binary symmetric case, for \( L = 1 \) even equality in Equation 2.2 is sufficient.

**Lemma 2.3** Let \( L = 1. \) If \( \sum xy e_{xy} k_{xy} < N, \) then \( \hat{p} = p. \)

**Proof:** Note that \( c \) is a sequence consisting of alternating symbols. Suppose \( \sum xy e_{xy} k_{xy} < N. \) The proof is by natural induction on the number of errors \( \sum xy e_{xy}. \) If no errors occurred during the transmission, then \( c = r, \) so \( \hat{p} = p. \) Suppose that \( \sum xy e_{xy} > 0. \)

Suppose no forbidden subsequence occurs in \( r. \) Then \( d = r \) and \( \hat{p}_1 = r_1, \) and furthermore, an \( x \rightarrow y \) error is directly followed by less than \( k_{xy} \) error-free transmissions (of symbol \( x \)). Assume \( \hat{p} \neq p, i.e. r_1 \neq t_1. \) If the last error was an \( \chi \rightarrow \psi \) error, then it occurred not later than transmission \( \sum xy k_{xy} e_{xy} - k_{x\psi} + 1. \) Therefore, there are at least \( N - (\sum xy k_{xy} e_{xy} - k_{x\psi} + 1) \geq k_{x\psi} \) correctly received symbols after the last error, as depicted in Figure 2.6, where \( \chi = 0 \) and \( \psi = 2. \) This means that the last error is corrected, and a forbidden subsequence occurs in \( r, \) which is a contradiction. It follows that \( \hat{p} = p. \)

Suppose a forbidden subsequence occurs in \( r. \) Let \( r_n \ldots r_{n+k_{xy}} = yx^{k_{xy}} \) be the rightmost forbidden subsequence of \( r. \) Four cases are distinguished:

\( r_n \neq t_n \) and \( r_{n+1} = t_{n+1}. \) In this case \( t_n = \ldots = t_{n+k_{xy}} = x. \) Let \( t' = t_1 \ldots t_{n-1} t_{n+k_{xy}} \ldots t_N, \) and \( r' = r_1 \ldots r_{n-1} r_{n+k_{xy}} \ldots r_N, \) then \( t' \) and \( r' \) represent a transmission of \( p \) with a multiple-repetition strategy. Since the substitution \( r_n \ldots r_{n+k_{xy}} \rightarrow x \) eliminates one \( x \rightarrow y \) transmission error, the induction hypothesis can be applied to \( t' \) and \( r', \) so \( \hat{p} = p. \)
\( r_n \neq t_n \) and \( r_{n+1} \neq t_{n+1} \). Here \( t_n = \ldots = t_N = \tau \) for some \( \tau \), \( x \neq \tau \neq y \). The same reduction as in the previous case to \( t' \) and \( r' \) eliminates \( k_{xy} - 1 \) errors of type \( \tau \rightarrow y \), and one error of type \( \tau \rightarrow y \). Since \((k_{xy} - 1)k_{rx} + k_{ry} \geq k_{xy}\), the induction hypothesis can be applied to \( t' \) and \( r' \), so \( \hat{p} = p \).

\( r_n = t_n \) and \( r_{n+1} = t_{n+1} \). We have \( t_n = y \), \( t_{n+1} = x \), and \( t_{n+2} = \ldots = t_N = \tau \) for some \( \tau \), \( \tau \neq x \). Because the tail is alternating, \( \tau = y \). The same reduction as in the previous cases to \( t' \) and \( r' \) eliminates \( k_{xy} \) errors of type \( y \rightarrow x \). Since \( k_{xy} \geq 3 \), and \((k_{xy} - 2)k_{yx} \geq (k_{xy} - 2)3 \geq k_{xy}\), the induction hypothesis can be applied to \( t' \) and \( r' \), so \( \hat{p} = p \).

\( r_n = t_n \) and \( r_{n+1} \neq t_{n+1} \). We have \( t_n = y \), and \( t_{n+1} = \ldots = t_N = \tau \) for some \( \tau \), \( \tau \neq x \). Write \( r = r_1 \ldots r_{n-1}yx^kr_{n+k+1} \ldots r_N \) such that \( k \) is maximal.

Suppose there exists a \( \nu \), \( n + k + 1 \leq \nu \leq N \), such that \( r_\nu \neq \tau \). Take the last such \( \nu \), then \( r_{\nu+1} = \ldots = r_N = \tau \). Since \( r_n \ldots r_{n+k_{xy}} = yx^k \) is the rightmost forbidden subsequence of \( r \), we have \( N - \nu < k_{xy} \). Let \( t' = t_1 \ldots t_{\nu-1} \) and \( r' = r_1 \ldots r_{N-1} \), then \( t' \) and \( r' \) represent a transmission of \( p \) with a multiple repetition strategy. Since the reduction of \( t \) and \( r \) to \( t' \) and \( r' \) eliminates one \( \tau \rightarrow r_\nu \) error, the induction hypothesis can be applied to \( t' \) and \( r' \). Let \( d' \) be the result of the decoding of \( r' \) from right to left, then \( d' \) ends with the sequence \( r'_{n+k} \ldots r'_{\nu-1} \), because \( r'_{n+k} \ldots r'_{N-1} \) contains no forbidden subsequence, and \( r'_{n+k} = x \neq r'_{n+k+1} \). For the same reason, \( d = d' \rho_\nu \ldots r_N \), so \( \hat{p} = p \). Therefore we assume w.l.o.g. that \( r_{n+k+1} = \ldots = r_N = \tau \), and consequently \( N - n - k < k_{rx} \).

Suppose \( \tau = y \), then the same reduction as in the previous cases eliminates \( k_{xy} - 1 \) errors of type \( y \rightarrow x \). Since \( k_{xy} \geq 2 \), and \((k_{xy} - 2)k_{yx} \geq (k_{xy} - 2)2 \geq k_{xy}\), the induction hypothesis can be applied, so \( \hat{p} = p \). So we assume w.l.o.g. that \( \tau \neq y \).

Then \( t_{n+1} \neq t_n \), so all transmission errors that occurred during the first \( n \) transmissions are corrected. Since the decoding of \( r_1 \ldots r_N \) is not influenced by \( r_{n+k+1} \ldots r_N \), it is sufficient to show that Lemma 2.2 can be applied to \( t_1 \ldots t_n \) and \( r_1 \ldots r_n \), when \( x^k \) is appended to \( r_1 \ldots r_n \). Let \( d' \) be the result of the decoding of \( r_1 \ldots r_n \) from right to left. Since \( \tau \neq x \), \( y \neq x \), and the tail consists only of symbols \( y \) and \( \tau \), the symbol \( x \) does not occur in \( d' \). It remains to show that \( k \leq D' \), where \( D' \) is the length of sequence \( d' \). Let \( e'_{xy} \) be the number of \( x \rightarrow y \) errors that occurred during the first \( n \) transmissions. We derive \( D' = n - \Sigma_{xy} e_{xy} k_{xy} = n - (\Sigma_{xy} e_{xy} k_{xy} - k \cdot k_{rx} + 1 - N) \geq n + k \cdot k_{rx} + 1 - (n + k + k_{rx} - 1) = 1 + (k - 1)(k_{rx} - 1) \geq k \), because \( k_{rx} \geq 2 \). It follows that \( \hat{p} = p \).

For each case \( \hat{p} = p \).

The result of Lemma 2.3 is used to proof the next theorem which generalizes Theorem 2.1.

**Theorem 2.2** If \( \Sigma_{xy} e_{xy} k_{xy} < N - L \), then \( \hat{p} = p \).

**Proof:** Suppose \( \Sigma_{xy} e_{xy} k_{xy} < N - L \). Let \( e_{xy} \) and \( e'_{xy} \) be the number of \( x \rightarrow y \) errors that occurred during the transmission of respectively, the information part \( p \), and it's tail \((e_{xy} = e_{xy} + e'_{xy})\). Each \( x \rightarrow y \) transmission error in the information part leads to \( k_{xy} \) extra repetition symbols, and a reduction of the tail by \( k_{xy} \). So it takes \( L + \Sigma_{xy} e_{xy} k_{xy} \) transmissions to transmit the information part such that all occurring errors are corrected,
and there are \( N' = N - (L + \sum_{xy} e_{xy}k_{xy}) > \sum_{xy} e_{xy}k_{xy} \) transmissions left to transmit the remaining tail. From Lemma 2.3 follows that when the remaining tail is transmitted, the received remaining tail \( r_{N'-N'+1} \ldots r_N \) is decoded to a sequence \( \delta \) such that \( \delta_1 \) equals the first symbol of the remaining tail (which is \( c_{L+1} \) and also equals \( t_{N'-N'+1} \)). The situation is illustrated in Figure 2.7.

Since \( r_1 \ldots r_{N'-N'} \) is decoded to \( p \), and \( r_{N'-N'+1} \ldots r_N \) is decoded to \( \delta \), it is sufficient to show that \( \delta \) does not affect the decoding of \( r_1 \ldots r_{N'-N'} \) so that \( r \) is decoded to \( p \delta \). Since \( r_{N'-N'} = c_L \neq c_{L+1} \), this is only possible when \( \delta \) begins with \( c_{L+1} \). In that case, the sequence \( r_{N'-N'} \ldots r_N \) would be decoded to a sequence that begins with \( c_{L+1} \). But because \( t_{N'-N'} = r_{N'-N'} = c_L \), Lemma 2.3 can be applied to deduce that \( r_{N'-N'} \ldots r_N \) is decoded to a sequence that begins with \( c_L \), which would be a contradiction. It follows that \( r \) is decoded to \( p \delta \) and consequently \( \hat{r} = p \).

### 2.3 Conclusions

We described how the idea of repetition feedback coding can be used for block coding on arbitrary discrete memoryless channels with a noiseless delayless feedback link. The error correcting capabilities are characterized by Equation 2.2. A suitable tail has to be used to achieve these error correcting capabilities. Four Tail Conditions were shown to be correct: when a tail is constructed that satisfies any of these Tail Conditions, the error correcting capabilities are achieved. For some choices of the repetition parameters, a tail can be constructed that satisfies Tail Condition 2, while no (long) tails can be constructed that satisfy any of the other three Tail Conditions. Consequently, we reject the Conjecture made by Becker [7, p. 73] that his tail Conditions can not be further relaxed. We indicated that our four Tail Conditions are not easily further relaxed by rejecting an obvious extension.
Chapter 3

Recursive coding

In subsection 1.4.2 it is shown how repetition feedback coding can be used in a recursive way for the BSC. These ideas are generalized to arbitrary DMC’s with feedback in this Chapter. Furthermore, the code word estimator’s error exponent is determined for the general case.

3.1 Binary symmetric recursive coding

In this Section the results of Schalkwijk and Post [72, 73], as described earlier in subsection 1.4.2, are recapitulated. Let \( p \) be the channel error probability of the BSC (\( X = Y = \{0,1\} \)). Let \( k \geq 3 \) be the repetition parameter. In order to make a possible generalization to arbitrary DMC’s with feedback easier, we assume \( a = 1/2 \) (i.e. we consider only median paths), which implies that the transmission rate is maximal (see Equation 1.71). We also assume that an infinitely long binary precoded sequence \( p \) is given. This assumption will be discussed later.

The transmitter sends a precoded sequence and each time a channel error occurs, the erroneously received symbol is repeated \( k \) times. The main difference with block coding is the way of decoding the received sequence by the receiver.

The transmitted and the received sequence are denoted by \( t \) and \( r \) respectively. The decoding of \( r \) is done in two stages. In the first stage the receiver obtains an estimate \( \hat{t} \) of the transmitted sequence. Schalkwijk and Post called it the code word estimator, but “transmitted sequence estimator” would be more appropriate. In the second stage, the sequence \( \hat{t} \) is transformed to an estimate \( \hat{p} \) of the precoded sequence. The second stage, which was called the inverse encoder, can be seen as the inverse operation of the channel encoder: the receiver works through the sequence \( \hat{t} \) from left to right, and for each bit \( \hat{t}_n \) such that \( \hat{t}_n \neq r_n \), the receiver deletes \( k \) not previously deleted bits on the right-hand side of \( \hat{t}_n \) (provided the bits to be deleted equal \( \hat{t}_n \), otherwise a first stage decoding error is detected). In this way, \( \hat{p} \) is equal to \( p \) when \( \hat{t} = t \). An example for \( k = 3 \) is depicted in Figure 3.1. During the first stage, the receiver computes an estimate for each symbol of the transmitted sequence. More precisely, the receiver works through the sequence \( r_n \ldots r_{n+D} \) from left to right to obtain an estimate \( \hat{t}_n \) of \( t_n \). The number \( D \) (not to be confused with the length of the decoded sequence in case of block coding) denotes the delay, which may be variable, i.e. dependent on \( r_n, r_{n+1}, \ldots \), or fixed. These two decoding techniques will be treated separately. In the binary symmetric case, the estimation algorithm is described by a state diagram. The state diagram for \( k = 3 \) is depicted in Figure 3.2. The estimation algorithm starts in state 0. If \( r_n = 1 \), the state moves up, and if \( r_n = 0 \), it moves down,
Received sequence

Estimated transmitted sequence

Estimated precoded sequence

Figure 3.1: Inverse encoding

all along the arrows. The receiver continues similarly with the succeeding received symbols. An overview of the complete system is depicted in Figure 3.3.

Let \( p_k \) be the solution \( 0 < p < \frac{1}{2} \) of

\[
2p(2-2p)^{k-1} = 1. \tag{3.1}
\]

The idea behind the estimation algorithm is the moving value of the median in median feedback as described earlier in subsection 1.4.2. When a 1 is received, the median moves up, and when a 0 is received, the median moves down (see Figure 1.12). Furthermore, for \( p = p_k \), a movement away from the centre, followed by \( k-1 \) movements towards the centre, brings the median back to the same value. To obtain this regular median path behaviour, Schalkwijk and Post [72, 73] assumed that \( p = p_k \), and thereupon were able to compute the error probability of their estimation algorithm. In Section 3.2 these results are generalized to arbitrary channel error probability \( p \).

3.1.1 Fixed decoding delay

When the delay \( D \) is fixed, the estimation algorithm takes exactly \( D+1 \) steps. The end-state is determined by the sequence \( r_n \ldots r_{n+D} \). If the end-state is positive, the receiver decides for \( i_n = 1 \), and if the end-state is negative, the receiver decides for \( i_n = 0 \). If the end-state is the 0-state, a fair coin is tossed. Let \( P_e^f \) be the symbol error probability \( \Pr\{i_n \neq \hat{i}_n\} \) of the code word estimator. The asymptotic symbol error exponent \( E^f \) was shown [72] to satisfy:

\[
E^f = \lim_{D \to \infty} -\frac{\log_2 P_e^f}{D} = D \left( \frac{1}{k} \| p_k \right), \tag{3.2}
\]

although Schalkwijk and Post did not find the notation in terms of binary informational divergence (note that \( D \left( \frac{1}{k} \| p_k \right) \right) = D \left( \frac{1}{k} \| \frac{1}{2} \right) \) in conformity with Equation 1.72). Furthermore, an upper bound was obtained for finite delay:

\[
P_e^f \leq \frac{1}{k-2} \cdot 2^{-E^f \cdot D}, \tag{3.3}
\]

which means that the asymptotic exponent also gives a good upper bound for finite delay.

3.1.2 Variable decoding delay

In this case the number of steps of the estimation algorithm is not fixed. The receiver decides upon \( i_n \) when a certain state threshold \( s > 0 \) is reached, i.e. when a state is reached equal to or larger than \( s \), the receiver decides for \( i_n = 1 \), and when a state is reached equal to or smaller than \( -s \), the receiver decides for \( i_n = 0 \). The number of steps of the estimation algorithm until a state threshold is reached depends on the received sequence \( r_n r_{n+1} \ldots \)
Figure 3.2: State diagram for $k = 3$
Figure 3.3: Overview of multiple-repetition recursive coding
Let $P^v$ be the symbol error probability $Pr\{t_n \neq \hat{t}_n\}$ of the code word estimator. Let $w_k$ be the solution $w$, $0 < w < 1$ of $w^k - 2w + 1 = 0$. It is easily verified that $w_k = 1/(2 - 2p_k)$. The symbol error probability was shown [72] to satisfy:

$$P^v = \frac{1}{2} \cdot w_k^s. \quad (3.4)$$

While calculating $P^v$, Schalkwijk and Post forgot that the estimation algorithm does not necessarily end in state $s$, but could also end in states $s + 1$ up to $s + k - 1$. Therefore, it turns out that $P^v = c_k(s) \cdot \frac{1}{2} \cdot w_k^s$ for some constant $c_k(s)$, $w_k^{k-1} \leq c_k(s) \leq 1$. However, since Equation 3.4 is only used to compute the limiting value of $(\log P^v)/s$ when $s$ increases, all further deductions remain valid.

The average number of steps $\bar{D}(s)$ to reach state threshold $s$ was shown to satisfy:

$$\bar{D}(s) = \frac{2}{k-2} \cdot (s-1) + \mathcal{O}(1) \quad (s \to \infty), \quad (3.5)$$

where $\mathcal{O}(1)/s \to 0$ for $s \to \infty$. Therefore, the asymptotic symbol error exponent $E^v$ equals

$$E^v = \lim_{s \to \infty} \frac{-\log_2 P^v}{\bar{D}(s)}$$

$$= \frac{k-2}{2} - \log_2 w_k \quad (3.7)$$

$$= D\left(\frac{1}{2} \parallel p_k\right). \quad (3.8)$$

From Equation 3.8 and Equation 3.2 follows automatically that $E^v > E^f$ as was expected.

### 3.2 Recursive coding for arbitrary channels

A recursive coding scheme is presented with fixed delay for arbitrary discrete memoryless channels, using the ideas of multiple-repetition feedback coding, thereby generalizing the results of Schalkwijk and Post [72, 73] from subsection 3.1.1. Although the theoretical results are valid for all DMC's, the reader is reminded that for some type of channels multiple-repetition strategies are less suitable (see subsection 1.4.4).

Consider a discrete memoryless channel with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and channel probabilities $p_{xy}$ ($x \in \mathcal{X}, y \in \mathcal{Y}$). The repetition parameters are denoted by $k_{xy}$ ($x \in \mathcal{X}, y \in \mathcal{Y}, x \neq y$). We assume that an infinitely long $\mathcal{X}$-ary precoded sequence $p$ is given. This assumption will be discussed later. For asymmetric channels it is useful for the precoder to fix the symbol precoding distribution in order to increase the transmission rate (see Section 4.1).

The transmitter sends a precoded sequence and each time an $x \rightarrow y$ channel error occurs, the transmitter repeats $k_{xy}$ times the symbol $x$. The transmitted and the received sequence are denoted by $\hat{x}$ and $\hat{r}$ respectively. As in the binary symmetric case, the decoding of $\hat{r}$ is done in two stages. In the first stage the receiver obtains an estimate $\hat{t}$ of the transmitted sequence. In the second stage, the sequence $\hat{t}$ is transformed to an estimate $\hat{p}$ of the precoded sequence. The second stage can be seen as the inverse operation of the channel encoder: the receiver works through the sequence $\hat{t}$ from left to right, and for each digit $\hat{t}_n$ such that $\hat{t}_n \neq r_n$, the receiver deletes $k_{\hat{t}_n r_n}$ not previously deleted digits on the right-hand side of
in. In this way, \( \hat{p} \) is equal to \( p \) when \( \hat{t} = t \). An example is depicted in Figure 3.4, where \( X = Y = \{0, 1, 2\} \) and \( k_{12} = 3 \). As usual, a white square denotes a 0, a grey square denotes a 1, and a black square denotes a 2.

During the first stage of decoding, the receiver obtains an estimate for each symbol of the transmitted sequence. Only the case of fixed delay \( D \) is considered. Since the repetition parameters can take different values, and the input alphabet is not necessarily binary, it seems impossible to use a left-to-right estimation algorithm as described in Section 3.1. Several attempts to find a general left-to-right estimation algorithm failed, even when using more dimensional state diagrams. Instead, we use the right-to-left substitution algorithm (as was done in case of block coding): the receiver scans the sequence \( r_n \ldots r_{n+D} \) from right to left and substitutes each subsequence \( yx^{k_{xy}} \) by \( x \). This is repeated until no forbidden subsequences occur anymore. The first symbol of the resulting sequence is the estimate \( \hat{t}_n \).

An example of the first stage decoding algorithm is depicted in Figure 3.5, where \( X = Y = \{0, 1, 2\} \), \( D = 9 \), \( k_{02} = 3 \) and \( k_{01} = 4 \). An overview of the recursive system is depicted in Figure 3.3.

### 3.2.1 Comparison with the binary symmetric estimation algorithm

One might think that a left-to-right estimation algorithm would be more efficient than an estimation algorithm that works from right to left, since the input data are also received from left to right. However, this is only true in an implementation where \( D \) parallel processors are available that each keep track of the state of a state diagram corresponding to a transmitted symbol that is to be estimated. When the adjustment of the states caused by a new received symbol is not done in parallel, the \( D \) new states each time have to be computed separately. When the right-to-left substitution algorithm is used, a sequence of length \( D \) has to be scanned. When the length of the most recent run of equal symbols is kept during the scan, one scan is sufficient, so the number of computation steps is also equal to \( D \). Consequently,
when parallel computations are not available, a right-to-left substitution algorithm is equally efficient as a left-to-right state-based algorithm. In terms of memory complexity, a right-to-left substitution algorithm requires the storage of \( D \) output symbols, while during a left-to-right estimation algorithm \( D \) different diagram states have to be kept. The latter seems to require more amount of storage.

However, a left-to-right state-based algorithm is more easily generalized to an efficient code word estimator with variable delay. For a variable-delay estimator it is necessary to have an indication of the reliability of the current estimate after each transmission. In a state-based algorithm, the reliability increases with the value of the state. In a right-to-left substitution algorithm the number of symbols that remain after all substitutions have been made fulfills the same role, but it would not be efficient to scan and substitute the complete sequence after each transmission.

It is shown that in the binary symmetric case, the left-to-right estimation algorithm and the right-to-left substitution algorithm are essentially equivalent in the sense that they produce the same estimate.

**Theorem 3.1** Suppose \( \mathcal{X} = \mathcal{Y} = \{0, 1\} \) and \( k_{01} = k_{10} = k \). We are given the received (sub)sequence \( r_n \ldots r_{n+D} \). The estimate produced by the left-to-right estimation algorithm is denoted by \( i^1_n \). The estimate obtained by the right-to-left estimation algorithm is denoted by \( i^2_n \). If the left-to-right estimation algorithm does not end in state 0 after having processed the \( D \) received symbols, then \( i^1_n = i^2_n \).

**Proof:** The proof is by natural induction on the delay \( D \). If \( D = 0 \), then \( i^1_n = i^2_n = r_n \), so suppose \( D > 0 \).

If no forbidden subsequence occurs in \( r_n \ldots r_{n+D} \), then \( i^2_n = r_n \). Assume w.l.o.g. \( r_n = 1 \). If \( r_n \ldots r_{n+D} \) is a concatenation of sequences \( 10^k-1 \), then the left-to-right algorithm ends in state 0. Otherwise, the left-to-right algorithm will end in a positive state and obtain \( i^1_n = 1 \). In each case the assertion holds.

Suppose that \( r_n \ldots r_{n+D} \) has a forbidden subsequence. Let \( r_{\nu} \ldots r_{\nu+k} \) be the rightmost forbidden subsequence. Let \( s_1 \) and \( s_2 \) be the states of the left-to-right algorithm after input \( r_n \ldots r_{\nu-1} \). Let \( s_1 \) and \( s_2 \) be the states of the left-to-right algorithm after inputs \( r_n \ldots r_{\nu-1} r_{\nu+k} \) and \( r_n \ldots r_{\nu+k} \) respectively. We have to show that the estimate of the left-to-right algorithm is not influenced by the substitution \( r_{\nu} \ldots r_{\nu+k} \rightarrow r_{\nu+k} \). Assume w.l.o.g. \( r_\nu = 0 \). If \( s \leq 0 \), then \( s_1 = s_2 = s + 1 \). The induction argument then completes the proof. If \( s > 0 \), then \( s_1 = s + k - 1 \) and \( s_2 = s - 1 + k(k - 1) \). Both states are positive, and since the sequence \( r_{\nu+k+1} \ldots r_{n+D} \) does not contain a forbidden subsequence, the end-state of the left-to-right algorithm will be positive in either case.

Because the decoding algorithms are essentially equivalent in the binary symmetric case, their symbol error probabilities are nearly the same. It can be shown that the results described by Equations 3.2 and 3.3 also hold for the substitution algorithm when \( \mathcal{X} = \mathcal{Y} = \{0, 1\} \), \( k_{01} = k_{10} = k \) and \( p_{01} = p_{10} = p_k \).

### 3.2.2 Asymptotic error exponent

In this subsection the asymptotic error exponent of the general fixed delay code word estimator is computed. For convenience the numbers \( k_x (x \in \mathcal{X}) \) are defined as 0. Remember that all logarithms are to the base \( |\mathcal{X}| \).
Figure 3.6: Alternative estimation error

Let \( x \in \mathcal{X} \). Assume w.l.o.g. that \( x \in \mathcal{Y} \). Consider the infinite Markov chain with states 0, 1, 2, …, and state transition probabilities \( \Pr\{s + k_{xy} - 1 \mid s\} = p_{xy} \) for \( y \in \mathcal{Y} \) and \( s > 0 \). This Markov chain represents the error correction process initiated by an incorrectly received \( x \)-symbol. The process starts in state \( k_{xy} \) in case an \( x \rightarrow y \) error occurred and stops when arriving at state 0: “the error is corrected”. When \( \sum_{y \in \mathcal{Y}} p_{xy} k_{xy} < 1 \), the process always will eventually end up in state 0. The process does not necessarily end when \( \sum_{y \in \mathcal{Y}} p_{xy} k_{xy} \geq 1 \).

Let \( s > 0 \) be an arbitrary state. Let \( p^*_n \) be the probability of going from state \( s \) to state 0 in exactly \( n \) steps. In appendix A.1 the asymptotic exponent of \( p^*_n \) is computed:

\[
E_1(x) = \lim_{n \to \infty} -\log \frac{p^*_n}{n} = \log w_x. \tag{3.9}
\]

The number \( w_x \) equals \( (\sum_{y \neq x} p_{xy} k_{xy} c_{xy}^{k_{xy} - 1})^{-1} \), where \( c_x > 0 \) is the solution of \( p_{xx} = \sum_{y \neq x} p_{xy} (k_{xy} - 1) c_x^{k_{xy}} \). Note that in case of constant repetition parameters: \( k_{xy} = k_x \) for each \( y \neq x \), the asymptotic exponent \( \log w_x \) equals \( D(\frac{1}{k_x} || 1 - p_{xx}) \), where \( D \) is the informational divergence, which is comparable with Equation 3.2.

Let \( P_r \) be the symbol error probability \( \Pr\{t_n \neq \hat{t}_n\} \) of the code word estimator. Since an estimation error could be caused by an unfinished error correction process initiated at transmission \( n \), we have \( \Pr\{t_n \neq \hat{t}_n \mid t_n = x\} \geq \sum_{y \neq x} p_{xy} k_{xy}^{k_{xy}} \). It follows that the asymptotic symbol error exponent satisfies:

\[
E = \lim_{D \to \infty} -\log \frac{P_r}{D} \leq E_1 = \min_{x \in \mathcal{X}} E_1(x). \tag{3.10}
\]

However, an estimation error is not necessarily caused by an unfinished error correction process initiated at transmission \( n \). This is illustrated by an example in Figure 3.6, where \( \mathcal{X} = \mathcal{Y} = \{0, 1, 2\} \), \( k_{10} = 3 \), and \( D = 4 \). This alternative type of estimation error can occur when the precoded sequence contains a flip sequence, i.e. a sequence that equals \( y_1 x^{k_{y_1} - 1} y_2 x^{k_{y_2} - 1} \cdots y_n x^{k_{y_n} - 1} \) for some \( x \in \mathcal{X} \) and \( y_n \neq x \) (1 \( \leq n \leq \nu \)). When such a flip sequence is followed by an (erroneous) \( x \), the right-to-left estimation algorithm produces the estimate \( x \) instead of \( y_1 \) (the flip sequence “collapses” into a single \( x \)). This type of estimation error could be avoided when the precoder does not allow sequences that contain flip sequences of a certain length, but that would also reduce the transmission rate (see e.g. subsection 7.2.5 of this thesis). In appendix A.2 the asymptotic exponent for the probability of this type of estimation error is calculated:

\[
E_2(x) = \log \frac{c_{yx}}{p_{xx}}, \tag{3.11}
\]
where \( \log c \) is the amount of information per precoded symbol. In Section 4.1 it is shown how \( c \) can be computed. The numbers \( \gamma_x > 0 \) are the solutions of \( \sum_{y \neq x} p_{xy} \gamma_x^{k_y} = p_{xx} \). The exponent \( E_2(x) \) is of course an upper bound for \( E \):

\[
E \leq E_2 = \min_{x \in X} E_2(x). \tag{3.12}
\]

In Figure 3.7 the upper bounds \( E_1 \) and \( E_2 \) are depicted for the binary symmetric channel with \( k = 3 \).

Nevertheless, we suspect that equality holds in Equation 3.10, provided that the repetition parameters are suitably chosen (see Chapter 5 how to choose the repetition parameters). E.g. for the binary symmetric channel this would be the case when \( p \approx p_k \), where \( p_k \) is the solution of Equation 3.1, or in other words \( k \approx -\log_2 p \) (see Chapter 5). This opinion is strengthened by the fact that in the symmetric case: \( p_{xy} = p \) and \( k_{xy} = k \) for each \( y \neq x \), the upper bound in 3.10 equals the right-hand side of 3.2 when \( 1 - p_{xx} = (|Y| - 1) p = p_k \), and thereby establishes equality in 3.10 due to Theorem 3.1. This leads to the following Conjecture:

**Conjecture 3.1** When the repetition parameters are suitably chosen, the asymptotic error exponent of the code word estimator satisfies

\[ E = E_1. \]

### 3.3 Conclusions

We described how the idea of repetition feedback coding can be used for recursive coding on arbitrary discrete memoryless channels with a noiseless delayless feedback link.

A recursive repetition feedback strategy obtained by Schalkwijk and Post was set out. Their results are suitable for the binary symmetric channel. When a binary symmetric repetition strategy is used with repetition parameter \( k \), the cross-over probability \( p \) has to
satisfy $p = p_k$, where $p_k$ is the solution of Equation 3.1. The first stage decoding is done by a left-to-right estimation algorithm. The delay can be fixed or variable. For both cases decoding error probabilities were calculated.

A recursive repetition feedback strategy with fixed delay was presented that can be used for discrete memoryless channels with arbitrary channel probabilities. The first stage decoding is done by a right-to-left estimation algorithm. For the binary symmetric case this algorithm was shown to be closely related to the left-to-right estimation algorithm. Two upper bounds were calculated for the asymptotic decoding error exponent, one of them conjectured to be tight when the repetition parameters are suitably chosen. We believe that the discovery of a general left-to-right estimation algorithm (if it exists) would also lead to an easier analysis and eventually to an exact determination of the error exponent.

We assumed that an infinitely long precoded sequence was available without worrying about the practical realization. A thorough investigation and solution of this problem would go beyond the scope of this thesis, but we would like to mention a possible solution which is also described in subsection 7.2.6 of this thesis. One could construct a sliding block code [1] by means of a finite state automaton that outputs the required constrained sequences from an unconstrained input. Much work has been done in this area. Although a slight loss in precoding rate is unavoidable when using sliding block codes, it presents a way to reduce the error propagation, which brings us to the second (neglected) problem of recursive coding: we considered only the error probability $Pr\{t_n \neq \hat{t}_n\}$ of the code word estimator, and not the error probability $Pr\{p_n \neq \hat{p}_n\}$ of the inverse encoder, or even better the error probability $Pr\{m_n \neq \hat{m}_n\}$ of the inverse precoder (in case the message is an unconstrained sequence of independently and uniformly chosen symbols). When $\hat{t} = \hat{t}_n$, it is clear that $\hat{p} = \hat{p}_n$ and $\hat{m} = \hat{m}_n$, but when there is an error in the estimated transmitted sequence, this error could cause a large number of errors in the estimated precoded sequence during the inverse encoding, and later in the inverse precoding: the error propagates. A sliding block code can be used to reduce the problem of error propagation [1] during the postcoding. The precoding problem is further addressed in Section 7.2.
Chapter 4

Achieving channel capacity

In this Chapter the maximal achievable transmission rate of a multiple-repetition feedback strategy with given repetition parameters is determined. This applies to block coding as well as recursive coding. It is shown that there exists a discrete memoryless channel such that the achievable transmission rate equals the channel capacity. The results are illustrated by some example channels.

4.1 The transmission rate

Consider a discrete memoryless channel with a noiseless delayless feedback link, input symbol alphabet \( \mathcal{X} \), output alphabet \( \mathcal{Y} \), and channel probabilities \( p_{xy} (x \in \mathcal{X}, y \in \mathcal{Y}) \). W.l.o.g. \( \mathcal{X} \subseteq \mathcal{Y} \) because when \( |\mathcal{X}| > |\mathcal{Y}| \) it is possible to eliminate a suitably chosen input symbol without affecting the capacity of the channel [88]. Suppose a multiple-repetition feedback strategy is used with repetition parameters \( k_{xy} (x \in \mathcal{X}, y \in \mathcal{Y}, x \neq y) \). Since a received symbol \( y (y \in \mathcal{Y}, y \notin \mathcal{X}) \) is always erroneous, it is assumed that the repetition numbers \( k_{xy} (x \in \mathcal{X}, y \in \mathcal{Y} \setminus \mathcal{X}) \) are equal to 1 (see subsection 1.4.4). For reasons of convenience the numbers \( k_{xx} (x \in \mathcal{X}) \) are defined as 0.

Suppose block coding is used with block length \( N \) and \( L \) information symbols, like in Section 2.2. Let \( M \) be the number of messages. Then the amount of information that is transmitted over the channel per channel use, which is called the transmission rate, equals

\[
R^b = \frac{\log M}{N}.
\]  

Remember that all logarithms are to the base \( |\mathcal{X}| \). Since it is not yet clear that, given the repetition parameters, the maximal achievable transmission rate of a multiple-repetition block coding strategy equals the maximal achievable transmission rate of a multiple-repetition recursive coding strategy, the transmission rates are provisionally denoted by \( R^b \) and \( R^r \) respectively. The block error probability is \( P_e(N) = \Pr\{\sum_{xy} k_{xy} E_{xy} \geq N - L\} \) (see subsection 2.2.1), where the random variable \( E_{xy} \) denotes the number of \( x \rightarrow y \) channel errors during one block.

The concept of transmission rate only makes sense when the error probability is small so we introduce the notion of achievable rate (similar to Definition 1.2).

**Definition 4.1** A rate \( R^b \) is said to be achievable, if for each sufficiently large block length \( N \), it is possible to find \( M(N) \) (precoded) messages of length \( L \) such that \( R^b \leq \frac{\log M(N)}{N} \), and \( P_e(N) \) becomes arbitrarily small with increasing \( N \).
The supremum over all achievable rates is denoted by $\bar{R}^k$.

As an example consider the binary symmetric channel with cross-over probability $p$ on which a binary symmetric repetition feedback strategy is used with repetition parameter $k \geq 3$. Let $M_L$ be the number of binary sequences of length $L$ that do not contain subsequences $01^k$ and $10^k$. Because of the recurrent relation $M_L = 2M_{L-1} - M_{L-k}$ for $L > k$, it follows that $\frac{\log_2 M_L}{L}$ goes to $\log_2 c_k$ when $L$ increases, where $c_k$ is the solution $c > 1$ of $c^k = 2c^{k-1} - 1$. The error probability $P_e(N)$ equals $\Pr\{k \cdot E \geq N - L\}$, where $E$ is a binomially distributed random variable with parameter $p$ and expectation $pN$. Suppose we fix $\omega = \frac{L}{N}$ when $N$ increases, i.e. we choose $L$ such that $\frac{L}{N}$ approximates $\omega$. The rate in bits equals $R^k = \frac{\log_2 M_L}{L} = \frac{\log_2 M_L}{L}$, which goes to $\omega \cdot \log_2 c_k$ when $N$ increases. Using the Chernoff bound it follows that the error probability $\Pr\{k \cdot E \geq N - L\}$ goes to zero if and only if $kpN < N - L$ or in other words $\omega < 1 - kp$. Therefore each rate $R^k = \omega \log_2 c_k$ with $\omega < 1 - kp$ is achievable and

$$R^k = (1 - kp) \log_2 c_k. \quad (4.2)$$

If recursive coding would have been used for the binary symmetric channel, as described in Section 3.1, the maximal achievable transmission rate would also have been equal to $\bar{R}^k$. To see this, observe that the number $\log_2 c_k$ is the (maximal) precoding rate $R_p$, i.e. the amount of information in bits that is contained in one symbol of an (infinitely long) precoded sequence. Let $\tau$ be the expected number of transmissions needed to send a symbol of the precoded sequence and correct all occurring transmission errors. Since an erroneous transmission introduces $k$ extra symbols to be transmitted, we derive $\tau = 1 + kp\tau$, so $\tau = 1/(1 - kp)$. Therefore the maximal transmission rate $\bar{R}^r$ of the binary symmetric recursive coding strategy satisfies

$$\bar{R}^r = R_p/\tau = (1 - kp) \log_2 c_k. \quad (4.3)$$

By increasing the delay, or in case of variable delay by increasing the state threshold, the error probability can be made arbitrarily small as long as $kp < 1$.

We now try to generalize these results to our arbitrarily chosen discrete memoryless channel. Suppose we use block coding, as described in Section 2.2. Let $\tau_x (x \in X)$ be the expected number of transmissions needed to send an $x$-symbol of the precoded sequence and correct all occurring transmission errors. Since an $x \rightarrow y$ channel error introduces $k_{xy}$ extra $x$-symbols to be transmitted, we derive $\tau_x = 1 + \sum_y p_{xy}k_{xy}\tau_x$, so

$$\tau_x = 1/(1 - \sum_y k_{xy}p_{xy}). \quad (4.4)$$

Note that when $\sum_y k_{xy}p_{xy} \geq 1$, it will on average be impossible to correct a transmission error that occurred while sending symbol $x$, so the transmission rate is zero. Let $L_x$ be the number of $x$-symbols in the precoded sequence of length $L$ for each $x \in X$. Then the total expected number of transmissions needed to send the precoded sequence is $\sum_x L_x\tau_x$. Let the block length $N$ be equal to $\sum_x L_x\tau_x$, then the transmission rate for $M$ messages equals

$$R^c = \frac{\log M}{\sum_x L_x\tau_x}. \quad (4.5)$$

Note that $N$ should actually be chosen slightly larger than $\sum_x L_x\tau_x$ to obtain an exponential decrease of error probability for increasing block length, but this is neglected here to avoid distracting the reader.
Since $r_x$ in general depends on $x$, it follows that the transmission rate depends on the symbol precoding distribution $q$, a vector with components $q_x \ (x \in X)$, where $q_x = L_x \cdot L$. Therefore we introduce the rate function $\bar{R}(q)$ to denote the maximal achievable transmission rate as a function of the symbol precoding distribution. Consequently, each message is precoded to a sequence of length $L$ such that $L_x \approx q_x \cdot L$. We use $qL$ to denote the vector $L$ with components $L_x \approx q_x \cdot L \ (x \in X)$. Let $M(L)$ be the set of $X$-ary sequences containing $L_x$ symbols $x \ (x \in X)$, and no forbidden subsequences. Denote its cardinality by $M(L)$. Then the maximal precoding rate $R_p$ as a function of the symbol precoding distribution $q$ satisfies

$$R_p(q) = \lim_{L \to \infty} \frac{M(qL)}{L},$$

and the maximal achievable transmission rate for a given symbol precoding distribution $q$ is

$$\bar{R}(q) = \frac{R_p(q)}{\sum_x q_x r_x}.$$  \hspace{1cm} (4.6)

By properly choosing the symbol precoding distribution the transmission rate is maximized:

$$\bar{R} = \sup_{q \in Q} \bar{R}(q),$$

where $Q$ is the set $\{q \mid q_x \geq 0, x \in X, \sum_{x \in X} q_x = 1\}$ of symbol precoding distributions. Note that the symbol precoding distribution that achieves the maximum in Equation 4.8 not necessarily maximizes the precoding rate. Equations 4.7 and 4.8 hold for block coding as well as recursive coding. Note that for block coding the effect of the tail symbol distribution on the symbol (precoding) distribution vanishes for increasing transmission rate, i.e. when $N$ is chosen close to $\sum_x L_x r_x$.

It is difficult to derive a general expression for $R_p$ in terms of the repetition numbers and the symbol precoding distribution, although a fractional generating polynomial can be computed, which is shown next. The generating polynomial $p$ in variables $z_x \ (x \in X)$ is defined as $\sum_L M(L) \prod_{x \in X} z_x^{L_x}$, where the sum is taken over all vectors of length $|X|$ with non-negative integer components. We also define $M^x(L)$ for $x \in X$ as the number of $X$-ary sequences in $M(L)$ ending with symbol $x$. The corresponding generating polynomials are denoted by $p_x \ (x \in X)$. From the Definitions follows that $p = 1 + \sum_x p_x$. Let $e_x$ denote the unity vector with 1 at component $x$ and 0 elsewhere. The following recurrent relations for the numbers $M^x(L)$ are easily derived:

$$M^x(L) = \begin{cases} M(L - e_x) - \sum_{y \in X \setminus \{x\}} z_{xy} M^y(L - k_{xy} e_x), & \text{if } L_x > 0, \\ 0, & \text{if } L_x = 0. \end{cases}$$

In terms of generating polynomials Equation 4.9 is equivalent to $p_x(z) = z_x \cdot p(z) - \sum_{y \in X \setminus \{x\}} z_{xy} \cdot p_y(z) \ (x \in X)$. Therefore, we define the $|X|$ by $|X|$ matrix $D$ (not to be confused with the coding delay, it is the “$D$” from “Denominator” as will become clear later on) by $D_{xy} = z_{xy} - z_x \ (x, y \in X)$, such that the vector $p$ of polynomials is the solution of $Dp = e$. Let $D^x$ ($x \in X$) be the matrix $D$ with the $x^{th}$ column replaced by the transpose of $e_x$; then according to Cramer’s rule $p$ satisfies $p_x = \frac{\det D^x}{\det D} \ (x \in X)$. It follows that the generating polynomial $p$ is the fraction of a numerator polynomial $p^a = \det D + \sum_{x \in X} \det D^x$ and a denominator polynomial $p^d = \det D$. In the following Lemma it is proved that $p^a = \det N$ (“$N$” from “Numerator”), $N$ being the $|X|$ by $|X|$ matrix with entries $N_{xy} = z_{xy}^k \ (x, y \in X)$.
Lemma 4.1 The numerator polynomial satisfies $p^n = \det N$.

Proof: Assume w.l.o.g. $\mathcal{X} = \{0, 1, \ldots, X - 1\}$. Let $N^s$ ($0 \leq s \leq X$) be the $X$ by $X$ matrix with entries $N_{xy}^s = z_x^{ksy}$, if $x \in \mathcal{X}, 0 \leq y < s$, and $N_{xy}^s = z_x^{ksy} - z_x$, if $x \in \mathcal{X}, s \leq y < X$. We proof by induction to $s$ that

$$\det D + \sum_{0 \leq x < s} \det D^x = \det N^s. \quad (4.10)$$

The basis $s = 0$ is easy because $N^0 = D$ by definition. Suppose $\det D + \sum_{0 \leq x < s} \det D^x = \det N^s$ for some $s \in \mathcal{X}$. Since the $s$th column of $N^s$ satisfies $N_{xs}^s = z_x^{ksy} - z_x$ ($x \in \mathcal{X}$), we derive $\det N^s = \det N^{s+1} - \det \tilde{N}^s$, where $\tilde{N}^s$ is the matrix $N^s$ with the $s$th column replaced by the transpose of $z$. If we take column $s$ of $\tilde{N}^s$ and substract it from column $0$ to $s - 1$ we obtain matrix $D^s$. Since these operations do not influence the value of the determinant, $\det \tilde{N}^s = \det D^s$. Consequently, $\det D + \sum_{0 \leq x < s+1} \det D^x = \det N^{s+1}$. This completes the proof by induction. By substituting $s = X$ in Equation 4.10, it follows that $p^n = \det N$. \hfill $lacksquare$

We conclude that the generating polynomial $p$ satisfies

$$p = \frac{\det N}{\det D}. \quad (4.11)$$

For some instances of the repetition parameters, an explicit formula for $M(L)$ can be obtained from Equation 4.11, and consequently an expression for $R_p(q)$.

4.2 When is capacity achieved?

In Section 4.3 it is shown that $R_p(q)$ is continuous in $Q$. Consequently, the supremum $\bar{R}$ of $R(q)$ over all $q \in Q$ is attainable, because $Q$ is a compact set. An interesting characterization of the attainability of a rate, using the polynomial $p(z)$ of Equation 4.11 that generates the number of precoded sequences, is presented in the following Theorem that is proved in Section 4.3. This result is similar to the first Theorem of Shannon in [81, p. 384]. Although Theorem 4.1 only deals with a particular type of constrained sequences, the proof is easily generalized to arbitrary constraints so in fact Theorem 4.1 generalizes Shannon's Theorem to non-integer durations.

Theorem 4.1 Let $R \geq 0$.

1. If $R < \bar{R}$, then $p(z) = \infty$ when $z$ is such that $z_x = (|\mathcal{X}|)^{-\tau_x}R$ for each $x \in \mathcal{X}$.

2. If $p(z) = \infty$ when $z$ is such that $z_x = (|\mathcal{X}|)^{-\tau_x}R$ for each $x \in \mathcal{X}$, then there exists a $q \in Q$ such that $R(q) = R$.

From Theorem 4.1 follows that the maximal achievable rate $\bar{R} = \sup_{q \in Q} R(q)$ is a solution of $p^\Delta(z) = 0$ when $z$ is such that $z_x = (|\mathcal{X}|)^{-\tau_x}R$ for each $x \in \mathcal{X}$.

We wonder when $\bar{R}$ equals the channel capacity. Suppose recursive coding is used. Consider the receiver at moment $T$, so the received sequence $r_0, \ldots, r_{T-1}$ is known. Suppose $r_T = x$ is received. Led by the idea of repetition feedback coding as described in Section 1.4,
the probability that an \( x \) was transmitted, should not be altered by receiving \( r_T \ldots r_{T+k_{xy}} = yx^{k_{xy}} \), i.e.

\[
\Pr\{t_T = x \mid r_0 \ldots r_{T-1}, r_T = x\} = \Pr\{t_{T+k_{xy}} = x \mid r_0 \ldots r_{T-1}, r_T \ldots r_{T+k_{xy}} = yx^{k_{xy}}\}.
\]  

(4.12)

Equation 4.12 should hold for each \( x \in \mathcal{X} \) and \( y \in \mathcal{Y} \) to obtain a regular “median” path. In order to achieve channel capacity, the channel output probabilities have to be equal to \( \pi_y \) (\( y \in \mathcal{Y} \)), the unique capacity achieving output distribution. Furthermore, the channel outputs have to be independent. Since \( t_T+k_{xy} = x \) and \( rr \ldots rr+k_{xy} \) implies that \( t_T = \ldots t_{T+k_{xy}} = x \), condition 4.12 then reduces to \( \frac{P_{xy}}{\pi_x} = \frac{P_{xy}(p_{xy})^{k_{xy}}}{\pi_y^{k_{xy}}} \). It turns out that this necessary condition to achieve capacity is also sufficient.

**Theorem 4.2** Let channel probabilities \( p_{xy} (x \in \mathcal{X}, y \in \mathcal{Y}) \), and repetition numbers \( k_{xy} (x \in \mathcal{X}, y \in \mathcal{Y}) \) be given. Assume that \( k_{xy} = 1 \) for \( x \in \mathcal{X}, y \in \mathcal{Y} \setminus \mathcal{X} \). Let \( P \) be the \( |\mathcal{X}| \) by \( |\mathcal{Y}| \) matrix with entries \( P_{xy} = p_{xy} \). Let \( \pi_y (y \in \mathcal{Y}) \) be the unique capacity achieving output distribution. If the channel probabilities satisfy the Equations

\[
\frac{P_{xy}}{\pi_y} \left( \frac{P_{xxy}}{\pi_y} \right)^{k_{xy}-1} = 1
\]  

(4.13)

for each \( x \in \mathcal{X}, y \in \mathcal{Y} \), and \( \det P \neq 0 \), then there exists a symbol precoding distribution \( \tilde{q} \) such that the maximal achievable rate of the corresponding multiple-repetition strategy \( \bar{R}(\tilde{q}) \) is equal to the capacity of the discrete memoryless channel with channel probabilities \( (p_{xy}) \).

**Proof:** Remember that \( k_{xx} = 0 \) (\( x \in \mathcal{X} \)). Suppose the channel probabilities \( (p_{xy}) \) of a discrete memoryless channel with capacity \( C \), and capacity achieving output distribution \( \pi_y (y \in \mathcal{Y}) \) satisfy Equation 4.13. Let \( \pi \) be the vector with components \( \pi_x (x \in \mathcal{X}) \). Since for \( x \in \mathcal{X}, y \in \mathcal{Y} \setminus \mathcal{X} \) the repetition numbers \( k_{xy} \) are equal to 1, the corresponding channel error probabilities \( p_{xy} \) are equal to \( \pi_y \). Assign to each \( z_x (x \in \mathcal{X}) \) the value \( \frac{z_x}{p_{xx}} \). We show that \( p(z) = \infty \). The matrix \( D \) has entries \( D_{xy} = z_x^{k_{xy}} - z_x \) which equal \( z_x \left( \frac{z_x}{p_{xx}} - 1 \right) \), so \( (D \cdot \pi')_x = z_x \sum_{y \in \mathcal{X}} (p_{xy} - \pi_y) = z_x \sum_{y \in \mathcal{Y}} (p_{xy} - \pi_y) = 0 \). Therefore the denominator polynomial \( p_d = \det D \) of \( p \) is equal to 0. The matrix \( N \) has entries \( N_{xy} = z_x^{k_{xy}} \) which equal \( \frac{z_x}{p_{xx}} \), so \( \det N = \left( \prod_{x \in \mathcal{X}} p_{xx} \right)^{-1} \det P \). Since \( \det P \neq 0 \), the numerator polynomial \( p_n = \det N \) of \( p \) is unequal to 0. This proves that \( p(z) = \infty \). By Theorem 4.1 it is sufficient to show that \( z_x = (|\mathcal{X}|)^{-1} r_{xy} (x \in \mathcal{X}) \). This follows from \( -(\pi_y)^{-1} \log z_x = -\sum_{y \in \mathcal{Y}} (1 - k_{xy}) p_{xy} \log(z_x) = \sum_{y \in \mathcal{Y}} p_{xy} \log \left( \frac{z_x}{p_{xx}} \right) \), which is the conditional mutual information between the channel input and the channel output, given that the channel input is \( x \), and thus equals the channel capacity \( C \).

If \( \det P = 0 \), it is possible to omit a suitable input without affecting the capacity of the channel [88]. The elimination of inputs can be repeated until \( \det P \neq 0 \). The resulting multiple-repetition strategy will then achieve capacity.

The capacity achieving symbol precoding distribution \( \tilde{q} \) of Theorem 4.2 is unique and computable. To see this, observe that the capacity achieving channel input distribution \( \rho_x (x \in \mathcal{X}) \) is unique because \( \det P \neq 0 \). Given the capacity achieving channel output distribution, which can be computed from the channel probabilities using Arimoto-Blahut's algorithm [6, 12], the capacity achieving channel input distribution is obtained by solving \( \rho \cdot P = \pi \). However, the channel input distribution induced from the multiple-repetition
strategy with symbol precoding distribution $\bar{q}$ has components equal to $q_x \tau_x / \sum_{y \in X} q_y \tau_y$ ($x \in X$). By equalizing the induced channel input distribution to the capacity achieving channel input distribution one obtains for each $x \in X$:

$$\bar{q}_x = (\rho_x / \tau_x) / \sum_{y \in X} (\rho_y / \tau_y).$$  \hfill (4.14)

### 4.3 Proof of Theorem 4.1

The proof of Theorem 4.1 is split up in several Lemmas. The first three Lemmas are used to proof the continuity of the maximal precoding rate as a function of the symbol precoding distribution, which is then proved in Lemma 4.5. Lemmas 4.6 and 4.7 are the main tools for proving Theorem 4.1, which is done at the end of this Section.

We assume that $k_{xy} \geq 2$ for $x, y \in X$ in order to proof Lemma 4.2. Remember that all logarithms are to the base $|X|$. In Lemmas 4.3, 4.5, and 4.6 we assume w.l.o.g. that $L'$ is an integer to avoid distracting the reader.

The first Lemma shows that $M(\vec{v})$ is ascending in $\vec{v}$.

**Lemma 4.2** Let $\vec{v}$ and $\vec{w}$ be vectors of non-negative integers. If $v_x \leq w_x$ ($x \in X$), then $M(\vec{v}) \leq M(\vec{w})$.

**Proof:** W.l.o.g. is assumed that $v_x = w_x$ for $x \in X, x \neq x$, and $w_x = v_x + 1$ for some $\chi \in X$. We construct an injective function from $M(\vec{v})$ to $M(\vec{w})$. Let $m \in M(\vec{v})$, then $m$ can be written as $a^s t$ for some $a \in X$, maximally chosen $s > 0$, and $X$-ary sequence $t$. Consider the sequence $a^{s-1} \chi at$. Since if $a \neq \chi, k_{ax} > 1$ and $k_{xa} > 1$, this sequence contains no forbidden subsequences, and is an element of $M(\vec{w})$. The described function is indeed injective, so $M(\vec{v}) \leq M(\vec{w})$. \hfill \blacksquare

A way of comparing the precoding rates of two symbol precoding distributions $q$ and $q'$ is presented next. First the case where the zero components of $q'$ cover the zero components of $q$.

**Lemma 4.3** Let $q, q' \in Q$. Let $S = \{x \in X \mid q_x > 0\}$. Suppose that $q'_x = 0$ whenever $q_x = 0, x \in X$. Let $L$ be a positive integer. Let $L' = L \cdot \max_{x \in S} \frac{q'_x}{q_x}$. Then $\frac{\log M(q_L)}{L} \leq \frac{\log M(q'_L)}{L'} \cdot \max_{x \in S} \frac{q'_x}{q_x}$.

**Proof:** Since $q'_x \leq q_x L'$ for $x \in S$, and $q'_x = 0$ for $x \not\in S$, it follows from Lemma 4.2 that $M(q'_L) \leq M(qL')$. \hfill \blacksquare

The second case of comparing precoding rates is the case where $q$ has a zero component that might not be equal to zero in $q'$. Let $K$ be the maximum over all finite $k_{xy} (x, y \in X)$.

**Lemma 4.4** Let $s \in X$. Let $\vec{v}$ be a vector of non-negative integers with sum $V$. Let $\vec{w}$ be the vector $\vec{v}$, except $w_s = 0$. Then $M(\vec{v}) \leq \binom{V}{v_s} M(\vec{w})(v_s K + 1)|X| (|X| K)^{v_s}$.

**Proof:** An $X$-ary sequence of length $V$ containing exactly $v_s$ symbols $s$ has $\binom{V}{v_s}$ possible ways of positioning the $s$-symbols. Fix one of these positionings. It is sufficient to show
that the number of elements of $M(y)$ with symbol $s$ on the fixed positions is not more than $M(w)((v,K+1)|X|K)^{v_s}$.

Let $m \in M(y)$, such that $m$ has symbol $s$ on the fixed positions. Remove all symbols $s$ from $m$ and denote the resulting sequence by $m'$. Note that $m'$ can easily be constructed from $m$, since the positioning of the $s$-symbols is known. The sequence $m'$ has symbol distribution $u_s$, but might contain some forbidden subsequences.

When a forbidden subsequence occurs in $m'$, the corresponding segment of $m$ can be written as $as^\alpha b^\beta$ for some $a, b \in X$, $\alpha, \beta > 0$ such that $a \neq s \neq b$, and $\beta$ is maximally chosen. To obtain a sequence that does not contain any forbidden subsequence, we remove $b^\beta$ from $m'$ if $a = b$, and $b^{\beta-1}$ if $a \neq b$. When the same procedure is repeated for all segments of $m'$ that contain a forbidden subsequence, the sequence $m''$ is obtained that contains no forbidden subsequences. The sequence $m''$ contains at most $v_s$ forbidden subsequences, and for each forbidden subsequence in $m''$ at most $K - 1$ symbols are removed from $m'$, so $m'' \in M(w - y)$ for some vector $y$ with $\sum_{x \in X} u_x \leq (K - 1)v_s$. Since the number of possible vectors $y$ is upper bounded by $(v,K+1)|X|K)^{v_s}$, and by Lemma 4.2 $M(w - y) \leq M(w)$ for each $y$, it is sufficient to show that given $y$, an element $m'' \in M(w - y)$ can be the image of at most $(|X|K)^{v_s}$ sequences $m'$.

Let $y$ be a vector such that $\sum_{x \in X} u_x \leq (K - 1)v_s$. Let $m'' \in M(w - y)$. If $m''$ is constructed as described above from some $m'$, then there exist vectors $u_i (0 \leq i < v_s)$ such that $\sum_{0 \leq i < v_s} u_i = y$ and $u_i$ has $|X| - 1$ components equal to zero and one component in $\{0, \ldots, K - 1\}$, from which the sequence $m'$ can uniquely be constructed. A vector $u_i$ corresponds with the removed subsequence $b^\beta$ or $b^{\beta-1}$ at the position of the $i$th symbol $s$. The total number of possibilities to choose the vectors $u_i (0 \leq i < v_s)$ given $y$ is upper bounded by $(|X|K)^{v_s}$, and therefore so is the number of sequences $m'$ from which $m''$ can be constructed.

The next Lemma shows the continuity of the precoding rate as a function of the symbol precoding distribution. For this purpose the Euclidean norm $(\sqrt{q, q})$ on $Q$ is used.

**Lemma 4.5** The function $R_p(q) = \lim_{L \to \infty} \frac{\log M(q^L)}{L}$ is continuous in $Q$.

**Proof:** Let $q \in Q$. Let $(q^n)_{n \geq 0}$ be a sequence in $Q$ that converges to $q$. It is sufficient to show that the sequence $(R_p(q^n))_{n \geq 0}$ converges to $R_p(q)$.

We first show that $\lim_{n \to \infty} R_p(q^n) \leq R_p(q)$. Let $S$ be the set $\{x \in X | q_x > 0\}$. Let $n \geq 0, L > 0$. Let $L'$ be the integer $L \cdot \max_{x \in S} \frac{q_x}{q_L}$. Let $q''$ be the vector $q^n$, except $q''_x = 0$ for $x \notin S$. From Lemma 4.3 follows that $\frac{\log M(q^{nL'})}{L} \leq \frac{\log M(q^{L'})}{L'} \cdot \max_{x \in S} \frac{q_x}{q_L}$. When repeatedly combining this with Lemma 4.4 we obtain $\frac{\log M(q^nL)}{L} \leq \frac{\log M(q^{L'})}{L'} \cdot \max_{x \in S} \frac{q_x}{q_L} + (1/L) \sum_{x \in S} \log \left( \frac{L}{q_L} (q_x K + 1) |X|K \right)^{v_x} \cdot \max_{x \in S} \frac{q_x}{q_L} + \sum_{x \in S} (h(q_x) + q_x |X|K).$ Let $n \to \infty$, then $\lim_{n \to \infty} R_p(q^n) \leq R_p(q)$.

Finally we show that $\lim_{n \to \infty} R_p(q^n) \geq R_p(q)$. Let $S_n$ be the set $\{x \in X | q^n_x > 0\}$. Choose $n$ sufficiently large such that $q^n_x > 0$ for all $x \in S$. Let $L > 0$. Let $L'$ be the integer $L \cdot \max_{x \in S} \frac{q_x}{q^n_L}$. From Lemma 4.3 follows that $\frac{\log M(q^{nL'})}{L} \leq \frac{\log M(q^{L'})}{L'} \cdot \max_{x \in S} \frac{q_x}{q^n_L}$. Let $L \to \infty$, then $R_p(q) \leq R_p(q^n) \cdot \max_{x \in S} \frac{q_x}{q^n_L}$. Let $n \to \infty$, then $R_p(q) \leq \lim_{n \to \infty} R_p(q^n)$. 

We proceed towards the proof of Theorem 4.1 with two Lemmas, for which some Definitions have to be made. Let $0 < z_x < 1 (x \in X)$. Let $L$ be a positive integer. Define the
function $F^L(q)$ on $Q$ by

$$F^L(q) = - \frac{\log M(qL)}{L} - \sum_{x \in X} q_x \log z_x.$$  \hfill (4.15)

Let $c > 0$. Let $Q^L_c$ be the set $\{q \in Q \mid q_x L \text{ is an integer}, q_x \geq c \ (x \in X)\}$. Define the function $F(q)$ on $Q$ as $\lim_{L \to \infty} F^L(q)$.

**Lemma 4.6** Let $c > 0$. If $F(q) > 0$ for all $q \in Q$, then $\min_{q \in Q^L_c} F^L(q) \geq \frac{1}{2} \inf_{q \in Q} F(q) > 0$, for sufficiently large $L$.

**Proof:** Suppose that $F(q) > 0$ for all $q \in Q$. From Lemma 4.5 follows that $F(q)$ is a continuous function on $Q$. Since $Q$ is a compact set, $\inf_{q \in Q} F(q) = F(\hat{q}) > 0$, for some $\hat{q} \in Q$. For each $q \in Q$ there is an integer, say $L_q$, such that $F^L(q) \geq \frac{3}{2} F(q)$ when $L \geq L_q$. Choose integer $l$ sufficiently large such that $-\sum_{x \in X} \log z_x + \frac{1}{c} \leq \frac{1}{2} F(\hat{q})$. Since $Q^L_c$ is a finite set, we define $l' = \max_{q \in Q^L_c} L_q$. We show that $\min_{q \in Q^L_c} F^L(q) \geq \frac{1}{2} F(\hat{q})$ for $L \geq \max\{l, l'\}$.

Let $L \geq \max\{l, l'\}$. Let $q \in Q^L_c$. Since $L \geq l$ there exists a $q' \in Q^L_c$ such that $|q_x - q'_x| < \frac{1}{L}$ for $x \in X$. Choose such a $q' \in Q^L_c$. Let $L = L \cdot \max_{x \in X} q'_x$, then $L \geq L'$, and

$$\begin{align*}
F^L(q) - F^L(q') &= \sum_{x \in X} (q'_x - q_x) \log z_x + \frac{\log M(q_x L)}{L} - \frac{\log M(q'_x L)}{L} \\
&\geq \frac{1}{2} \sum_{x \in X} \log z_x + \frac{\log M(q_x L)}{L} \geq \frac{1}{2} \sum_{x \in X} \log z_x + \frac{\log M(q'_x L)}{L} \geq \frac{1}{2} \sum_{x \in X} \log z_x - \frac{1}{c} \geq -\frac{1}{4} F(\hat{q}).
\end{align*}$$

Therefore $F^L(q) \geq F^L(q') - \frac{1}{4} F(q') \geq -\frac{3}{4} F(q') - \frac{1}{4} F(\hat{q}) \geq \frac{1}{2} F(\hat{q}).$

**Lemma 4.7** If $F(q) > 0$ for all $q \in Q$, then $p(\hat{z}) < \infty$.

**Proof:** Suppose that $F(q) > 0$ for all $q \in Q$. Choose $c > 0$ such that $-c \sum_{x \in X} \log z_x \leq \frac{1}{4} F(\hat{q})$ (as in Lemma 4.6). Define, for any subset $S$ of $X$, the set $V^S_c$ as the set of vectors $v$ such that $v_x \leq c \sum_{x \in X} v_x$, if and only if $s \in S$. Then $p(\hat{z}) = \sum_{S \subseteq X} \sum_{v \in V^S_c} M(v) \prod_{x \in X} z_x^{v_x}$. Since the number of subsets of $X$ is finite, it is sufficient to show that $\sum_{v \in V^S_c} M(v) \prod_{x \in X} z_x^{v_x} < \infty$ for each subset $S$ of $X$.

Let $S$ be a subset of $X$. Let $v \in V^S_c$. Let $V = \sum_{x \in X} v_x$. Let $v' = v_x$, if $x \notin S$, and $v'_x = v_x + c$, if $x \in S$. Let $V' = \sum_{x \in V} v'_x$, where $V' = V(1 + c|S|)$, and $v'_x \geq c' = \frac{c}{1 + c|S|}$ for $x \in X$. Furthermore $M(v') \prod_{x \in X} z_x^{v'_x} \leq M(v') \prod_{x \in X} z_x^{v_x}$, since $c' \leq c$, the second term $-c \sum_{x \in X} \log z_x \leq \frac{1}{4} F(\hat{q})$. It is sufficient to show that $\sum_{v' \in V'} M(v') \prod_{x \in X} z_x^{v'_x} \prod_{x \in X} z_x^{v_x} < \infty$.

For $v' \in V^S_c$ we derive

$$\log M(q'_x) \prod_{x \in X} z_x^{v'_x} = \log M(q'_x L) z_x^{v'_x} = \log M(q'_x L) z_x^{v_x}.$$  \hfill (4.16)

From Lemma 4.6 follows that the first term $-F^W(v'/v') \leq -\frac{1}{2} F(\hat{q})$ for $V' \geq W$, where $W$ is a constant independent of $v'$. Since $c' \leq c$, the second term $-c \sum_{x \in X} \log z_x \leq \frac{1}{2} F(\hat{q})$.

Furthermore, $V^S_c = \mathcal{U}_{v' > 0} \{q^{v'} \mid q \in Q^{v'}\}$, and $|Q^{v'}| \leq V^{|X|}$, so $\sum_{v' \in V^S_c} M(v') \prod_{x \in X} z_x^{v_x} \prod_{x \in X} z_x^{v_x} \prod_{x \in X} z_x^{v_x} \prod_{x \in X} z_x^{v_x} < \sigma + \sum_{v' \geq W} V^{|X|} \cdot (|X|)^{-W} F(\hat{q})$ for some constant $\sigma$, so $p(\hat{z})$ converges.

**Proof of Theorem 4.1:** Let $R > 0$. Let $z_x = I^{-\tau_x} R (x \in X)$.

1. Suppose $R < \sup_{q \in Q} R(q)$. Then there exists a $\delta > 0$ and a $q \in Q$ such that $\hat{R}(q) = R + \delta$. Then $R_p(q) + \sum_{x \in X} q_x \log z_x = \delta \sum_{x \in X} q_x \tau_x > \delta$, because $\tau_x > 1$ (x \in X). Thus $p(\hat{z}) \geq \sum_{L \geq 0} M(qL) \sum_{x \in X} z_x^{q_L} \geq \sigma + \sum_{L \geq 0} (|X|)^{L\delta}$ for some constant $\sigma$ and sufficiently large $L$. Therefore, $p(\hat{z})$ diverges.
2. Suppose \( p(z) = \infty \). From Lemma 4.7 follows that there exists a \( q \in Q \) such that
\( F(q) \leq 0 \). It follows that \( R_p(q) + \sum_{z \in \mathcal{X}} q_z \log z_2 \geq 0 \), so \( \sum_{z \in \mathcal{X}} R_p(q)_z \geq R \). Since \( R(q) \) is continuous in \( Q \), \( R \) is attainable. 

4.4 Examples

We illustrate our results for a couple of channels. When a channel (error) probability \( p_{xy} = 0 \) for some \( x \) and \( y \), we allow subsequences \( y x^k \) for any finite \( k \) in the precoded sequence. Or in other words, \( k_{xy} = \infty \). For our results to remain valid, we then define \( z_{xy}^k \) and \( k_{xy} p_{xy} \) as \( 0 \), and leave out Equation \( \sum \frac{p_{xy}}{\pi_y} \left( \frac{\pi_x}{\pi_x} \right)^{k_{xy}-1} = 1 \) from Equation 4.13.

4.4.1 Binary symmetric channel

Suppose a binary symmetric repetition feedback strategy with repetition parameter \( k \geq 3 \) is used for a binary symmetric channel with cross-over probability \( p \) and \( \mathcal{X} = \mathcal{Y} = \{0, 1\} \). From Equation 4.11 follows that the generating polynomial equals \( p(z_0, z_1) = (1 - (z_0 z_1)^k)/(1 - z_0 - z_1 + z_0 z_1^k + z_1 z_0^k - (z_0 z_1)^k) \). Since \( \tau = \tau_0 = \tau_1 = 1/(1 - kp) \), the maximal achievable rate equals, by Theorem 4.1, \( (1/\tau) \log_2 (1/z_0) \) where \( z_0 \) is the solution \( 0 < z < 1 \) of \( p^d(z, z) = 0 \). Since \( p^d(z, z) = (z - 1)^2 - (z^k - z)^2 \), it follows that \( z = z_0 \) satisfies \( z^k - 2z + 1 = 0 \), so

\[
R = (1 - kp) \log_2 c_k, \tag{4.16}
\]
where \( c_k \) is the solution \( c > 1 \) of \( c^k - 2c^{k-1} + 1 = 0 \). This agrees with Equations 4.2 and 4.3 from Section 4.1.

The capacity of the binary symmetric channel is equal to \( 1 - h(p) \), where \( h \) is the binary entropy function, which is achieved with a uniform output distribution. From Equation 4.13 follows that capacity is achieved when \( p \) equals \( p_k \), the solution \( 0 < p < \frac{1}{2} \) of

\[
2p(2 - 2p)^{k-1} = 1. \tag{4.17}
\]
Note that by substituting \( p = 1/2 \) in Equation 4.17, it easily follows that \( p_k = 1 - \frac{1}{2} c_k \). For \( k = 3, 4, 5 \), the rate \( R \) is depicted in Figure 4.1.

4.4.2 Z-channel

The Z-channel is a binary channel \( (\mathcal{X} = \mathcal{Y} = \{0, 1\}) \) with \( p_{01} = 0 \), as depicted in Figure 1.5. Its capacity is equal to \( \log_2 (1 + 2^{-k p_0}) \), which is achieved with output distribution \( \pi_0 = 2^{-k p_0}/(1 + 2^{-k p_0}) \), and \( \pi_1 = 1/(1 + 2^{-k p_0}) \), where \( h \) is the binary entropy function. A multiple-repetition strategy is used with \( k_0 = \infty \), and \( k_1 = k \). From Equation 4.11 follows that the generating polynomial \( p(z_0, z_1) \) is equal to \( 1/(1 - z_0 - z_1 + z_0 z_1^k) \), and from Equation 4.4 follows that \( \tau_0 = 1 \) and \( \tau_1 = 1/(1 - pk) \). From the generating polynomial is derived that

\[
M(L_0, L_1) = \sum_{0 \leq i \leq L_0, 0 \leq L_1/k} \binom{L_0}{i} (-1)^i \binom{L_0 + L_1 - ki}{L_0}. \tag{4.18}
\]
The maximal attainable rate $R$ given $k$ and $p$ is, according to Theorem 4.1, the solution $R > 0$ of the Equation

$$2^{-R} + 2^{-\frac{R}{1-p^k}} - 2^{-R\frac{1-p^k}{1-p^k}} = 1.$$  \hspace{1cm} (4.19)

For $k = 2, 3, 4, 5$, the rate $R$ as a function of $p$ is depicted in Figure 4.2. For each $k$ there is exactly one $p = p_k$ (the solution of Equation 4.13 for which capacity is attained. As can be seen from Figure 4.2, even when $p \neq p_k$ for some $k$, there are multiple-repetition strategies that nearly attain capacity. The density of relatively high rate strategies increases with decreasing $p$.

Another interesting property of this particular multiple-repetition strategy is that decoding can be simply done from left to right, because when the subsequence $01^k$ occurs in the received sequence, the decoder knows that the received 0 is the result of a 1 $\rightarrow$ 0 error and can immediately replace the subsequence by 1.

### 4.4.3 A “super” channel

The final example channel is depicted in Figure 4.3. The input and output alphabets equal $\{0, 1, 2, 3\}$. This “super” channel with parameter $p$ ($0 \leq p \leq 1$) follows from research on coding for memories with known defects [79], which is described in subsection 7.3.4 of this thesis. In fact, this “super” channel equals $BDC^2$ restricted to $2^2$ optimal inputs [77]. It turns out that capacity achieving strategies for this channel can be used to achieve essentially error free storage of information in memories with known defects. The capacity of this channel is equal to $1 - H((1-\frac{1}{2})^2, \frac{1}{2}p^2, \frac{1}{2}p^2, p(1-\frac{3}{2}p))$, which is achieved by a uniform input (and output) distribution, where $H$ is the quaternary entropy function.

A multiple-repetition strategy is used with repetition numbers $k_{xy} = 5$, if $p_{xy} = \frac{1}{4}p^2$ (the dashed lines in Figure 4.3), and $k_{xy} = 2$, if $p_{xy} = p(1-\frac{3}{2}p)$ (the dotted lines in Figure 4.3), for $x \in X, y \in Y, x \neq y$. From Equation 4.4 it follows that $\tau_x = 1/(1-2p-p^2)$ ($x \in X$). For reasons of symmetry it is sufficient to compute generating polynomial $p$ with

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig4.1}
\caption{Binary symmetric channel performance}
\end{figure}
Figure 4.2: Z-channel performance

Figure 4.3: A "super" channel
Figure 4.4: "super" channel performance

\[ z_x = z \ (x \in \mathcal{X}) \]. From Equation 4.11 it follows that generating polynomial \( p(z, z, z, z) = \frac{((z + 1)(2z^4 - 2z^3 + 2z^2 - z + 1))}{((z - 1)(2z^4 + 2z^3 + 2z^2 + 3z - 1))}. \) Let \( z = c \) be the unique solution \( 0 < z < 1 \) of \( 2z^4 + 2z^3 + 2z^2 + 3z - 1 = 0 \) \( (c \approx 0.27) \). Then it follows from Theorem 4.1 that the maximal attainable rate \( \bar{R} \) given \( p \) satisfies

\[ \bar{R} = -(1 - 2p - p^2) \log_4 (c). \]  \( (4.20) \)

Note that \( \bar{R} = 0 \) when \( \sqrt{2} - 1 < p \leq 1 \). The maximal rate is attained for the uniform symbol precoding distribution. The capacity of the "super" channel, and the rate \( \bar{R} \) of the proposed multiple-repetition strategy, both as a function of parameter \( p \), are depicted in Figure 4.4. For \( p \approx 0.07 \), the rate approaches capacity, but is not equal to it. That \( R \neq C \) at this point, can be seen when Equation 4.13 is rewritten as

\[ p(2 - p)^4 = 1, \]  \( (4.21) \)
\[ p(4 - 3p)(2 - p)^2 = 1, \]  \( (4.22) \)

which have \( p = 1 \) as only solution.

4.5 Conclusions

The transmission rate of multiple-repetition feedback strategies was computed for block coding schemes with finite block length and recursive coding schemes with finite delay. This led to the fixation of the symbol precoding distribution. An expression for the maximal achievable rate was obtained which held for block coding as well as for recursive coding. A fractional generating polynomial for the number of precoded sequences was derived and used in Theorem 4.1 to characterize the attainability of a rate. In Theorem 4.2 it was shown for which discrete memoryless channel a given multiple-repetition feedback strategy can obtain the channel capacity. The results were illustrated by some example channels.
Chapter 5

Choosing the parameters

Given a discrete memoryless channel with feedback, how should the repetition numbers be chosen to maximize the transmission rate? And how should the symbol precoding distribution be chosen? The answers are given in this Chapter. It is also indicated how closely the channel capacity can be approached in general. The simplest case, namely the binary symmetric channel, is considered first. The results obtained for the binary symmetric case are then generalized to the class of so-called uniform channels, and finally some results for arbitrary channels are derived.

5.1 Binary symmetric channel

Consider the binary symmetric channel with crossover-probability $p$ and a noiseless delayless feedback link. Suppose a binary symmetric repetition feedback strategy is used with repetition parameter $k$, $k \geq 3$. As was shown in subsection 4.4.1, independent of the type of coding, block coding or recursive coding, the maximal achievable transmission rate satisfies

$$R = (1 - kp) \log_2 c_k,$$

where $c_k$ is the solution $c > 1$ of $c^k - 2c^{k-1} + 1 = 0$. The rate $R$ equals the channel capacity when $p = p_k = 1 - \frac{1}{2}c_k$ (see subsection 4.4.1). In table 5.1 the values of $p_k$ and $c_k$ (together with the maximal achievable transmission rate) are depicted for $k = 3, 4, \ldots, 9$. When the crossover-probability equals $p_k$ for some $k \geq 3$, it is obvious that this $k$ should be the repetition parameter in order to achieve a maximal transmission rate. The following Lemma shows how $k$ is easily determined from $p$ in case $p = p_k$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$p_k$</th>
<th>$c_k$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.1910</td>
<td>1.618</td>
<td>0.2964</td>
</tr>
<tr>
<td>4</td>
<td>0.08036</td>
<td>1.839</td>
<td>0.5965</td>
</tr>
<tr>
<td>5</td>
<td>0.03621</td>
<td>1.928</td>
<td>0.7755</td>
</tr>
<tr>
<td>6</td>
<td>0.01703</td>
<td>1.966</td>
<td>0.8756</td>
</tr>
<tr>
<td>7</td>
<td>0.008209</td>
<td>1.984</td>
<td>0.9314</td>
</tr>
<tr>
<td>8</td>
<td>0.004018</td>
<td>1.992</td>
<td>0.9622</td>
</tr>
<tr>
<td>9</td>
<td>0.001984</td>
<td>1.996</td>
<td>0.9793</td>
</tr>
</tbody>
</table>

Table 5.1: Capacity achieving crossover-probabilities
Lemma 5.1 Let $k \geq 4$. The number $p_k$ satisfies $k - \frac{1}{2} < -\log_2 p_k < k$.

Proof: Define the function $f_k(p)$ for $0 < p < \frac{1}{2}$ as $2p(2 - 2p)^{k-1} - 1$. The number $p_k$ is the solution of $f_k(p) = 0$. Since $f_k(2^{-k}) = (1 - 2^{-k})^{k-1} - 1 < 0$ and $f_k(2^{1-k}) = 2(1 - 2^{1-k})^{k-1} - 1 < 0$, it follows that $2^{-k} < p_k < 2^{1-k}$. Furthermore, $k + \log_2 p_k = (1 - k) \log_2(1 - p_k) < (1 - k) \log_2(1 - 2^{1-k}) < \frac{k-1}{\ln 2} \cdot 2^{2-k}$, where the last inequality follows from $-\ln(1 - x) = \sum_{n \geq 1} \frac{1}{n} x^n < x + x^2$ for $0 < x < \frac{1}{2}$. Since $\frac{k-1}{\ln 2} \cdot 2^{2-k} < \frac{1}{2}$ for $k \geq 6$, the proof is completed by verifying the cases $k = 4$ and $k = 5$: $f_4(2^{-4(4-\frac{1}{2})}) \approx 0.07$ and $f_5(2^{-5(5-\frac{1}{2})}) \approx 0.18$, which are both positive, and therefore $k - \frac{1}{2} < -\log_2 p_k$ for $k = 4, 5$.

From Lemma 5.1 it follows that for arbitrary $p$ the optimal choice for $k$ is either $\lfloor -\log_2 p \rfloor$ or $\lceil -\log_2 p \rceil$. A reasonable choice for $k$ would be the closest integer to $-\log_2 p$, so

$$k = \lfloor -\log_2 p \rfloor. \quad (5.2)$$

The following Theorem indicates how close the transmission rate can approximate the channel capacity when $k = \lfloor -\log_2 p \rfloor$ is chosen.

Theorem 5.1 Let $k(p)$ denote the closest integer to $-\log_2 p$. Let $\tilde{R}(p)$ be the transmission rate $(1 - k(p)p)\log_2 c_k(p)$, and $C(p)$ be the channel capacity $1 - h(p)$. Then

$$C(p) - \tilde{R}(p) = O(p) \quad (p \to 0). \quad (5.3)$$

Proof: Since $c_k = 2(1 - p_k) = 2(1 - \mathcal{O}(2^{-k}))$ for $k \to \infty$, and $k(p)p = h(p) + \mathcal{O}(p)$ for $p \to 0$, the result follows easily.  

When $k$ is fixed, the exact difference between $C$ and $\tilde{R}$ can be written as an informational divergence, as is shown in the following Theorem.

Theorem 5.2 Let $k \geq 3$. Then

$$C - \tilde{R} = D(p\| p_k) \quad (5.4)$$

Proof: We derive $C - \tilde{R} = 1 - h(p) - (1 - kp)\log_2 c_k = p\log_2 p + (1 - p)\log_2(1 - p) + p\log_2(2c_k^{k-1}) - (1 - p)\log_2(c_k/2)$. Since $p_k = 1 - c_k/2$ and $c_k^{k-1}(c_k - 2) + 1 = 0$, it follows that $1/p_k = 2c_k^{k-1}$, so $C - \tilde{R} = D(p\| p_k)$.  

The result of Theorem 5.2 also follows from the graphical interpretation of the informational divergence, since $\tilde{R}$ is a straight line that is tangent to the capacity curve at $p = p_k$, as is depicted in Figure 4.1. Theorem 5.2 shows that the order of approximation in Equation 5.3 is tight, because $\lim_{p \to 0} D(p\| \gamma p) / p = (\gamma - 1) / \ln 2 - \log_2 \gamma$ when $\gamma > 0$. Moreover, a general upper bound is easily derived.

Corollary 5.1 Let $0 \leq p \leq 1/3$. Then $C(p) - \tilde{R}(p) \leq p$.

Proof: The repetition parameter $k$ is chosen such that $k = \lfloor -\log_2 p \rfloor$. Since $k = \lfloor -\log_2 p \rfloor$, it follows that $p/2 \leq p_k \leq 2p$. According to Equation 5.4, the worst cases are $p_k = p/2$ and $p_k = 2p$. First, suppose $p_k = p/2$. Then $D(p\| p_k) = p + (1 - p)\log_2 \frac{1-p}{1-p/2} \leq p$. Secondly, suppose $p_k = 2p$. Then $D(p\| p_k) = -p + (1 - p)\log_2 \frac{1-p}{1-2p} = -p - \frac{1}{\ln 2} \ln(1 - p/(1 - p))$. Since $-\ln(1 - x) \leq x + x^2$ for $0 \leq x \leq 1/2$, it follows that $D(p\| p_k) \leq p(1/\ln 2 - 1) + \frac{p^2}{(1-p)^2}$. Since $p_k < 0.2$, $p < 0.1$, so $D(p\| p_k) \leq p(1/\ln 2 - 1 + \frac{1}{2}\ln 2) \leq p$. Consequently $C(p) - \tilde{R}(p) \leq p$.  


5.2 Uniform channels

In this Section we generalize the results of the former Section to a wider class of channels, the so-called uniform channels. Assume w.l.o.g. \( \mathcal{X} \subseteq \mathcal{Y} \). Note that as usual all logarithms are to the base \(|\mathcal{X}|\).

**Definition 5.1** A discrete memoryless channel is called a uniform channel if the channel considered as a directed graph with labeled edges has all of the following properties:

1. All input nodes \( x, x \in \mathcal{X} \), have the same bag of outgoing edge labels.
2. All output nodes \( y, y \in \mathcal{Y} \), have the same bag of incoming edge labels.
3. The labels of the edges that come in at the output nodes \( y, y \in \mathcal{Y} \setminus \mathcal{X} \), are such that \( \sum_{y \in \mathcal{Y} \setminus \mathcal{X}} p_{xy} \log \sum_{x \in \mathcal{X}} p_{xy} \) is independent of \( s, s \in \mathcal{X} \).
4. All edges from input node \( x \) to output node \( x, x \in \mathcal{X} \), have the same label.

Note that a bag is a multi-set where elements can occur more than once. The fourth property is simply a renaming of the output symbols (in the back of our mind we think of \( p_{x\mathcal{X}} \) as the largest of all \( p_{xy}, y \in \mathcal{X} \)). A uniform channel has the property that capacity is achieved by a uniform input distribution, because then \( I(X = x; Y) \) is independent of \( x \) for each \( x \in \mathcal{X} \).

The class of uniform channels as defined in Definition 5.1 generalizes the class of wide-sense symmetric channels as defined by Becker [7, pp.6-8] (see also subsection 1.4.3). Uniform channels are similar to symmetric channels as defined by Gallager [35, p.94]. By partitioning the set of outputs into \( \mathcal{X} \) and \( \mathcal{Y} \setminus \mathcal{X} \), it follows that a uniform channel is approximately a symmetric channel in Gallager’s sense. A uniform channel is slightly more general than a symmetric channel with output partitioning \( \{\mathcal{X}, \mathcal{Y} \setminus \mathcal{X}\} \), because property 3 of Definition 5.1 is slightly more general than requiring that all output nodes \( y, y \in \mathcal{Y} \setminus \mathcal{X} \), have the same bag of incoming edge labels. Our restriction to output partitioning \( \{\mathcal{X}, \mathcal{Y} \setminus \mathcal{X}\} \) is related to the restriction to hard decision as mentioned in Section 8.2 of this thesis.

Fix some \( \chi \in \mathcal{X} \). Let \( p_{y} = p_{x\chi}, y \in \mathcal{Y} \), then each channel probability equals some \( p_{y} \). Furthermore, each edge that comes in at one of the output nodes \( y \in \mathcal{X} \) has a label that equals some \( p_{y} \) where \( y \in \mathcal{X} \), and each edge that comes in at one of the output nodes \( y \in \mathcal{Y} \setminus \mathcal{X} \) has a label that equals some \( p_{y} \) such that \( y \in \mathcal{Y} \setminus \mathcal{X} \). As an example the uniform channel with input alphabet \( \mathcal{X} = \{0, 1, 2\} \) and output alphabet \( \mathcal{Y} = \{0, 1, 2, 3\} \) is depicted in Figure 5.1. The dashed edges have label \( p_{0}(\chi = 0) \).

Suppose a multiple-repetition feedback strategy is used for such a channel such that each channel probability \( p_{y} \) corresponds to a repetition number \( k_{y} \ (y \in \mathcal{Y}) \). Note that \( k_{y} = 1 \) for \( y \in \mathcal{Y} \setminus \mathcal{X}, k_{\chi} = 0 \), and \( k_{y} \geq 2 \) for \( y \in \mathcal{X}, y \neq \chi \). For reasons of symmetry, the symbol precoding distribution is not fixed, i.e. all precoded sequences without the forbidden subsequences are allowed. According to Equation 4.4 \( \tau_{x} = 1/(1 - \sum_{y \in \mathcal{Y}} k_{y} p_{y}) \) for \( x \in \mathcal{X} \). From Theorem 4.1 follows that

\[
\bar{R} = (1 - \sum_{y \in \mathcal{Y}} k_{y} p_{y}) \log c, \tag{5.5}
\]

where \( 1 < c < |\mathcal{X}| \) is the solution of \( \sum_{x \in \mathcal{X}} c^{-k_{x}} = |\mathcal{X}| \). Note that for \( z_{x} = 1/c (x \in \mathcal{X}) \) indeed \( p^{\mathcal{X}}(z) = 0 \) because the rows of matrix \( D \) sum to the zero-vector; property 1 of Definition 5.1 makes this solution unique. An estimate for \( c \) follows from the next Lemma.
Lemma 5.2 Let \( k \geq 2 \). Let \( \gamma, 1 < \gamma < |\mathcal{X}| \) be the solution of \( \gamma + (|\mathcal{X}| - 1)\gamma^{1-k} = |\mathcal{X}| \), then
\[
|\mathcal{X}| - (|\mathcal{X}|)^{2-k} \leq \gamma < |\mathcal{X}| - (|\mathcal{X}|)^{1-k}.
\]

Proof: Let \( f \) be the function \( f(\gamma) = \gamma + (|\mathcal{X}| - 1)\gamma^{1-k} - |\mathcal{X}| \) for \( 1 \leq \gamma \leq |\mathcal{X}| \). We derive \( f(|\mathcal{X}| - (|\mathcal{X}|)^{1-k}) = -(|\mathcal{X}|)^{1-k} + (|\mathcal{X}| - 1)(|\mathcal{X}| - (|\mathcal{X}|)^{1-k})^{1-k} \). Since \( (1 - (|\mathcal{X}|)^{-k})^{1-k} > 1 \), it follows that \( f(|\mathcal{X}| - (|\mathcal{X}|)^{1-k}) > 0 \). On the other hand \( f(|\mathcal{X}| - (|\mathcal{X}|)^{2-k}) = -(|\mathcal{X}|)^{2-k} + (|\mathcal{X}| - 1)(|\mathcal{X}| - (|\mathcal{X}|)^{2-k})^{1-k} \). Since \( (1 - (|\mathcal{X}|)^{2-k})^{1-k} \leq |\mathcal{X}| / (|\mathcal{X}| - 1) \), it follows that \( f(|\mathcal{X}| - (|\mathcal{X}|)^{2-k}) \leq 0 \). Therefore the solution of \( f(\gamma) = 0 \) satisfies \( |\mathcal{X}| - (|\mathcal{X}|)^{2-k} \leq \gamma < |\mathcal{X}| - (|\mathcal{X}|)^{1-k} \).

Let \( k^+ \) and \( k^- \) be the maximum respectively minimum over all \( k_x, x \in \mathcal{X}, x \neq \chi \). Since \( c \) is the solution of \( c + \sum_{x \in \mathcal{X}, x \neq \chi} c^{1-k_x} = |\mathcal{X}| \), it follows from Lemma 5.2 that
\[
|\mathcal{X}| - (|\mathcal{X}|)^{2-k^-} \leq c < |\mathcal{X}| - (|\mathcal{X}|)^{1-k^+}.
\]

The capacity achieving channel output distribution \( \pi \) satisfies \( \pi_y = \frac{1}{|\mathcal{Y}|} \sum_{x \in \mathcal{X}} p_{xy} \) (\( y \in \mathcal{Y} \)). Note that \( \pi_y = \frac{1}{|\mathcal{Y}|} \sum_{x \in \mathcal{X}} p_x \) for each \( y \in \mathcal{X} \). The capacity is given by
\[
C = \sum_{y \in \mathcal{Y}} \log(p_y / \pi_y).
\]

From Theorem 4.2 follows that \( \bar{R} = C \) when
\[
|\mathcal{X}| p_y = c^{1-k_y} \sum_{x \in \mathcal{X}} p_x, \quad \text{if} \quad y \in \mathcal{X},
\]
\[
p_{xy} = p_y, \quad \text{if} \quad y \in \mathcal{Y} \setminus \mathcal{X} \quad \text{and} \quad x \in \mathcal{X}.
\]

The following Theorem generalizes Theorem 5.2 to uniform channels. It shows that the difference between \( C \) and \( \bar{R} \) can be written as an \( \mathcal{X} \)-ary informational divergence.

**Theorem 5.3** Let \( \tilde{p}_y, y \in \mathcal{Y} \) be the solution of Equation 5.8 such that \( \tilde{p}_y = \frac{1}{|\mathcal{Y}|} \sum_{x \in \mathcal{X}} p_{xy} \) for each \( y \in \mathcal{Y} \setminus \mathcal{X} \). Let \( p \) and \( \tilde{p} \) be the vectors with components \( p_y \) (\( y \in \mathcal{Y} \)) and \( \tilde{p}_y \) (\( y \in \mathcal{Y} \)) respectively. Then
\[
C - \bar{R} = D_X(p || \tilde{p}).
\]
Proof: Since \( \tilde{p}_y = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} p_{xy} \) for \( y \in \mathcal{Y} \setminus \mathcal{X} \), it follows that \( \sum_{x \in \mathcal{X}} p_x = \sum_{x \in \mathcal{X}} \tilde{p}_x \), so \( \tilde{p}_y = c^{-k_y} \pi_y \) for \( y \in \mathcal{Y} \). Therefore \( C - \tilde{R} = \sum_{y \in \mathcal{Y}} p_y \log(p_y/\pi_y) - (1 - \sum_{y \in \mathcal{Y}} k_y p_y) \log c = \sum_{y \in \mathcal{Y}} p_y \{ \log(p_y/\pi_y) - (1 - k_y) \log c \} = D_X(p \| \tilde{p}) \).

In the proof of Theorem 5.3 was shown that \( \tilde{p}_y = c^{-k_y} \pi_y \) for \( y \in \mathcal{Y} \), from which follows that \( k_y = 1 - (1 + \log(\tilde{p}_y/\sum_{x \in \mathcal{X}} \tilde{p}_y))/\log c \) for \( y \in \mathcal{X} \). This indicates how the repetition parameters of an arbitrary uniform channel should be chosen. For larger repetition parameters, the number \( c \) approximates \( |\mathcal{X}| \), see e.g. Equation 5.6 or table 5.1 for the binary symmetric case. So a suitable choice for the repetition parameters \( k_y (y \in \mathcal{X}) \), given the channel probabilities \( p_x (x \in \mathcal{X}) \), would be

\[
k_y = -\log(p_y/\sum_{x \in \mathcal{X}} p_x). \tag{5.10}
\]

The next Theorem demonstrates that this choice is indeed suitable.

**Theorem 5.4** Let \( p \) be the vector with components \( p_y (y \in \mathcal{Y}) \). Let \( k_y(p) (y \in \mathcal{X}) \) be chosen according to Equation 5.10. Let \( k_x(p) = 0 \) and \( k_{\mathcal{X}}(p) = 1 \) \( (y \in \mathcal{Y} \setminus \mathcal{X}) \). Let \( R(p) = (1 - \sum_{y \in \mathcal{Y}} k_y(p) p_y) \log c(p) \), where \( c(p) = c, 1 < c < |\mathcal{X}| \), the solution of \( \sum_{x \in \mathcal{X}} c^{-k_x(p)} = |\mathcal{X}| \). Let \( C(p) \) be the channel capacity of a uniform channel with \( p_{xy} = p_y (y \in \mathcal{Y}) \) for some \( x \in \mathcal{X} \). Then

\[
C(p) - R(p) = \mathcal{O}(\sum_{y \in \mathcal{Y}, y \neq x} p_y), (\sum_{y \in \mathcal{Y}, y \neq x} p_y \rightarrow 0). \tag{5.11}
\]

Proof: We use Theorem 5.3 and note that \( \tilde{p}_y = c^{-k_y} \pi_y \) for \( y \in \mathcal{Y} \). We have \( \tilde{p}_x = c \pi_x = c/|\mathcal{X}| \sum_{y \in \mathcal{X}} p_y \geq c/|\mathcal{X}| p_x \), so \( \log(p_x/\tilde{p}_x) \leq 1 - \log c \). Let \( y \in \mathcal{X}, y \neq x \), then \( \log \tilde{p}_y = (1 - k_y) \log c + \log \pi_y \). Since \( \pi_y = (1/|\mathcal{X}|) \sum_{x \in \mathcal{X}} p_x \) and \( k_y \) is chosen according to Equation 5.10, we conclude \( \log \tilde{p}_y \geq (1 - k_y) \log c - 1 + \log p_y + k_y \), so \( \log(p_y/\tilde{p}_y) \leq 2 - \log c + (\log c - 1) k_y < 2 \). Let \( y \in \mathcal{Y} \setminus \mathcal{X} \), then \( \tilde{p}_y = \pi_y = (1/|\mathcal{X}|) \sum_{x \in \mathcal{X}} p_{xy} \geq p_y/|\mathcal{X}| \), so \( \log(p_y/\tilde{p}_y) \leq 1 \). Summarizing, we obtain \( C - \tilde{R} = D_X(p \| \tilde{p}) \leq 1 - \log c + 2 \sum_{y \in \mathcal{Y}, y \neq x} p_y \).

From Equation 5.6 follows that \( 1 - \log c \leq -\log(1 - (|\mathcal{X}|) \cdot k^-) \), where \( k^- \) is the minimum over all \( k_y, y \in \mathcal{X}, y \neq x \). \( k^- \) is the maximum over all \( p_y, y \in \mathcal{X}, y \neq x \), then follows from Equation 5.10 that \( k^- \geq -\log(p^+ / \sum_{x \in \mathcal{X}} p_x) - 1 \), so \( -\log(1 - (|\mathcal{X}|) \cdot k^-) \leq -\log(1 - (|\mathcal{X}|) \cdot k^+) \leq -\log(1 - (|\mathcal{X}|) \cdot k^+) \). Since \( -\log(1 - x) \leq 2x/\ln |\mathcal{X}| \) and \( 1/(1 - x) \leq 2 \) for \( 0 \leq x \leq 1/2 \), we derive \( 1 - \log c \leq 4(|\mathcal{X}|)^2/\ln |\mathcal{X}| \) \( p^+ \) when \( \sum_{y \in \mathcal{Y}, y \neq x} p_y \leq 1/2 \). Consequently Equation 5.11 holds.

### 5.3 Arbitrary channels

Finally, the results of the former Sections are generalized to arbitrary discrete memoryless channels with feedback. Although the theoretical results are valid for all DMC's, the reader is reminded that for some type of channels multiple-repetition strategies are less suitable (see subsection 1.4.4). It is also demonstrated how the symbol precoding distribution should be chosen.
5.3.1 Repetition parameters

As opposed to uniform repetition strategies, for non-uniform repetition strategies the symbol precoding distribution has to be carefully chosen for fine-tuning the transmission rate. The following Lemma shows how the maximal precoding rate with a fixed symbol precoding distribution behaves for increasing repetition parameters. The notation $\approx$ is used to denote equality in order of magnitude, and $H_\mathcal{X}$ stands for the $\mathcal{X}$-ary entropy function.

**Lemma 5.3** Let $q$ be a symbol precoding distribution. Then the maximal precoding rate as a function of the repetition parameters satisfies

$$
\hat{R}_p(q) \doteq H_\mathcal{X}(q) - (1/\ln |\mathcal{X}|) \sum_{x,y \in \mathcal{X}, x \neq y} q_y q_x^{k_{xy}} \quad (5.12)
$$

for increasing repetition parameters.

**Proof:** The set of $\mathcal{X}$-ary sequences of length $L$ with symbol distribution $q$ is denoted by $\mathcal{H}(qL)$. Its cardinality is denoted by $H(qL)$. Remind that $\mathcal{M}(qL)$ is the set of $\mathcal{X}$-ary sequences of length $L$ with symbol distribution $q$ which contain no forbidden subsequences. We derive

$$
\Pr(m \in \mathcal{M}(qL) | m \in \mathcal{H}(qL)) \approx \prod_{1 \leq n \leq L} \Pr(m_n \ldots m_{n+k_{xy}} \neq y x^{k_{xy}} | m \in \mathcal{H}(qL)) \quad (5.13)
$$

$$
= \prod_{1 \leq n \leq L} (1 - \sum_{x,y \in \mathcal{X}, x \neq y} \Pr(m_n \ldots m_{n+k_{xy}} = y x^{k_{xy}} | m \in \mathcal{H}(qL))) \quad (5.14)
$$

$$
\approx (1 - \sum_{x,y \in \mathcal{X}, x \neq y} q_y q_x^{k_{xy}})^L. \quad (5.15)
$$

Consequently $M(qL) \approx H(qL) \cdot (1 - \sum_{x,y \in \mathcal{X}, x \neq y} q_y q_x^{k_{xy}})^L$. Taking logarithms on both sides and dividing by $L$, leads to $\hat{R}_p(q) \doteq H_\mathcal{X}(q) + \log(1 - \sum_{x,y \in \mathcal{X}, x \neq y} q_y q_x^{k_{xy}}) \doteq H_\mathcal{X}(q) - (1/\ln |\mathcal{X}|) \sum_{x,y \in \mathcal{X}, x \neq y} q_y q_x^{k_{xy}}$.

A more detailed proof for the case of binary input alphabet can be found in appendix B.

Consider an arbitrary discrete memoryless channel with channel probabilities $p_{xy} (x \in \mathcal{X}, y \in \mathcal{Y})$. Inspired by the previous Section, a suitable choice for the repetition parameters would be

$$
k_{xy} = (-\log(p_{xy} / \sum_{y \in \mathcal{Y}} p_{xy})) \quad (5.17)
$$

for $x, y \in \mathcal{X}, x \neq y$. The following generalization of Theorem 5.4 shows that this choice is indeed appropriate. The abbreviation $x \neq y$ is used in summations to denote the sum over all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $x \neq y$.

**Theorem 5.5** Consider an arbitrary discrete memoryless channel with channel probabilities $p_{xy} (x \in \mathcal{X}, y \in \mathcal{Y})$. Let $k_{xx} = 0 (x \in \mathcal{X})$ and $k_{xy} = 1 (x \in \mathcal{X}, y \in \mathcal{Y} \setminus \mathcal{X})$. Choose the remaining repetition parameters according to Equation 5.17. Let $\bar{R}$ be the maximum of $\hat{R}(q)$ over all $q \in Q$. Let $C$ be the channel capacity. Then

$$
C - \bar{R} = O\left(\sum_{x \neq y} p_{xy} \right) \left(\sum_{x \neq y} p_{xy} \to 0\right). \quad (5.18)
$$
Note that for small channel error probabilities, \( \log \sum_{y' \in \mathcal{Y}} p_{xy'} \approx -\frac{1}{\ln |\mathcal{X}|} \sum_{y' \in \mathcal{Y}} \ln p_{xy'} \) \((x \in \mathcal{X})\), so we might just as well choose \( k_{xy} = -\log p_{xy} \).

Intuitively this is a plausible choice, because \(-\log p_{xy}\) is the self-information of an \( x \rightarrow y \) error, which is the minimal amount of information that is needed for the transmitter to inform the receiver about the error. The information about the error is given by \( k_{xy} \) repetitions.

In order to prove Theorem 5.5 we need the following Lemma.

**Lemma 5.4** Let \( Q_x = 1/|\mathcal{X}| + \varepsilon_x \) \((x \in \mathcal{X})\), such that \( \sum_{x \in \mathcal{X}} \varepsilon_x = 0 \). Then

\[
H_x(Q) = 1 - \left( \frac{|\mathcal{X}|}{|\mathcal{X}|} \right) \sum_{x \in \mathcal{X}} \varepsilon_x^2.
\]

**Proof:** We derive \( H_x(Q) = -\sum_x Q_x \log Q_x = -\sum_x Q_x (\log(1/|\mathcal{X}|) + \log(1 + |\mathcal{X}| \varepsilon_x)) \approx 1 - (1/\ln |\mathcal{X}|) \sum_x Q_x |\mathcal{X}| \varepsilon_x = 1 - (\sum_x Q_x |\mathcal{X}| \varepsilon_x) \). \( \blacksquare \)

An idea in the proof of Theorem 5.5 is that a channel input distribution \( \Pi \) can be accomplished by choosing a precoding symbol distribution \( q \) such that \( q_x = (Q_x/\tau_x)/\sum_{y \in \mathcal{Y}} (Q_y/\tau_y) \) \((x \in \mathcal{X})\) (see Equation 4.14). In this way a capacity achieving input distribution can be accomplished. However, this is not sufficient for actually achieving channel capacity.

**Proof of Theorem 5.5:** In Section 4.1 was shown that the maximal achievable transmission rate given an arbitrary symbol precoding distribution \( q \) equals \( \bar{R}(p) = \bar{R}_p(q) = \sum_{x \in \mathcal{X}} q_x \log q_x \). Let \( p_x \) \((x \in \mathcal{X})\) be the \( \mathcal{X} \)-ary vector with components \( p_{xy} \) \((y \in \mathcal{Y})\). Since \( k_{xy} = -\log p_{xy} + O(\sum_{y' \in \mathcal{Y} \setminus \{y\}} p_{xy'}) \) for \( x, y \in \mathcal{X}, x \neq y \) and \( p_xx \log p_{xx} = -p_{xx} / \ln |\mathcal{X}| \), we obtain

\[
1/\tau_x = 1 - H_x(p_x) + O(\sum_{y \in \mathcal{Y}, y \neq x} p_{xy}).
\]

Let \( Q \) be an arbitrary channel input distribution. Let \( q \) be the symbol precoding distribution such that \( q_x = (Q_x/\tau_x)/\sum_{y \in \mathcal{Y}} (Q_y/\tau_y) \) \((x \in \mathcal{X})\), then \( \bar{R}(p) = \bar{R}_p(q) = \sum_{x \in \mathcal{X}} q_x \log q_x \). From Lemma 5.3 follows that

\[
\bar{R}(Q) = H_x(Q) = \sum_{x \in \mathcal{X}} H_x(p_x) + O(\sum_{x \in \mathcal{X}} \sum_{y \neq x} k_{xy} p_{xy}).
\]

We will show that the second part is of order \( \sum_{x \neq y} p_{xy} \). We frequently use that \( H_x(s) + H_x(t) \) is of order \( \sum_{x \neq y} p_{xy} \) for arbitrary \( s, t \in \mathcal{X} \). We obtain from Equation 5.20 that \( \tau_x = 1 + H_x(p_x) + O(\sum_{y \neq x} p_{xy}) \). Therefore, \( \tau_x' \sum_{y \in \mathcal{Y}} (Q_y/\tau_y) = 1 + H_x(p_x) + O(\sum_{y \neq x} p_{xy}) \) \((x' \in \mathcal{X})\), and consequently \( \ln(\tau_x' \sum_{y \in \mathcal{Y}} (Q_y/\tau_y)) = H_x(p_x) + O(\sum_{y \neq x} p_{xy}) \). Finally, we derive \( \sum_{x \in \mathcal{X}} (Q_x/\tau_x) \log(\tau_x' \sum_{y \in \mathcal{Y}} (Q_y/\tau_y)) = (1/\ln |\mathcal{X}|) \sum_{x \in \mathcal{X}} H_x(p_x) - \sum_{x \in \mathcal{X}} Q_x H_x(p_x) + O(\sum_{x \neq y} p_{xy}) \).

Summarizing, we obtain

\[
\bar{R}(Q) = H_x(Q) + \sum_{x \in \mathcal{X}} H_x(p_x) Q_x \log Q_x + O(\sum_{x \neq y} p_{xy}).
\]

On the other hand, the channel capacity given channel input distribution \( Q \) equals

\[
C(Q) = -\sum_{y \in \mathcal{Y}} \pi_y \log \pi_y + \sum_{x \in \mathcal{X}} Q_x \sum_{y \in \mathcal{Y}} p_{xy} \log p_{xy},
\]

where \( \pi \) is the channel output distribution with \( \pi_y = \sum_{x \in \mathcal{X}} Q_x p_{xy} \) \((y \in \mathcal{Y})\). Since \( \pi_y \) is equal to \( Q_y + O(\sum_{x \in \mathcal{X}, x \neq y} p_{xy}) \), we obtain

\[
\bar{R}(Q) = C(Q) + O(1/\ln |\mathcal{X}|) \sum_{x \in \mathcal{X}} \varepsilon_x^2.
\]
when \( y \in \mathcal{X} \), we obtain
\[
C(Q) = H_X(Q) - \sum_{x \in \mathcal{X}} Q_x H_X(p_x) + \sum_{x \in \mathcal{X}, y \in \mathcal{Y} \setminus \mathcal{X}} Q_x p_{xy} (\log p_{xy} - \log(\sum_{x' \in \mathcal{X}} Q_{x'p_{x'y}})) + O(\sum_{x,y \in \mathcal{X}, x \neq y} p_{xy}).
\]

We will show that the last (double) sum is of order \( \sum_{x,y \in \mathcal{X}, x \neq y} p_{xy} \). The first observation is that \( \sum_{x' \in \mathcal{X}} Q_{x'p_{x'y}} \geq Q_{xp_{xy}} \), so the double sum is upper bounded by \( \sum_{x \in \mathcal{X}, y \in \mathcal{Y} \setminus \mathcal{X}} p_{xy} \sum_{x \in \mathcal{X}} Q_x \log(1/Q_x) \).

Secondly, \( \sum_{x' \in \mathcal{X}} Q_{x'p_{x'y}} \) is upper bounded by the maximum over all \( p_{xy} \) (\( x' \in \mathcal{X} \)), say \( p_y \). Therefore \( p_{xy} (\log p_{xy} - \log(\sum_{x' \in \mathcal{X}} Q_{x'p_{x'y}})) \) is lower bounded by \( p_{xy} (\log p_{xy} - \log(1/p_y)) \) which is lower bounded by \( -p_{xy}/(e \log |\mathcal{X}|) \) because \( 0 \leq p_{xy}/p_y \leq 1 \). So the double sum is lower bounded by \( -(e/\log |\mathcal{X}|) \sum_{y \in \mathcal{Y} \setminus \mathcal{X}} p_{xy} \).

Summarizing, we obtain
\[
C(Q) = H_X(Q) - \sum_{x \in \mathcal{X}} Q_x H_X(p_x) + O(\sum_{x \in \mathcal{X}, y \neq x} p_{xy}). \tag{5.22}
\]

By substituting the uniform distribution in Equation 5.22 it is concluded that \( C \geq 1 - (1/|\mathcal{X}|) \sum_{x \in \mathcal{X}} H_X(p_x) + O(\sum_{x \neq y} p_{xy}) \). Let \( \hat{Q}_x = 1/|\mathcal{X}| + \varepsilon_x \) (\( x \in \mathcal{X} \)) be the channel input distribution that maximizes \( C(Q) \). From the lower bound on the channel capacity and Lemma 5.4 follows that \( \varepsilon_y \) should be of order \( \sqrt{\sum_{x \in \mathcal{X}} H_X(p_x)} \) for each \( y, y \in \mathcal{X} \). Consequently, for \( Q = \hat{Q}_x \), the term \( \sum_{x \in \mathcal{X}} H_X(p_x) \log(\sum_{y \in \mathcal{Y} \setminus \mathcal{X}} p_{xy}) \) in Equation 5.21 is equal to the term \( -\sum_{x \in \mathcal{X}, y \in \mathcal{Y} \setminus \mathcal{X}} Q_x H_X(p_x) \) in Equation 5.22 up to order \( \sum_{x \neq y} p_{xy} \).

It only remains to show that \( \sum_{x, y \in \mathcal{X}, x \neq y} h_{xy} q_{xy} \) is of order \( \sum_{x, y \in \mathcal{X}, x \neq y} p_{xy} \) when \( q_x = \hat{q}_x = (\hat{Q}_x/\tau_x)/\sum_{y \in \mathcal{Y} \setminus \mathcal{X}} (\hat{Q}_y/\tau_y) \) (\( x \in \mathcal{X} \)). Using Equation 5.20, it follows that \( \hat{q}_y = 1/|\mathcal{X}| + O(\sqrt{\sum_{x \in \mathcal{X}} H_X(p_x)}) \) (\( y \in \mathcal{X} \)). We also remind that \( h_{xy} = -\log p_{xy} + O(\sum_{x' \in \mathcal{X}} H_X(p_{x'y})) \) for \( x, y \in \mathcal{X}, x \neq y \). Therefore \( q_{xy} = O(p_{xy}) \) because \( -\log p_{xy} \log(1 + \sqrt{\sum_{x' \in \mathcal{X}} H_X(p_{x'y})}) \) becomes arbitrarily small.

### 5.3.2 Symbol precoding distribution

At the beginning of this Section it was shown how the repetition parameters should be chosen given an arbitrary discrete memoryless channel. For asymmetric channels the symbol precoding distribution is important for fine-tuning the transmission rate. As was mentioned at page 69, a symbol precoding distribution \( q \) induces a channel input distribution \( p \) such that \( p_x = q_x \tau_x/\sum_{y \in \mathcal{Y} \setminus \mathcal{X}} q_y \tau_y \) (\( x \in \mathcal{X} \)). So choosing \( q \) such that \( p \) equals the capacity achieving channel input distribution seems appropriate. This idea was used in the proof of Theorem 5.5.

However, there is a way to compute the exact value of the symbol precoding distribution that maximizes the achievable transmission rate. It follows easily from Theorem 8 of Shannon in [81, p. 400] which is related to Shannon’s Theorem 1 [81, p. 384]. Although Shannon only used integer durations, the result also holds for real valued durations [101]. We describe these Theorems using Shannon’s notation and show how they can be used to compute the optimal symbol precoding distribution.

Shannon describes how the capacity of a discrete noiseless channel with constraints on the input symbols can be computed. The system is represented by a finite directed graph. The allowed sequences correspond with the paths through the graph. Each edge is labeled by the symbol that is produced when ‘walking’ over this edge. Each symbol has a certain duration that can be seen as the number of transmissions over the noiseless channel needed to send the symbol. When two or more different edges are labeled by the same symbol, this symbol may have different durations for each edge. Let \( N(T) \) be the number of allowed...
sequences with a total duration of $T$, then the capacity of the discrete noiseless channel is defined as

$$C = \lim_{T \to \infty} \frac{\log N(T)}{T}. \quad (5.23)$$

Shannon's Theorem 1 shows how $C$ can be computed.

**Theorem 5.6 (Shannon)** Let $b_{ij}^s$ be the duration of the $s^{th}$ symbol which is allowable in state $i$ and leads to state $j$. Let $M(W)$ be the matrix with entries $M_{ij} = \sum_s W^{-b_{ij}^s} - \delta_{ij}$, then the channel capacity $C$ is equal to $\log W_0$ where $W_0$ is the largest real root of the Equation

$$\det M(W) = 0. \quad (5.24)$$

Theorem 5.6 is similar to our Theorem 4.1. Our constrained sequences are generated by a graph with $k_{ji} - 1$ edges going from state $i$ to state $j$, the $s^{th}$ label being $j^s$ with duration $b_{ij}^s = s \cdot \tau_j \ (1 \leq s < k_{ji})$. Unfortunately, Shannon's proof [81, pp. 418-419] of Theorem 5.6 is only valid for integer durations. On the other hand, Theorem 4.1 is only valid for our type of constrained sequences, although it can be generalized to arbitrarily constrained sequences with fixed duration per symbol [101]. Note that fixing the duration per symbol does not reduce the applicability of the Theorem, because when two or more different edges are labeled by the same symbol with different durations, these can be considered as being different symbols. It is easily seen that Theorem 5.6 and Theorem 4.1 lead to the same answer by observing that $D_{xy} = z_{2xy}^k - z_x = (z_{xy} - 1) \sum_{1 \leq s < k_{xy}} z_x^s$, so $\det D = 0$ is equivalent to $M(z_{xy}) = 0$, where $M(z_{xy}) = \sum_{1 \leq s < k_{xy}} z_x^s$ when $x \neq y$ and $M(z_{xx}) = -1$. Also observe that $z_x = (|X| \tau)^{-1}$, $R = \log W$.

Suppose we would assign probabilities to each edge in the graph. The graph would then become a source with a certain entropy $H$. Let $p_{ij}^s$ be the probability of going from state $i$ to state $j$ via the $s^{th}$ edge. The state probability distribution $P$ can be computed by solving $\sum_i P_i \sum_s p_{ij}^s = P_j$ such that $\sum_i P_i = 1$. The entropy of the source per time unit is then given by

$$H = \frac{-\sum_i P_i \sum_s p_{ij}^s \log p_{ij}^s}{\sum_i P_i \sum_s p_{ij}^s b_{ij}^s}. \quad (5.25)$$

Shannon's Theorem 8 shows that the source entropy can never exceed the capacity and that there is a probability assignment such that $H = C$.

**Theorem 5.7 (Shannon)** Given the conditions of Theorem 5.6, if we assign $p_{ij}^s$ such that

$$\log p_{ij}^s = \log \left(\frac{B_j^s}{B_i^s}\right) - b_{ij}^s C, \quad (5.26)$$

where $B$ satisfies

$$M(W_0) \cdot B^T = 0, \quad (5.27)$$

then $H$ is maximized and equal to $C$.

Shannon's proof [81, pp. 421-423] of Theorem 5.7 does not use the integrality of the durations, and since we showed that Theorem 5.6 also holds for real valued durations, Theorem 5.7 is also valid for real valued durations.
Figure 5.2: A binary channel

We are now ready to show how an optimal symbol precoding distribution can be computed. We apply Theorem 5.7 to our constrained sequences, i.e. \( b_{xy} = s \cdot \tau_y \) for \( 1 \leq s < k_{yx}, x, y \in \mathcal{X} \), and compute the state probability distribution \( \overline{P} \). Since a 'walk' over the \( s \)th edge from state \( x \) to state \( y \) produces a subsequence consisting of \( s \) consecutive symbols \( y \), the induced symbol precoding distribution \( \overline{q} \) satisfies

\[
q_y = \theta \sum_{x \neq y} P_x \sum_{1 \leq s < k_{yx}} s p_y^s, \tag{5.28}
\]

where positive constant \( \theta \) is chosen such that \( \sum_{y \in \mathcal{X}} q_y = 1 \).

### 5.3.3 Example

In this subsection an example is given for the results that are derived in this Section. We take a binary channel \( (\mathcal{X} = \mathcal{Y} = \{0, 1\}) \) with \( p_{01} = 0.001 \) and \( p_{10} = 0.003 \) as depicted in Figure 5.2. In subsection 5.3.1 it is shown that the repetition parameters should be chosen such that \( k_{xy} = \left(-\log_2 p_{xy}\right) \), so we choose \( k_{01} = 10 \) and \( k_{10} = 8 \).

The outcomes of the forthcoming numerical computations, as suggested in subsection 5.3.2, are depicted in table 5.2 correct to ten decimal places. We compute \( \tau_0 = 1/(1 - k_{01}p_{01}) \) and \( \tau_1 = 1/(1 - k_{10}p_{10}) \). The matrix \( M(W) \) from Theorem 5.6 has entries \( M(W)_{00} = \sum_{1 \leq s < 8} W^{-sz_1}, M(W)_{10} = \sum_{1 \leq s < 10} W^{-sz_0}, \) and \( M(W)_{00} = M(W)_{11} = -1 \). Solving for \( \det M(W) = 0 \) yields \( W_0 \) and the maximal achievable rate \( \bar{R} = \log_2 W_0 \). In table 5.2 the capacity \( C \) of the channel in Figure 5.2 is also given in order to compare with Theorem 5.5. This comparison suggests that the (hidden) constant in Equation 5.18 is rather small.

Solving for \( B \) yields \( B_0/B_1 = M(W_{01}) \), and consequently the optimal source probabilities \( p_{01}^* = (B_0/B_1)W^{-sz_0} \) (\( 1 \leq s < 10 \)) and \( p_{10}^* = (B_1/B_0)W^{-sz_1} \) (\( 1 \leq s < 8 \)). The source has a uniform state probability distribution. From Equation 5.28 finally follows \( \overline{q} \).

On the other hand, the capacity achieving channel input distribution \( \overline{\rho} \) for the channel of Figure 5.2 can be computed, which suggests that we should use the symbol precoding distribution \( \overline{q} \) as computed using Equation 4.14. The precoding rates for symbol precoding distributions \( q \) and \( \overline{q} \) are computed according to the method described in the proof of Lemma B.1. The difference between \( \bar{R}(\overline{q}) \) and \( \bar{R} = \bar{R}(\overline{q}) \) turns out to be relatively small.
5.4 Conclusions

By considering an increasing class of DMC's, namely the BSC, the class of uniform channels, and finally the class of arbitrary DMC's, it was shown that the repetition parameters should be chosen close to the (absolute value of the) logarithm of the channel error probabilities to obtain a maximal transmission rate. Since the negative logarithm of a channel error probability equals the self-information of that error, this choice of the repetition parameter indeed accomplishes the minimal amount of repetitions that is needed to inform the receiver about the channel error. It was illustrated for each particular class of channels, that by properly choosing the repetition parameters, the difference between the channel capacity and the transmission rate is of the same order as the channel error probabilities which is the best achievable result one could expect (see e.g. Figure 4.1). Furthermore, it was shown by using a result of Shannon how the optimal symbol precoding distribution can be computed when the channel is non-uniform. Finally, the results were illustrated by an example.
Chapter 6

Variations and extensions

We present an extension of the multiple-repetition feedback strategies. Although the results are easily generalized to arbitrary (mind the practical restriction as mentioned in subsection 1.4.4) discrete memoryless channels, we restrict ourselves to the binary symmetric channel \( \mathcal{X} = \mathcal{Y} = \{0,1\} \). A class of alternative strategies is derived also. The new strategies are suitable for channels with high channel error probabilities, and require less than one bit information feedback per transmission.

6.1 Generalized scheme

In [11] Berlekamp derived upper bounds on the correctable error fraction of arbitrary block coding strategies for the binary symmetric channel with noiseless, delayless feedback. These bounds are depicted in Figure 1.6, where \( f \) denotes the correctable error fraction and \( R \) the transmission rate. We recapitulate the Definition of correctable error fraction (see Definition 1.5). A coding strategy with block length \( N \) that can transmit \( M \) different messages has a transmission rate \( R = \log_2 M/N \). When such a coding strategy achieves a correctable error fraction \( f \), then all error patterns in a block with a total number of \( e \) errors can be corrected as long as \( e/N < f \). When the correctable error fraction \( f \) can be achieved for arbitrary long block length while the transmission rate if fixed, the pair \( (f,R) \) is called achievable. Figure 1.6 shows the region of achievable pairs. It is bounded by the Volume bound \( R = 1 - h(f) \), and by the Tangent bound, a straight line through \((1/3,0)\) that is tangent to the Volume bound.

Since Berlekamp’s notion of achievability is more strict than Shannon’s requirement that the error probability vanishes with increasing block length, Berlekamp’s achievable region is smaller than the capacity of the binary symmetric channel (as was previously mentioned at page 17).

A binary symmetric block repetition strategy with repetition number \( k \) can correct all errors as long as \( ke < N - L \) (see subsection 2.1.1). In Figure 2.3 it was shown that some error patterns with \( ke = N - L \) cannot be corrected. So the maximal correctable error fraction is \( f = (1 - L/N)/k \), given a transmission rate \( R = \log_2 M_L/N \), where \( M_L \) is the number of precoded sequences of length \( L \). If we increase the block length while keeping \( \omega = L/N \) constant (similar as on page 66), the achievable pair \(((1-\omega)/k, \omega \log_2 c_k)\) is obtained, where \( c_k \) is the solution \( c, c > 1 \) of \( c^k - 2c^{k-1} + 1 = 0 \). These achievable pairs form a straight line through \((1/k,0)\) (\( \omega = 0 \)) that is tangent to the Volume bound at \( \omega = 1 - k + kc_k/2 \), as follows from subsection 4.4.1. Consequently, the binary symmetric
block repetition strategy with repetition number \( k = 3 \) achieves the Tangent bound.

The interesting question arises whether a block coding strategy can be constructed that goes beyond the Tangent bound in the Shannon sense, or: can we construct block coding strategies that achieve positive rates on a binary symmetric channel with feedback for channel error probability \( p > 1/3? \)

### 6.1.1 Description

One of the first ideas that comes to mind is to combine \( \nu \) transmissions. In other words, in stead of considering the BSC, one should consider BSC\(^\nu\) (the DMC with \( 2^\nu \) inputs and outputs that combines \( \nu \) transmissions on a BSC and has capacity \( \nu \cdot (1 - h(p)) \)). A multiple-repetition strategy could then be used on BSC\(^\nu\). The advantage of considering BSC\(^\nu\) is that the (highest) error probability has reduced from \( p \) to \( (1-p)^\nu \). The disadvantage is that we can only obtain a positive transmission rate when \( \sum_y p_{xy} k_{xy} < 1 \) (see Equation 4.4) for each input symbol \( x \in \mathcal{X} \). Even when all \( k_{xy} (y \neq x) \) are chosen equal to 2, this requirement is equivalent to \( (1-p)^\nu > \frac{1}{2} \) for the BSC\(^\nu\). Therefore, this idea is not useful when \( p > 1/3 \).

A different approach is presented that at the same time reduces the amount of required feedback. Consider an arbitrary strategy (not necessarily a multiple-repetition strategy) for a binary (not necessarily symmetric) channel with feedback. Let \( \rho = (\rho_0, \rho_1) \) be the vector of channel error probabilities, where \( \rho_0 = \rho_{01} \) and \( \rho_1 = \rho_{10} \). Let \( R(p) \) be the transmission rate and \( \rho(p) = (\rho_0(p), \rho_1(p)) \) be the channel input distribution induced by the strategy averaged over all possible messages.

We generalize the strategy as follows: let \( \nu \) be an arbitrary positive integer. Suppose that binary symbol \( x \) has to be transmitted at a given time instance. Instead of simply transmitting \( x \), we transmit \( x^\nu \). When this is received as \( 0^\nu \) or \( 1^\nu \), the receiver decides that respectively a zero or a one was transmitted. Otherwise the receiver makes no decision, and the symbol \( x \) is retransmitted in the same way until \( 0^\nu \) or \( 1^\nu \) is received after which the receiver decides for zero or one respectively. Let \( R^\nu(p) \) be the transmission rate of the new generalized strategy, then

\[
R^\nu(p) = \frac{\rho_0(p)(\rho_0^\nu + (1-\rho_0)^\nu) + \rho_1(p)(\rho_1^\nu + (1-\rho_1)^\nu)}{\nu} \cdot R(p), \tag{6.1}
\]

where \( \rho_i^\nu = \frac{\rho_i^\nu}{\rho_i^\nu + (1-\rho_i)^\nu} (i = 0, 1) \). Note that for \( \nu = 1 \), the original strategy is obtained. Since the transmitter only needs to know whether the receiver received \( 0^\nu \), or \( 1^\nu \), or something else, the required amount of feedback is \( \log_2(3)/\nu \) bit per transmission (\( \nu > 1 \)). Since the generalization introduces large redundancy it is not suitable for low channel error probabilities.

The above mentioned construction is applied to the binary symmetric repetition strategy on the binary symmetric channel with channel error probability \( p \), \( 0 \leq p \leq 1/2 \). Note that the induced channel input distribution \( \rho \) is uniform. If we generalize the symmetric repetition strategy with repetition parameter \( k \), we obtain a transmission rate of

\[
R^\nu(p) = \frac{(1-p)^\nu - (k-1)p^\nu}{\nu} \cdot \log_2 c_k. \tag{6.2}
\]

For \( \nu = 2, 3, 4 \) and \( k = 3 \), \( R^\nu \) is depicted in Figure 6.1. In Figure 6.1 it can be seen that the transmission rate of the new strategies indeed exceeds the Tangent bound and approximates the Volume bound.
Figure 6.1: Performance of generalized strategies
Figure 6.2: Meta channel

If we take a closer look at our generalized strategy, we see that we actually use the channel that is depicted in Figure 6.2, where \( p_e = 1 - p^\nu - (1 - p)^\nu \) for \( i = 0, 1 \). The output symbol \( e \) denotes an erasure. The sequences \( 0^\nu \) and \( 1^\nu \) are replaced by the 'meta symbols' \( 0 \) and \( 1 \) respectively. Let \( C(p) = 1 - h(p) \) be the capacity of the binary symmetric channel, then the capacity of the meta channel equals \( (p^\nu + (1 - p)^\nu) \cdot C(p^\nu/(p^\nu + (1 - p)^\nu)) \). Since one transmission in the meta channel takes \( 11 \) transmissions in the binary symmetric channel, we obtain the following bound:

\[
R^\nu(p) \leq C^\nu(p) = \frac{p^\nu + (1 - p)^\nu}{\nu} \cdot C \left( \frac{p^\nu}{p^\nu + (1 - p)^\nu} \right). \tag{6.3}
\]

Note that \( C^1(p) = C(p) \). The capacities \( C^\nu \) for \( 1 \leq \nu \leq 4 \) are depicted in Figure 6.3. In appendix C a first order approximation of \( C^\nu(p) \) is derived which suggests that \( C^\nu(p) \geq C^{\nu+1}(p) \) with equality only for \( p = 1/2 \). Therefore, it seems that a generalized strategy \( (\nu > 1) \) cannot achieve the Volume bound for \( p < 1/2 \). However, since \( C(p) - R(p) = D(p||p_k) \) (see Theorem 5.2), it follows that

\[
C^\nu(p) - R^\nu(p) = \frac{(1 - p)^\nu + p^\nu}{\nu} \cdot D \left( \frac{p^\nu}{p^\nu + (1 - p)^\nu} || p_k \right), \tag{6.4}
\]

so that in particular \( R^\nu(p) = C^\nu(p) \) when \( p^\nu/(p^\nu + (1 - p)^\nu) = p_k \). Given \( k \), and consequently \( p_k \), the Equation \( p^\nu/(p^\nu + (1 - p)^\nu) = p_k \) has exactly one solution for \( p \). The generalized strategy determined by parameters \( k \) and \( \nu \) will perform well on the BSC for this \( p \), but will not achieve channel capacity.

### 6.1.2 Block error exponent

Suppose we use the strategy with repetition number \( k \) and parameter \( \nu \) for the binary symmetric channel with channel error probability \( p \). Let \( N = n \cdot \nu \) be the block length (\( n \) is the number of meta symbols in a block). Let \( l \) be the number of (meta) information symbols. Let \( e \) be the number of decision errors made by the receiver during one block. Let \( u \) be the number of erasures in a block. Then a block is decoded correctly whenever

\[
l + ke + u < n. \tag{6.5}
\]
Equation 6.5 follows automatically from Equation 2.2 when considering the generalized strategy as a multiple-repetition strategy on the meta channel depicted in Figure 6.2 with $k_{e} = k_{ie} = 1$. An example where $k = 3$, $\nu = 2$, $n = 10$, $e = 1$, $u = 2$, and $l = 4$ is depicted in Figure 6.4.

Let $E$ and $U$ be the random variables denoting the number of decision errors in a block and the number of erasures in a block, respectively. The random variables $E$ and $U$ satisfy the joint probability distribution

$$
\Pr\{E = e, U = u\} = \binom{n}{e, u} \cdot p^e \cdot (1 - p^\nu) \cdot (1 - p)^e \cdot (1 - p)^u \cdot (1 - p)^{n-e-u},
$$

(6.6)

where $e$ and $u$ are integers such that $e, u \geq 0$ and $e + u \leq n$. The block error probability $P_e$ satisfies

$$
P_e \leq \Pr\{l + k \cdot E + U \geq n\}.
$$

(6.7)
We fix \( \omega = l/n \) and let the block length increase. Then the upper bound in Equation 6.7 is exponentially tight. The transmission rate satisfies \( R = \frac{\nu}{\nu} \cdot \log_2 c_k \) provided \( 0 \leq \omega < (1 - p)^\nu - (k - 1)p^\nu \). The error exponent

\[
E^\nu(\omega) = \lim_{N \to \infty} -\frac{\log_2 \Pr\{k \cdot E + U \geq n(1 - \omega)\}}{N}
\]  

(6.8)

where \( n \) and \( N \) are related by \( N = n \cdot \nu \), is obtained in the following Theorem.

**Theorem 6.1** Let \( 0 \leq \omega < (1 - p)^\nu - (k - 1)p^\nu \). Let \( y > 1 \) be the solution of

\[
y(1 - p^\nu - (1 - p)^\nu)\omega + y^k p^\nu(k - 1 + \omega) = (1 - p)^\nu(1 - \omega),
\]  

(6.9)

then

\[
E^\nu(\omega) = -\frac{1}{\nu} \log_2(1 - p^\nu + (1 - p^\nu - (1 - p)^\nu)y + p^\nu y^k) + \frac{1 - \omega}{\nu} \log_2 y.
\]  

(6.10)

**Proof:** Let \( W \) be the random variable \( k \cdot E + U \). We show that \( -\log_2(\Pr\{W \geq n(1 - \omega)\})/N \) is equal to the right-hand side of Equation 6.10 when \( N \to \infty \). We first show the \( \leq \)-inequality.

We compute the Chernoff bound for \( \Pr\{W \geq n(1 - \omega)\} \). Let \( g_W(s) \) be the expected value of \( e^{s \cdot W} \) for \( s > 0 \), then \( \Pr\{W \geq n(1 - \omega)\} \leq g_W(s)e^{-sn(1 - \omega)} \). We compute \( g_W(s) = ((1 - p)^\nu + (1 - p^\nu - (1 - p)^\nu)e^s + p^\nu e^{sk})^n \). One finds that \( g_W(s_0)e^{-sn(1 - \omega)} \) is minimized for \( s_0 = \ln y \), where \( y > 1 \) is the solution of Equation 6.9. The value \( -\log_2(g_W(s_0)e^{-sn(1 - \omega)})/N \) equals the right-hand side of Equation 6.10. The Chernoff bound is only useful when the expected value of \( W \) is smaller than \( n(1 - \omega) \), or in other words \( \omega < (1 - p)^\nu - (k - 1)p^\nu \).

Finally it is shown that \( \lim_{N \to \infty} -(1/N) \log_2(\Pr\{W \geq n(1 - \omega)\}) \) is greater than or equal to the right-hand side of Equation 6.10. We fix \( \epsilon = \epsilon/n \) and \( v = u/n \) and compute \( -\log_2(\Pr\{E = e, U = u\})/N \) for \( N \to \infty \). From Equation 6.6 follows that this limit equals

\[
-\frac{\epsilon}{\nu} \log_2 p - \frac{v}{\nu} \log_2(1 - p^\nu - (1 - p)^\nu) - (1 - \epsilon - v) \log_2(1 - p),
\]  

(6.11)

where \( \theta \) is the entropy \( -\epsilon \log_2 \epsilon - v \log_2 v - (1 - \epsilon - v) \log_2(1 - \epsilon - v) \). The maximum of Equation 6.11 under the condition \( k \epsilon + v \geq 1 - \omega \) (corresponding with \( k \cdot E + U \geq n(1 - \omega) \)) is obtained (e.g. by means of the Laplace optimization method) when both \( k \epsilon + v = 1 - \omega \) and

\[
\epsilon \cdot (1 - p^\nu - (1 - p)^\nu) = v^k \cdot \left(\frac{(1 - p)^\nu}{1 - \epsilon - v}\right)^{k-1} \cdot p^\nu.
\]  

(6.12)

We show that when both \( k \epsilon + v = 1 - \omega \) and Equation 6.12 are satisfied, Equation 6.11 equals the right-hand side of Equation 6.10. Let \( y = \frac{(1 - p)^\nu}{1 - p^\nu - (1 - p)^\nu} \cdot \frac{v}{1 - \epsilon - v} \). It then follows from Equation 6.12 that \( y^k = \frac{(1 - p)^\nu}{1 - p^\nu - (1 - p)^\nu} \cdot \frac{v}{1 - \epsilon - v} \). The relation \( k \epsilon + v = 1 - \omega \) is then used to show that \( y \) is the solution of Equation 6.9. Finally, Equation 6.11 can be rewritten as the right-hand side of Equation 6.10 using \( \frac{1 - \omega}{\nu} \log_2 y = \frac{\epsilon}{\nu} \log_2 y^k + \frac{v}{\nu} \log_2 y \).

Note that for \( \nu = 1 \) the error exponent \( E^1(\omega) \) equals \( D((1 - \omega)/k||p) \), which is the straightforward Chernoff bound. For \( k = 3 \) and \( p = p_k = (3 - \sqrt{5})/4 \approx 0.19 \), the error exponent is plotted against the transmission rate in Figure 6.5 for \( \nu = 1, 2, 3 \).
6.2 Alternative schemes

The generalized strategies have the property that the receiver postpones his decision about the transmitted symbol until sufficient information is received. Erroneous decisions are corrected by the repetition mechanism. Using this idea a lot of alternative strategies can be thought out. Some of them are described in this Section. We only consider binary symmetric strategies, but the ideas are generalized easily to arbitrary strategies. The transmission rates are computed by the following formula:

\[
R = \frac{1 - k p_d}{\eta} \cdot \log_2 c_k, \tag{6.13}
\]

where \( \eta \) is the expected number of transmissions until the receiver decides what symbol was transmitted, and \( p_d \) is the probability that the receiver makes an erroneous decision. E.g. for the generalized strategies with parameter \( \nu \), we find \( \eta = \nu/(p^\nu + (1 - p)^\nu) \) and \( p_d = p^\nu/(p^\nu + (1 - p)^\nu) \). An overview of the transmission rates of our examples is depicted in Figure 6.6. We chose the repetition parameter \( k = 3 \) in Figure 6.6. The transmission rate of the generalized strategy with parameter \( \nu = 2 \) as depicted in Figure 6.1, exceeds the transmission rate of each of our examples in Figure 6.6 when the channel error probability is at least 0.27.

6.2.1 Majority decision

In this class of strategies, the transmitter sends each symbol a fixed number of times. The receiver then takes a majority vote or postpones its decision further and asks for a retransmission. Consider for example a strategy where each symbol is transmitted four times. If three or four zeros are received, the receiver decides for zero. If not more than one zero is received, a one is decided for, and if two zeros and two ones are received, the receiver asks for a retransmission. For this strategy we compute \( \eta = 4/\Pr\{\text{no retransmission}\} = \)
Figure 6.6: Some alternative schemes
Figure 6.7: A collective scheme

\[
\frac{4}{1 - 6p^2(1 - p)^2} \text{ and } p_d = \Pr\{\text{at least three channel errors}\}/\Pr\{\text{no retransmission}\} = (p^4 + 4(1 - p)p^3)/(1 - 6p^2(1 - p)^2).
\]

We obtain the transmission rate \( R_{\text{md}} \):

\[
R_{\text{md}} = \left\{ 1 - 6p^2(1 - p)^2 - k(p^4 + 4(1 - p)p^3) \right\} \log_2 c_k
\]

(6.14)

### 6.2.2 Collective

This class of strategies has the property that the receiver waits until a fixed number of equal symbols is received. For example, the scheme where the receiver waits until two equal symbols are collected, is depicted by means of a Markov chain in Figure 6.7. In Figure 6.7 the notation \( \bar{x} \) is used to denote an erroneous transmission of symbol \( x \). From Figure 6.7 we derive \( p_d = p^2 + p(1 - p)p + (1 - p)p^2 = p^2(3 - 2p) \) and \( \eta = 2 \cdot ((1 - p)^2 + p^2) + 3 \cdot 2p(1 - p) = 2(1 + p - p^2) \). This leads to a transmission rate \( R_{\text{col}} \) of

\[
R_{\text{col}} = \frac{1 - kp^2(3 - 2p)}{2(1 + p - p^2)} \log_2 c_k
\]

(6.15)

### 6.2.3 Consecutive

Finally, the receiver could wait until a fixed number of consecutive equal symbols is received. An example where the receiver waits until two consecutive equal symbols are received, is depicted in Figure 6.8 by means of a Markov chain. The states \( xx \) and \( \bar{x}x \) refer to the last two transmissions. We deduce from Figure 6.8 that \( p_d = p^2(1 + (1 - p))\sum_{i \geq 0}(p(1 - p))^i = p^2(2 - p)/(1 - p + p^2) \). Furthermore, we compute \( \eta = (p^2 + (1 - p)^2)\sum_{i \geq 1}2i(p(1 - p))^{i-1} + \sum_{i \geq 1}(2i+1)(p(1 - p))^{i} = (2+p-p^2)/(1-p+p^2) \). By Equation 6.13 we obtain a transmission rate

\[
R_{\text{con}} = \left\{ 1 - p + p^2 - k(p^2(2 - p)) \right\} \log_2 c_k
\]

(6.16)
6.3 Conclusions

We managed to find a generalization of multiple-repetition strategies that enables us to go beyond Berlekamp's Tangent bound in Shannon's sense of achievability. The error exponents of these generalized strategies were derived. In addition, several variations of multiple-repetition strategies were considered that exceed also Berlekamp's Tangent bound. All new strategies have the property that they behave well for high channel error probabilities ($p > 0.25$), and that they require less feedback than usual. Note that for small channel error probabilities, the original multiple-repetition strategies achieve much higher transmission rates.
Chapter 7

Practical aspects and possible applications

So far we were only interested in theoretical results concerning multiple-repetition feedback strategies. Without neglecting the scientific contribution of the results in this thesis, it is useful to describe how multiple-repetition feedback strategies could be of use in practical situations. Much attention is paid to the different precoding methods.

7.1 Further research

Research on coding for discrete memoryless channels with noiseless feedback is useful for research on related areas. It provides an interesting comparison with the one-way situation. Furthermore, by extending one-way coding problems with a noiseless feedback link, one might be able to derive new bounds.

On the other hand, results on coding for discrete memoryless channels with noiseless instantaneous feedback can be seen as preliminary results for other research areas like noisy, limited, or delayed feedback, or channels with memory, or channels with unknown parameters.

7.2 Precoding

Before describing the use of multiple-repetition feedback strategies in practice, we consider the problem of precoding. Whereas the precoding scheme is less important for the kind of theoretical problems that are considered in (the rest of) this thesis, it certainly is a key issue when implementing multiple-repetition feedback strategies. Without completely solving the problem of precoding, some possible solutions are presented and compared. The choice for a particular precoding scheme will depend on the application one has in mind.

A precoding system consists of a precoder and an inverse precoder (also called postcoder) as is shown in Figure 7.1. The precoder constructs a precoded sequence (without forbidden subsequences) from a given message, and the inverse precoder produces a message from a precoded sequence. In multiple-repetition block coding schemes, the length of the precoded sequence is fixed and finite. Recursive schemes on the other hand need an ongoing stream of information symbols, so the (inverse) precoder must continually transform symbols of an input stream into output symbols. One could implement a recursive precoding scheme by a
\[ \mathcal{M} = \{0, 1, \ldots, M - 1\} \]

\[ m \in \mathcal{M} \]

\[ \hat{p} = \hat{p}_1 \ldots \hat{p}_L \]

\[ \hat{\hat{p}} = \hat{\hat{p}}_1 \ldots \hat{\hat{p}}_L \]

\[ \hat{\hat{m}} \in \mathcal{M} \]

\[ \mathcal{M} = \mathcal{X}^{\infty} \]

\[ m \in \mathcal{M} \]

\[ \hat{p} = \hat{p}_1, \hat{p}_2, \ldots \]

\[ \hat{\hat{p}} = \hat{\hat{p}}_1, \hat{\hat{p}}_2, \ldots \]

\[ \hat{\hat{m}} \in \mathcal{M} \]

Figure 7.1: Precoding
block precoding scheme with a continuous stream of (large) blocks, but one should be aware that forbidden subsequences are not allowed to occur between two consecutive blocks.

The message set for a block coding scheme is finite. It may consist of the integers from 0 to \( M - 1 \), but could just as well consist of e.g. all \( X \)-ary sequences of a certain finite length. The message set in case of a recursive scheme on the other hand should be countably infinite. It could consist of all infinite \( X \)-ary sequences, or e.g. all infinite binary sequences.

An overview is presented of precoding systems. For simplicity we restrict ourselves to binary symmetric precoded sequences with parameter \( k \geq 3 \), i.e. the set \( \mathcal{P} \) of all binary sequences such that \( 01^k \) and \( 10^k \) do not occur as a subsequence. Most precoding systems can be generalized to arbitrary precoded sequences.

### 7.2.1 Enumerative coding

In his Ph.D. thesis [7] Becker describes an enumerative scheme for (inverse) precoding. At that time (1973) Cover [22] independently discovered the same coding scheme. Let \( \mathcal{P}_L \) be the set consisting of all elements of \( \mathcal{P} \) of length \( L \). The idea is to order all elements of \( \mathcal{P}_L \). The ordering is done lexicographically, like in an ordinary dictionary, under the interpretation that \( 0 < 1 \). The index of a precoded sequence \( p \) then is the number of elements of \( \mathcal{P}_L \) that are smaller than \( p \) in a lexicographical sense. Let \( M_L \) be the number of elements of \( \mathcal{P}_L \). Then each message \( m \in \{0, 1, \ldots, M_L - 1\} \) is precoded to the element of \( \mathcal{P}_L \) with index \( m \). The inverse precoder simply outputs the index of the (estimated) precoded sequence. The key formula of enumerative coding that relates a precoded sequence \( p \) with its index \( i(p) \) is

\[
i(p) = \sum_{l=1}^{L} p_l \cdot W(0, p_{l+1}, \ldots, p_L),
\]

(7.1)

where \( W(p_l, p_{l+1}, \ldots, p_L) \) is the number of elements of \( \mathcal{P}_L \) which end in \( p_l, p_{l+1}, \ldots, p_L \). When all the numbers \( W(0, p_{l+1}, \ldots, p_L) \) are given, the precoding algorithm follows easily from Equation 7.1. The inverse precoding algorithm is depicted below.

**Enumerative inverse precoding**

\[
\{0 \leq m < M_L\}
\]

BEGIN

\[
s := m;
\]

FOR \( l := L \) DOWNTO 1 DO

IF \( W(0, p_{l+1}, \ldots, p_L) \leq s \) THEN

BEGIN

\[
s := s - W(0, p_{l+1}, \ldots, p_L);
\]

\[
p_l := 1
\]

END

ELSE \( p_l := 0 \)

END

\[
\{m = i(p)\}
\]

Becker [7, p. 35] showed that when the number \( h, 0 \leq h \leq L - l \), is such that \( 0 = p_{l+1} = \ldots = p_{l+h} \) and either \( p_{l+h+1} = 1 \) or \( l + h = L \),

\[
W(0, p_{l+1}, \ldots, p_L) = \begin{cases} (M_l - \sum_{i=1}^{h} M_{i+i-k})/2 & \text{for } 1 \leq h \leq k - 2, \\ 1 & \text{for } h \geq k - 1. \end{cases}
\]

(7.2)
Therefore only the numbers $M_l$ for $1 \leq l \leq L$ have to be determined. They can be computed recursively by

$$M_l = \begin{cases} 2^l & \text{for } 1 \leq l \leq k, \\ 2 \cdot M_{l-1} - M_{l-k} & \text{for } l > k. \end{cases}$$ \tag{7.3}

(7.3)

It seems a good idea to precompute and store the numbers $M_l$ for $1 \leq l \leq L$ both at the transmitter and the receiver's side to reduce the time complexity of the precoding system. The enumerative precoding system then needs a memory consisting of $L$ memory cells of $L$ bits, and has a time complexity (i.e. the number of additions and subtractions of $L$ bits numbers) of the order $L \cdot k$.

Note that the memory can be further reduced by only storing $M_{l-k+1} \ldots M_l$. In each step of the algorithm the next memory value ($M_{l+1}$ during precoding and $M_{l-k}$ in the postcoding algorithm) is then computed by relation 7.3. This reduction was not mentioned by Becker.

### 7.2.2 Median approximation

As was indicated at page 33, the elements of $\mathcal{P}_L$ correspond to the medians in the zero-error tree at depth $L$. Equation 1.59 is recapitulated here to show the relation between a median and a precoded sequence.

$$m(p) = \frac{1}{2} + \sum_{l=1}^{L} (2p_l - 1)\left(\frac{1}{2} - \frac{1}{2^l}\right)/(2q)^l$$ \tag{7.4}

(7.4)

Note that $q = 1 - p$ such that $(2p)(2q)^{k-1} = 1$, and that $q > 1/2$. A median is a real number in the unit interval. The idea is to represent the message $m$ also as a real number $\theta$ in the unit interval. When there are $M = M_L$ messages, as in the case of block coding, the number $\theta$ could be computed by $\theta_m = m/(M - 1)$ for $m \in \{0, 1, \ldots, M - 1\}$. In case of recursive coding, the message $m$ can be presented as an infinite binary stream, so the number $\theta$ could be the binary fraction corresponding with $m = m_1m_2\ldots$:

$$\theta_m = \sum_{n=1}^{\infty} m_n \cdot 2^{-n}. \tag{7.5}$$

From the real number $\theta_m$, the precoded sequence can be constructed by successively choosing the $p_l$ such that $\theta_m$ is approximated as closely as possible by the successive medians at depth $l$, $l = 1, 2, \ldots$. So e.g. if $\theta_m > 0.5$ then $p_1 = 1$, and if $\theta_m < 0.5$ then $p_1 = 0$. As mentioned by Schalkwijk [69, p. 286], when both choices for $p_l$ would result in an equally good approximation, then $p_l$ should be chosen such that $p_l = 1 - p_{l-1}$ in order to avoid forbidden subsequences.

For inverse precoding, the receiver computes the median that corresponds with the estimated precoded sequence, and looks for the closest $\theta_m$ to obtain the estimated message. The median approximation precoding scheme was used by both Schalkwijk [69] and Zigangirov [117].

A disadvantage of the precoding scheme based on median approximation is that machine accuracy is limited while the required accuracy to determine $p_l$ increases with $l$. On the other hand, the required memory in e.g. enumerative coding also increases with the length of the precoded sequences, but in practice it is in general easier to increase the memory than to extend the machine accuracy. Another problem is that the number of medians at depth
In the zero-error tree is slightly less than $M_z$ because some precoded sequences lead to the same median. See e.g. Figure 1.12, where 011 and 100 lead to the same median, as well as 0100 and 0011. However, these median paths will diverge later. On closer examination it turns out that the paths that lead to the same median are the ones that correspond with the sequences of type $p_1 \ldots p_{l-k} 01^{k-1}$ and $p_1 \ldots p_{l-k} 10^{k-1}$. The first path is necessarily followed by a 0, whereas the second path must be followed by a 1. All other precoded sequences of length $l$ can be followed by both 0 and 1. So by extending the median paths at depth $l$ with one symbol it follows that the number of medians at depth $l$ equals $M_{l+1}/2$.

To see that a precoding system based on median approximation works, i.e. that the postcoder really is the inverse of the precoder, the distances between medians at a certain depth are investigated. From Equation 7.4 follows that $m(01^{k-1}) = m(10^{k-1})$. Moreover, any sequence $p'$ that is obtained from another sequence $p$ by substituting a subsequence $01^{k-1}$ of $p$ by $10^{k-1}$ or vice versa, satisfies $m(p) = m(p')$. Therefore, any two consecutive medians at depth $L$ are generated by sequences $p$ and $p'$ such that $p_l = p'_l$ for $1 \leq l \leq L - x$, $p_{L-x+1} \ldots p_L = 01^{x-1}$, and $p'_{L-x+1} \ldots p_L' = 10^{x-1}$ for some $x$, $1 \leq x < k$. The distance between such two consecutive medians equals $(2q)^{-L}(1 - (2q)(2q)^{x-1})$. (Remember that $(2 - 2q)(2q)^{x-1} = 1$.) This distance is maximal for $x = 1$ and minimal for $x = k - 1$. Note that the distance between 0 and the lowest median equals $(2q)^{-L}/2$, as well as the distance between 1 and the highest median. When the number of messages $M$ is chosen such that the distance between $\theta_m$ and $\theta_{m+1}$ is at least $(2q)^{-L}(2q - 1)$, then one is assured that the mapping from $\theta$ to the set of medians at depth $L$ that are closest to $\theta_m$ and $\theta_{m+1}$ respectively are different. It is also clear that when $\hat{p} = p$, the postcoder will be the inverse of the precoder. But one is not assured that $\hat{m} = m$ when $\hat{p} \neq p$, because $\hat{p}$ might correspond to one of the medians that are not used during precoding.

On the other hand, this could be an extra error-correcting mechanism.

### 7.2.3 Regular grammar

Schalkwijk [69, 75] discovered that the precoded sequences can be generated by a regular grammar. The grammar for $k = 3$ is depicted as a state diagram in Figure 7.2. The generalization to other values of $k$ is obvious. The grammar determines the rules by which a binary sequence can be transformed to a precoded sequence. By starting in state $S$, and following the arrows corresponding to the next source bit, the current state is determined. Only when in states $Z^{(2)}$ or $N^{(3)}$, the next step is fixed and the corresponding production rule is executed before introducing the next source bit. E.g. the input sequence 10001 leads to

$$S \overset{1}{\rightarrow} 1N \overset{0}{\rightarrow} 10Z^{(1)} \overset{0}{\rightarrow} 100Z^{(2)} \rightarrow 1001N^{(1)} \overset{0}{\rightarrow} 10010Z^{(1)} \overset{1}{\rightarrow} 100101N^{(1)};$$

where the fourth step is forced.

In other words, the grammar inserts a 1 after each 00 in the source sequence, and a 0 after each 11 in the source sequence. Therefore, the produced sequences can not contain subsequences 0111 and 1000. The postcoding is done by repeatedly scanning the estimated precoded sequence from right to left and substituting each subsequence 1001 by 100 and each subsequence 0110 by 011. Postcoding can also be done by going through the state diagram and omitting all forced steps.

A disadvantage of the precoding system based on a grammar is that the length of a precoded sequence depends on the source sequence.
Figure 7.2: Grammar

When computing the asymptotic precoding rate, the only interesting states are \( N^{(1)} \) and \( Z^{(1)} \). Assume that all source bits are generated independently with equal probabilities for 0 and 1. So e.g. when going from state \( N^{(1)} \) to \( Z^{(1)} \), with probability 1/2 the source bit is 0 and one bit (0) is produced, and with probability 1/2 the source bit is 1 and two bits (10) are produced. Thus, the expected number of precoded bits per source bit is 3/2, and the asymptotic precoding rate is 2/3. As mentioned in Section 4.1, the optimal precoding rate is \(-1 + \log_2(1 + \sqrt{5})\) which is approximately 0.694, so there is an unnecessary precoding loss here.

To overcome this precoding loss, Schalkwijk [75] proposed a modification of the production rules which removed the symmetry of the precoding scheme. The modification is described as follows: when the source bit is a 0, the output bit is such that it differs from the former output bit, and when the source bit is a 1, the output bit is equal to the former output bit. Only for the first source bit there is no previous output bit available, so the first output bit is always equal to the first source bit (a formal solution would be to define output bit 0 as 1). The same input sequence \(10001\) as in the previous example would lead to

\[
S \xrightarrow{0} 1N \xrightarrow{0} 10Z^{(1)} \xrightarrow{0} 101N^{(1)} \xrightarrow{0} 1010Z^{(1)} \xrightarrow{0} 10100Z^{(2)} \rightarrow 101001N^{(1)}
\]

in the modified scheme, where the sixth step is forced. Postcoding is done similarly as in the original scheme. An important property of the modified scheme is that it is not symmetric anymore. In general, a source bit equal to 0 leads to 1 output bit whereas a source bit 1 produces 2 output bits. The asymmetry of the modified scheme can be explored by assuming a non-uniform information source. Let \( p \) be the probability of the information source generating a 1. Schalkwijk [75] computed the asymptotic precoding rate \( R_0 \) of the
modified scheme:

\[ R_0 = \frac{h(p)}{1 + p}. \]  

(7.6)

Note that for \( p = 1/2 \) the precoding rate is still 2/3. By optimizing Equation 7.6, the minimum information loss in precoding is obtained for \( p = (3 - \sqrt{5})/2 \), in which case we have \( R_0 = -1 + \log_2(1 + \sqrt{5}) \). Consequently, for \( p = (3 - \sqrt{5})/2 \) the maximal achievable precoding rate is obtained. However, we are not allowed to use symmetrical source sequences which reduces the practicality of the scheme.

### 7.2.4 Prefix-synchronized codes

A prefix-synchronized code is defined as a set of code words with the property that each code word has a known sequence as a prefix, followed by a coded data sequence in which this prefix is not allowed to occur. In a coding scheme word synchronization might be affected when errors have occurred in the stream of code words, but by using a prefix-synchronized code, word synchronization can be easily recovered at the decoder side by scanning the incoming stream of symbols for the occurrence of the forbidden prefix. Morita, van Wijngaarden and Vinck [59] constructed encoding and decoding algorithms for a class of prefix-synchronized codes.

The constrained sequences in \( \mathcal{P} \) have two forbidden subsequences namely 01\(^k\) and 10\(^k\), but by using a simple transformation this can be reduced to one. Let \( \mathcal{P} = \{p_1 \ldots p_L\} \) be an element of \( \mathcal{P} \). Define the sequence \( D(p) \) by

\[ D(p)_l = p_l - p_{l-1} \mod 2 \quad (1 \leq l \leq L), \]

(7.7)

where \( p_0 := 0 \). Note that subtraction is used in Equation 7.7 instead of addition to facilitate generalization to arbitrary alphabeth sizes. The transformation \( D \) is also known as DPSK (differential phase shift keying) [13, p. 112]. The inverse of \( D \) is obvious:

\[ D^{-1}(p)_l = D^{-1}(p)_{l-1} + p_l \mod 2 \quad (1 \leq l \leq L), \]

(7.8)

where \( D^{-1}(p)_0 \) is defined as 0, or in other words \( D^{-1}(p)_l = \sum_{i=1}^{L} p_i \mod 2 \). The set \( D(\mathcal{P}) \) is roughly described as the set of binary sequences not containing the subsequence 0\(^k\)-1.

Let \( \mathcal{F}_L \) be the set of binary sequences of length \( L \) not containing the subsequence 0\(^k\)-1. Then the set \( D^{-1}(\mathcal{F}_L) \) is contained in the set \( \mathcal{P}_L \). In fact, \( \mathcal{P}_L \) is only slightly larger than \( D^{-1}(\mathcal{F}_L) \). For the set \( \mathcal{F}_L \), the precoding algorithms of Morita, van Wijngaarden and Vinck [59], which are presented below, can be used. Together with the transformation \( D \) and its inverse, a precoding system for repetition feedback (block) coding is obtained that achieves an asymptotically optimal precoding rate.

<table>
<thead>
<tr>
<th>Precoding of ( \mathcal{F}_L )</th>
</tr>
</thead>
<tbody>
<tr>
<td>{0 ≤ m &lt;</td>
</tr>
<tr>
<td>( p := \text{PRE}(m, L) )</td>
</tr>
<tr>
<td>{( p = p_1 \ldots p_L ) is the ( m )th element of ( \mathcal{F}_L )}</td>
</tr>
</tbody>
</table>

FUNCTION \( \text{PRE}(x, l) = \)

\{0 ≤ x < |\( \mathcal{F}_l \)|\}

104
BEGIN
IF \( l \geq k \) THEN
BEGIN
\( i := 1; \ y := x; \)
WHILE \( y \geq G_{l-i} \) DO
BEGIN
\( y := y - G_{l-i}; \)
\( i := i + 1 \)
END;
IF \( i = k \) THEN RETURN \( 0^l \)
ELSE RETURN \( (0^{i-1} \text{PRE}(y, l-i)) \)
END
ELSE RETURN \( \text{BIN}(x) \)
{\text{PRE}(x, l) is the } x^{th} \text{ element of } F_l } 
END

\{p = p_1 \ldots p_L \in F_L\}
m := \text{POST}(p, L)
{m, \ 0 \leq m < |F_L|, \ is \ the \ number \ of \ p \ within \ the \ set \ F_L}\}

FUNCTION POST(y, l) =
{y = v_1 \ldots v_l \in F_l}
BEGIN
IF \( l \geq k \) THEN
BEGIN
IF there exists an \( i, 1 \leq i < k, \) such that \( y = 0^{i-1}1_{p} \) THEN
RETURN(\( \sum_{j=1}^{i-1} G_{l-i} + \text{POST}(y, l-i) \))
ELSE RETURN \( (G_l - 1) \)
END
ELSE RETURN \( \text{INV\text{BIN}}(y) \)
{\text{POST}(y, l) \ is \ the \ number \ of \ y \ within \ the \ set \ F_l } \}
END

Both algorithms are recursively defined. The function BIN is assumed to provide the binary representation of a non-negative integer, and INV\text{BIN} is its inverse. It is also assumed that the numbers \( G_l (1 \leq l \leq L) \) are precomputed and stored in memory. They are equal to the cardinality of \( F_l \) and can be computed recursively by
\[
G_l = \begin{cases} 
2^l, & \text{if } l < k, \\
2 \cdot G_{l-1} - G_{l-k}, & \text{if } l \geq k.
\end{cases}
\quad (7.9)
\]

For more details and proofs we refer to [59]. The time complexity is of the order \( L \), and the memory needed consists of \( L \) memory cells of \( L \) bits. The precoding system based on prefix-synchronized codes is similar to the enumerative precoding scheme. The memory needed is equal, but it has a lower time complexity than the enumerative precoding scheme.

The same remark as at the end of subsection 7.2.1 applies here too. Namely, the memory size can be further reduced by only storing the numbers \( G_{l-k+1} \ldots G_l \). The next number (i.e. \( G_{l-k} \) for both precoder and postcoder) is during each step computed by using relation 7.9 backwards. This reduction was not mentioned by the authors of [59].
7.2.5 Dummy reversal

Another precoding idea is derived by Schalkwijk [70]. Assume that the messages are binary sequences. In a block coding scheme the length of the sequences would be fixed and finite, but in a recursive scheme the sequences have infinite length. The idea is to add one dummy reversal after each $k - 2$ information bits. E.g. when $k = 3$ and the input sequence is 10001, the output sequence would be 1001010110, where the dummy bits have been made boldface. Although the precoder and postcoder are very simple, the precoding rate is only $(k - 2)/(k - 1)$. On the other hand, the loss in precoding rate decreases when $k$ grows.

The rate of a complete transmission scheme equals $R = (1 - kp) \cdot R_p$, where $R_p$ is the precoding rate. For an optimal precoding rate, the rate $R$ is known to hit the capacity curve for each value of $k$ (see e.g. subsection 4.4.1 of this thesis). For the dummy reversal precoding scheme, where $R_p = (k - 2)/(k - 1)$, the rate lines are tangent to the elliptic curve

$$R = 1 - 2\sqrt{p(1-p)},$$

(7.10)

as was mentioned by Schalkwijk [76]. Both the capacity curve of the binary symmetric channel and the suboptimal elliptic curve are depicted in Figure 7.3.

The precoded sequences that are produced by dummy reversal precoding have the interesting property that flip sequences, i.e. sequences like $10^{k-1}10^{k-1} \ldots 10^{k-1}$ (see also subsection 3.2.2 and Section A.2 of this thesis), do not occur. Therefore, Conjecture 3.1 holds for all possible channel (error) probabilities, when dummy reversal precoding is used. This was also mentioned by Schalkwijk in Chapter IV of [76].

7.2.6 Sliding block code

A class of constrained codes that is widely used in magnetic recording is the class of runlength-limited (RLL) sequences, see e.g. the book by Immink [46]. These RLL sequences are very similar to the constrained sequences in $\mathcal{P}$. RLL-sequences are characterized by two
parameters, $d_{RL}$ and $k_{RL}$, which stipulate the minimum and maximum runlength, respectively, that may occur in the sequence. A run is a number of consecutive zeros between two ones. In other words, each subsequence of the form $10^x1$ must satisfy $d_{RL} \leq x \leq k_{RL}$. As mentioned in subsection 7.2.4, the set $P$ can be transformed to the set of binary sequences not containing the subsequence $0^{k-1}$. This set can be slightly decreased to obtain the set of RLL sequences with parameters $d_{RL} = 0$ and $k_{RL} = k - 2$. Various (practical) coding schemes have been developed for RLL sequences [46]. One of the most important coding schemes is the sliding block coding algorithm by Adler, Coppersmith, and Hassner [1]. The algorithm also provides a way to limit the error propagation which is important for recursive coding. An excellent survey paper on sliding block codes is by Marcus, Siegel, and Wolf [58].

In principle, every rational rate not exceeding the maximal precoding rate can be achieved by sliding block coding schemes. By a systematic procedure such a code can actually be constructed. For a detailed description of the procedure see [1]. A sliding block code that generates RLL sequences with $d_{RL} = 0$ and $k_{RL} = 1$, is presented in Figure 7.4 as a finite-state machine (see Siegel [89]). In each encoding step, three bits are produced by two input bits. The decoder has a sliding window of six bits. The first three bits form the current block, and the final three bits (look-ahead block) are used to determine the state in the finite-state machine. The sliding window shifts along the codestring one block (of three bits) at a time, producing at each shift a decoded data word, depending only on the window contents. The error propagation is therefore limited to four data bits (or six code bits), since an erroneous code word can only affect the decoding decisions while it is contained in the sliding window. This sliding block code can be used as a building block for our $k = 3$ precoder. The precoding rate is $2/3$, as was also the case for the regular grammar precoder (see subsection 7.2.3). The advantage of the sliding block precoder is that the length of the output sequence does not depend on the input sequence. Furthermore, the sliding block precoder is able to limit the error propagation.

Note that for coding for asymmetric channels it could be useful to fix the symbol precoding distribution, which requires a generalization of sliding block codes, as was done by Khayrallah and Neuhoff [50].
7.3 Practical applications

Essentially, each situation that can be modeled as a discrete memoryless channel with a noiseless feedback link is suitable for multiple-repetition feedback strategies. In this Chapter it is shown that various practical applications can be viewed as an attempt to send information over a channel with a noiseless feedback link. It will become clear that it is not necessarily to actually transmit information.

In each Section a particular application is described, some applications being more realistic than others. In some Sections it is clear that the forward channel is a DMC, and in others the forward channel could be a DMC but does not necessarily has to be so. All constructive strategies for DMCF that are known (see e.g. Chapter 1) can be used, in particular multiple-repetition feedback strategies, and consequently also all results described in this thesis.

7.3.1 Semi-conductor memories

Consider a ROM (Read Only Memory) that consists of semi-conductors. The ROM can be programmed once and is then used in an electronic device. Sometimes it happens that one of the semi-conductors of a manufactured ROM is defective in which case the complete ROM is useless. People in industrial life are thinking about using an error correcting mechanism for ROM's such that partially defect ROM's can still be used. The purpose is to lower the failure rate and eventually increase the yield.

A semi-conductor can be seen as a memory element that initially contains a zero. The contents can be changed by writing a one. After the memory element is programmed, a defect can be detected by reading the contents of the element. If we assume that each memory element has a fixed probability of being defective, the problem of programming such memories is essentially the same as transmitting information over a Z-channel (see Figure 1.5) with a noiseless feedback link. In other words, a repetition strategy can be used to "correct" all defective elements. Consequently, all results derived in this thesis can be used, including subsection 4.4.2. Note that it might be possible that a memory element initially contains a one instead of a zero. In that case not only 1 → 0 errors, but also 0 → 1 errors might occur, which essentially means that the considered channel is not a Z-channel, but an (asymmetric) binary channel and the repetition strategy has to be adjusted accordingly.

When the position of the defective memory elements is known in advance, and sound memory cells can not be broken during the programming of the ROM, another technique invented by Schalkwijk [77] can be used, as is explained in subsection 7.3.4.

7.3.2 Economic markets

This Section is a based on a paper by O'Neill [61]. By considering an established aggregate economic model and several of its variants as information channels, O'Neill shows that Shannon's coding Theorem allows one to precisely define the sense in which information is transmitted by an economic market. Unique to economic markets is the structural existence of a noiseless feedback path from the channel output, the market clearing price, to the channel input, market trader bids and offers which simultaneously account for source information and market clearing prices. O'Neill shows that economic equilibrium occurs
when market agents can predict the market input information from price observations with arbitrarily small error probability.

The market Equations

\[
D(t) = -\beta p(t) + D_0 \quad \text{(demand),} \\
S(t) = \gamma p_e(t) + S_0 + u(t) \quad \text{(supply),} \\
D(t) = S(t) \quad \text{(equilibrium),}
\]

are based on the work of Muth [60]. Here \( p(t) \) is the market price in period \( t \), \( p_e(t) \) is the market price expected in the \( t \)th period conditional on available information through period \( t-1 \), and \( u(t) \) is a random error term that is independently and identically distributed. The market Equations yield the price model

\[
p(t) = -\frac{1}{\beta} [\gamma p_e(t) + S_0 - D_0 + u(t)],
\]

with a static equilibrium price

\[
\bar{p} = \frac{D_0 - S_0}{\gamma + \beta}.
\]

A feedback strategy would give a mechanism by which the agents, using only knowledge of the market model and past prices, could achieve 7.15.

The price expectation \( \bar{p} \) held by agents gives rise to a sequence of market clearing prices \( p(1), p(2), \ldots, p(N) \). Equilibrium prevails if there exists a sequence of estimates \( p_e(1), p_e(2), \ldots, p_e(N) \), where \( p_e(t) \) (\( 1 \leq t \leq N \)) depends only on \( p(1), p(2), \ldots, p(t-1) \), such that for all \( \bar{p} \)

\[
\lim_{N \to \infty} \Pr\{p_e(N) \neq \bar{p}\} = 0.
\]

The Muth market is called reliable if Equation 7.16 is true for all possible price expectations \( \bar{p} \).

The relation of O’Neill’s theory to coding for channels with noiseless feedback is illustrated. The price model in Equation 7.14 can be seen as the channel model. The input of the channel is \( p_e(t) \), the noise is modeled by \( u(t) \), and the channel output is \( p(t) \). The transmitter is a market agent, who has a message \( \bar{p} \). The market itself acts as the receiver, who would like to get to equilibrium, or in other words the market wants to “know” the message \( \bar{p} \), the equilibrium price.

O’Neill showed how the result of Schalkwijk and Kailath [78], which describes a coding scheme for a continuous channel with feedback, can be used to achieve equilibrium in economic markets. A closer connection with discrete feedback strategies can be made by considering Horstein’s model which was described in Section 1.3 of this thesis. For this purpose, the market channel is assumed to behave like a BSC. The equilibrium price \( \bar{p} \) is comparable with Horstein’s message point. The expected market price \( p_e(t) \) relates to Horstein’s midpoint. The only difference is that not \( p_e(t) \) is actually transmitted, but a binary symbol indicating the position of \( p_e(t) \) with respect to \( \bar{p} \). More precisely, 0 is transmitted when \( \bar{p} < p_e(t) \) and in case \( \bar{p} > p_e(t) \), a 1 is sent over the channel. In terms of economic markets, an agent sends a 0 when the expected market price is too high, and a 1 when the expected market price is too low. Horstein [45] showed that the midpoint
eventually will approximate the message point arbitrarily close, which in terms of economic markets means that the market is reliable.

It is concluded that constructive strategies for channels with noiseless feedback can be used as a mechanism for agents operating on an economic market in order to achieve market equilibrium.

7.3.3 Channels with delayed feedback

So far we only considered DMC's with a noiseless and delayless feedback link. However, it is possible that in practice the $n^{th}$ output symbol is not available to the transmitter before the $(n+1)^{th}$ transmission takes place, but only at the $(n+\Delta)^{th}$ transmission. A possible solution for this problem is interleaving. It means that in stead of one message, $\Delta$ messages are transmitted in parallel. Message $i$ $(1 \leq i \leq \Delta)$ is transmitted at time instants $i + (n-1)\Delta$ for $n = 1, 2, \ldots$. Each message can be transmitted by means of an ordinary feedback strategy since the $n^{th}$ output symbol is always available to the transmitter at the $(n+1)^{th}$ transmission. If for example block strategies with block length $N$ are used, then the total number of transmissions for $\Delta$ messages is $\Delta \cdot N$ which would also be the case when $\Delta$ messages were transmitted without interleaving, so interleaving results in the same overall transmission rate. A disadvantage of interleaving is the increased amount of memory that is needed for both the transmitter and the receiver.

The capacity of a DMC is not altered by the availability of a noiseless feedback link, not even when there is a delay in the feedback link. Note that the capacity of a channel with memory can indeed be increased by the availability of a noiseless delayed feedback link [113].

Finally, the problem of delay in the forward channel (or a combination of delay in the forward and the feedback channel) can similarly be solved by interleaving.

7.3.4 Memories with known defects

In subsection 7.3.1 defective memories are considered. The position of the defects is revealed during the writing process, i.e. after a bit is stored, the cell is re-read to check whether the contents agrees with the stored value. If we assume that both $0 \rightarrow 1$ and $1 \rightarrow 0$ writing errors can occur with probability $p/2$, the maximal amount of information that can be stored in one memory cell is $1 - h(p/2)$ (the capacity of the BSC with error probability $p/2$). Note that we always assume that the decoder, i.e. the reader of the memory, has no information about the defects.

However, when the position of the defective memory cells is known in advance (e.g. when all memory cells are tested before using the memory device), this information can be used to increase the amount of information that can be stored in one memory cell. Since the fraction of good memory space is $1 - p$, the maximal amount of information that could be stored in one memory cell is $1 - p$ bit (when the decoder also knows the position of the defective cells). Kusnetsov and Tsybakov [55] showed by means of an existence proof, that $1 - p$ is (asymptotically) achievable even when the decoder is not aware of the position of the defects. The paper by Kusnetsov and Tsybakov initiated the search for good codes for defective memories. The difference $(1 - p) - (1 - h(p/2))$ can be explained as being necessary to inform the reader about the location of the defects.

One memory cell can be modeled as a binary defect channel (BDC), as depicted in Figure 7.5. When both encoder and decoder are not aware of the state of the channel, it is
equivalent with a BSC with error probability \( p/2 \). For example, the probability of receiving a 1 when transmitting a 1 equals \( (p/2) \cdot 0 + (1-p) \cdot 1 + (p/2) \cdot 1 = 1 - p/2 \). So in that case the capacity equals \( 1 - h(p/2) \). When both encoder and decoder know the state of the channel, the capacity is the weighted sum of the capacities of the three individual channels, which is \((p/2) \cdot 0 + (1-p) \cdot 1 + (p/2) \cdot 0 = 1 - p\). When the state of the channel is only known to the encoder, and not to the decoder, it is called a channel with side state information. When the channel state is known to the encoder, the encoder is able to determine what value will be stored in the memory cell after a bit has been written, or in other words a noiseless feedback link is available.

By using Shannon’s paper on channels with side state information [83], Schalkwijk [77] showed that the BDC with side information is equivalent to the discrete memoryless channel \( BDC \) (without side information), as depicted in Figure 7.6, in the sense that it has the same capacity and that a coding scheme for \( BDC \) can be translated into a coding scheme for the BDC with the same error probability. The channel probabilities of \( BDC \) are equal to \( p/2 \) for the dashed lines in Figure 7.6, and \( 1 - p/2 \) otherwise. An input \( z = (x_1, x_2, x_3) \) in the channel \( BDC \) is comparable with three inputs for the individual channels of the BDC: \( x_1 \) is put into the 0-defect channel, \( x_2 \) into the error-free channel, and \( x_3 \) into the 1-defect channel. Note that the probability of receiving \( y \) given input \( z \) in \( BDC \) equals \( p/2 \) exactly when \( y \neq x_2 \). Although \( BDC \) has eight inputs, Shannon [88] showed that capacity can be achieved by considering only two inputs. From Figure 7.6 follows that any two inputs with different \( x_2 \) will satisfy. The channel \( BDC \) then is equivalent to the BSC with error probability \( p/2 \), as expected. To see how a coding scheme for \( BDC \) can be translated into a coding scheme for BDC with side state information, let \( z = (x_1, x_2, x_3) \) be the input of
Figure 7.6: Binary defect channel without side information (BDC)
the channel $BDC$, then the input of the channel BDC (the value that is written into the memory cell) will be $x_1$ when the memory cell has a 0-defect, $x_2$ in case the memory cell is error-free, and finally $x_3$ when the memory cell is known to have a 1-defect. The decoding is the same for both schemes.

The reason that the capacity of $BDC$ is only $1 - h(p/2)$, and not $1 - p$, is that we considered only one memory cell. To obtain (more) gain from the side state information, we have to consider more memory cells, or in the words of Schalkwijk [77]: we have to anticipate into the future. Suppose we consider $n$ memory cells. The finite state channel that describes the $n$ cells is denoted by $BDC^n$. The equivalent DMC as mentioned by Shannon in [83] is denoted by $BDC^n$. Since each memory cell can be in 3 different states, the channel $BDC^n$ consists of $3^n$ states. Each individual channel of $BDC^n$ has $2^n$ inputs and $2^n$ outputs. The derived channel $BDC^n$ therefore has $(2^n)^3$ inputs and $2^n$ outputs. Again, $2^n$ inputs are sufficient to achieve capacity. The capacities $C_n(p)$ of $BDC^n$ are increasing and approach $1 - p$. The capacity $C_2(p)$ was derived by Schalkwijk [77].

$$C_2(p) = 1 - \frac{1}{2} \left( \frac{1}{2} p^2 + h(1 - \frac{1}{2} p^2) + (1 - \frac{1}{2} p^2) h \left( (1 - \frac{1}{2} p)^2 / (1 - \frac{1}{2} p^2) \right) \right)$$ \tag{7.17}

Multiple-repetition feedback strategies can be used to achieve capacity on $BDC^n$. By translating the strategies to coding schemes for $n$ memory cells, the capacity $1 - p$ can be approached in a constructive way. In subsection 4.4.3 of this thesis a multiple-repetition strategy is shown that nearly achieves capacity on the channel $BDC^2$. The capacity $C_2(p)$ is also depicted in Figure 4.4. When $n$ increases it is difficult to find the optimal $2^n$ inputs of $BDC^n$ that yield capacity, and consequently also the capacity $C_n(p)$. However, an upper bound on $C_n(p)$ was obtained by Janssen [47, pp. 40-51]. Because of the exponential increase of the size of $BDC^n$, the practical use of the derived memory coding schemes for large $n$ is limited.

### 7.3.5 Duplex channels

Schalkwijk [70, 76] proposes a coding scheme for duplex channels, i.e. a situation where two noisy channels are available in opposite directions. One would like to use multiple-repetition feedback strategies in stead of one-way error correcting codes to decrease the complexity of the coding scheme, and to increase the transmission rate, but a noiseless feedback channel is not available. However, Schalkwijk demonstrated that it is possible to use one of the noisy channels as a feedback link for the other one. Schalkwijk’s idea is depicted in Figure 7.7. The input of the channel is denoted by $X$, the forward noise is $N$, and the output is $Y$. The noise in the feedback channel is denoted by $N_f$, and the received feedback is $F$. All these variables are assumed to be binary. The forward and feedback channel are assumed to be BSC’s with error probability $p$, so $Pr\{N = 1\} = Pr\{N_f = 1\} = p$. In Figure 7.7 also are depicted two identical (linear) convolutional encoders (CE). Convolutional coding was briefly described in Chapter 1 of this thesis. For an example of a convolutional encoder see Figure 1.2. From Figure 7.7 follows that $Y = X \oplus N$ (modulo two addition), and

$$F = CE(Y) \oplus N_f \oplus CE(X) \tag{7.18}$$

$$= CE(Y \oplus X) \oplus N_f \tag{7.19}$$

$$= CE(N) \oplus N_f \tag{7.20}$$
The received feedback signal $F$ can therefore be seen as the encoded "signal" $N$ together with some noise $N_f$. Consequently, by decoding the convolutional code, the transmitter can obtain the forward noise $N$ with high reliability. However, the error probability of the convolutional code can only be made sufficiently small by increasing the decoding delay $D$ so that the forward noise at transmission $n$ is only known to the transmitter at time instant $n + D$. In subsection 7.3.3 was illustrated how the problem of delayed feedback can be handled by interleaving $D$ feedback strategies.

Schalkwijk [70] simulated the coding scheme described above with a rate 1/2 systematic convolutional code such that $CE(a_n, a_{n+1}, a_{n+2}, a_{n+3}, a_{n+4}, a_{n+5}, a_{n+6}, a_{n+7}) = a_n \oplus a_{n+1} \oplus a_{n+2} \oplus a_{n+4} \oplus a_{n+5} \oplus a_{n+7}$. The convolutional code was decoded using the Viterbi algorithm [106]. For a more efficient implementation of the convolutional decoder, syndrome decoding by Schalkwijk and Vinck [74] can be used. The block error probability of the complete scheme was essentially equal to the bit error probability of the convolutional code. But the forward transmission rate was approximately twice the rate of the convolutional code.

An alternative for the described duplex scheme would be to use both channels for forward transmission using the rate 1/2 systematic convolutional code. This would yield the same performance in terms of transmission rate and reliability. But the main advantage of Schalkwijk's approach is that the decoder has a considerably reduced complexity. The hardware or the program complexity has been "transferred" from the passive (receiving) side to the active (transmitting) side of the duplex channel. This coding strategy can be used to great advantage in a computer network with a star configuration. For the information flow from the central computer to the satellites one uses the duplex strategy thus only requiring one complex one-way decoder at the central facility. For the information flow from a satellite computer towards the central facility one uses one-way coding, again using the complex one-way decoder at the central computer. One thus saves a number of complex one-way decoders equal to the number of satellite computers in the multiple dialog system.
7.3.6 Statistical experiments

In several practical situations it is desired to measure the value of a parameter by a number of tests. One should think of fields like chemical experiments, medical examinations, and psychological tests. An overview of the theory of sequential statistical experiments is presented by Chernoff in [21]. Each test can be considered as a question that relates to the value of the parameter. The test result then is an answer to that question. Due to human error or other nonsystematic influences, some of the test results will be false. The goal is to obtain a reliable estimate of the parameter by a small number of tests. The performance will increase when the next test (or question) is allowed to depend on the outcomes of previous tests. From a theoretical viewpoint this problem is equivalent to Ulam's problem as described in subsection 1.2.4 of this thesis. Ulam’s problem is known to be equivalent to the problem of coding for channels with noiseless feedback, see e.g. Chapter nine of Ahlswede and Wegener [4].

Burnašev and Zigangirov studied a specific estimation problem. A device is assumed to “know” a parameter \( \theta \in [0, 1] \). Questions can be asked about \( \theta \) and the device outputs a “yes/no” answer, but it has an error probability \( p \). In their first paper [18] the questions are determined by a measurement point belonging to \([0, 1]\), and the device indicates whether \( \theta \) is to the left or right of the measurement point. The estimation method is based on ideas of Horstein [45] and Zigangirov [114, 115]. The reliability for a fixed number \( n \) of observations satisfies

\[
E_{bz}(R) = \log \left( \frac{2}{(1 - p + \sqrt{p})^2} \right) - R, \quad \text{if } R \leq R_{\text{crit}},
\]

\[
D(\rho || p), \quad \text{if } R = D(\rho || p) > R_{\text{crit}},
\]

where \( R_{\text{crit}} \) is the critical rate of the BSC. Note that \( E_{bz}(R) \) equals Elias’ random coding bound (see Theorem 1.4 of this thesis). The total number of operations that are needed for \( n \) observations is of the order \( n^4 \), and the amount of memory is on the order of \( n^2 \). When the rate approaches capacity however, the required computational accuracy increases, and the complexity grows. Burnašev and Zigangirov [18] also consider an observation scheme where the number of observations is variable such that a fixed error probability is guaranteed.

In their second paper [19], Burnašev and Zigangirov consider a device such that questions can be asked that are characterized by a subinterval of \([0, 1]\). The device indicates whether the parameter \( \theta \) belongs to the subinterval or not, again with an error probability of \( p \). Since the class of allowed estimation methods increases, it is clear that the performance of such observation schemes can increase. Burnašev and Zigangirov propose a method that is very similar to Zigangirov’s block coding scheme [115]. The performance is such that

\[
E_{bz}(R) = \max_{\lambda > 0, 0 < \alpha < 1/2} \left[ -\lambda \cdot R - \log \max \left\{ \frac{1}{2} \left( f(1/3, \alpha, \lambda) + f(-1/3, \alpha, \lambda) \right), f(1, \alpha, \lambda) \right\} \right].
\]

The function \( f(z, \alpha, \lambda) \) was previously defined in Theorem 1.14 of this thesis. The exponent \( E_{bz}(R) \) is smaller than Zigangirov’s exponent \( E_z(R) \) of Theorem 1.14. This difference is explained by the type of questions that is allowed to be put in the device. When the subinterval of \([0, 1]\) is allowed to consist of, say, \( k \) segments, (i.e. the device indicates whether \( \theta \) belongs to the set \([x_1, y_1] \cup [x_2, y_2] \cup \ldots \cup [x_k, y_k] \) or not) then the \( 1/3 \) in Equation 7.23 is replaced by \( 1/(2k + 1) \). So when \( k \to \infty \), the reliability of Zigangirov’s block coding scheme [115] is obtained. Burnašev and Zigangirov [19] also consider a variable length observation
scheme that achieves a reliability of \((1 - R/C) \cdot D(p\|1 - p)\), which is shown to be optimal under certain conditions by Burnašev [16].

Ahlswede and Wegener [4, p. 186] mention that the result of Burnašev and Zigangirov can be interpreted as a result on the stochastic approximation for the following situation. Let the regression function \(M\) increase strictly monotonically, and let \(\theta\) be the root of \(M(\theta) = 0\). A measurement at a point \(x\) determines whether \(M(x) > 0\) or \(M(x) < 0\). The result of the measurement is false with probability \(p\). We would like to find the root with \(\varepsilon\)-precision as fast as possible.

### 7.3.7 Other applications

There are several situations in practice where information can be send in two directions, not only from A to B, but also from B to A. In order to maximize the amount of information that can be send from A to B, one could use the channel from B to A as a feedback channel, especially when there is no information to be send from B to A. The feedback information then can also be used to reduce the complexity of encoder A as well as the complexity of decoder B. However, several practical limitations prevent the use of such a feedback channel. First, the feeding back of the received information costs bandwidth and transmitter power which is not always available. Second, for our purposes the feedback channel has to be noiseless which mostly is not the case. Only when transmitter B is very powerful compared to transmitter A, the feedback link from B to A can be considered being noiseless. An error correcting code can be used to correct the few errors in the feedback link (or a convolutional code as in subsection 7.3.5). Third, instantaneous feedback is required: the \(n^{th}\) feedback symbol has to be available before the \((n + 1)^{th}\) information symbol is transmitted. When there is a delay in the feedback channel, the interleaving technique as described in subsection 7.3.3 could be useful.

In the following subsections some practical situations are roughly described where information can be send both from A to B and from B to A, and where transmitter B is very powerful compared to transmitter A. Results on feedback strategies for channels with noiseless feedback might be useful in these situations.

### ADSL

Asymmetric Digital Subscriber Lines (ADSL) [66] are used to deliver high-rate digital data over existing ordinary phone-lines. ADSL facilitates the simultaneous use of normal telephone services, ISDN, and high speed data transmission, e.g., video. ADSL can be seen as the transition from existing copper-lines to the future fiber-cables. This makes ADSL economically interesting for the local telephone companies. They can offer customers high speed data services even before switching to fiber-optics.

ADSL is a new technology used to transmit information from the central office of a telecommunications network to a customer. The link from the customer to the central office is called the upstream link, and the link from the central office to the customer is called the downstream link. The capacity of the downstream link is very large compared to the upstream capacity (in the order of Mbit/s versus kbit/s) in order to facilitate e.g. internet surfing and video on demand services by the customer. Up to now the error correction is based on Reed-Solomon coding, but when the downstream link would be used as a feedback link, the transmission rate in the upward direction could be increased and the customer's encoder could be simplified. The use of a downstream feedback channel is especially suitable.
for customer applications such as one-way data transmission, in which a high upstream rate is needed, but little information has to be send from the central office to the customer.

**Satellite communication**

In satellite communication (see e.g. [37]), information is sent between the satellite and the ground station. The signalling power of the satellite is small compared to the signalling power of the ground station. Low complexity transmission systems are needed in the satellite. Therefore, the link from the ground station to the satellite (uplink) can be used as a (noiseless) feedback link for the noisy channel from the satellite to the ground station (downlink).

Some existing satellite communication systems use the uplink as a decision feedback line during transmission over the downlink. However, due to the large delays encountered at satellite communication, it is inefficient to wait for an acknowledgement of the previous block before sending the next block. Therefore, the transmitter sends block after block without waiting for acknowledgement. Then, after receiving an indication that an earlier block was received in error, the transmitter reinserts a copy of that block into the transmitted sequence. Upon receiving that block correctly, the receiver recognizes the proper location of that block in the message sequence and places it accordingly into the message. This procedure requires the receiver to buffer the received message for a time at least equal to the round-trip propagation delay.

**Army scouts**

In time of war it happens that an area has to be scouted by an army. For this purpose, scouts are sent into the field. Their goal is to transmit information about the area and the enemy to a central base station. Since the enemy is not entitled to know the geographical position of the scouts, and the scouts should be mobile, the transmitters of the scouts are very weak. The central base station on the other hand, whose position is known to the enemy, is equipped with a powerful transmitter. This situation is suitable for feedback communication, since the base station is able to provide (nearly) noiseless feedback to the scouts.

### 7.4 Conclusions

Several applications of multiple-repetition feedback coding were described. The applications go beyond transmission of information over a noisy channel. Furthermore, a number of precoding systems were mentioned which are important when implementing multiple-repetition strategies. For each particular application the most suitable precoding system can be chosen. For two precoding systems a possible reduction in memory complexity was indicated.
Chapter 8

Literature and open problems

Most of what is known about multiple-repetition feedback coding is described in this thesis. An overview of the literature concerning multiple-repetition feedback coding, among which the new results described in this thesis, is presented in this Chapter. Although much research has been done on the area of multiple-repetition feedback coding, several possibilities exist for further research. Therefore, some (yet) unsolved problems are mentioned also.

8.1 Literature

Multiple-repetition strategies were derived from Horstein’s scheme [45, 44]. Schalkwijk [69] developed the first block coding scheme for the BSC. Becker [7] generalized Schalkwijk’s idea to DMC’s with equal number of input and output symbols. Schalkwijk and Post [72, 73] computed error probabilities for binary recursive strategies. These leading papers on multiple-repetition feedback coding are described in the first Chapter of this thesis.

We continue with the contributions of the author of this thesis. Becker’s class of DMC’s for which multiple-repetition strategies can be used, is extended in the sense that extra erasure output symbols are allowed. In Chapter 2 multiple-repetition block coding strategies are considered. A new tail construction [104] is presented which broadens the formerly known class of allowable tail constructions.

Recursive multiple-repetition coding strategies are considered in Chapter 3. Existing methods for the BSC are generalized to DMC’s with arbitrary channel probabilities. The exponent that is denoted by $E_1$ in Equation 3.10 of this thesis, is computed in [102]. In [102] a Conjecture is posed that is equivalent with $E_1 \leq E_2$. In Figure 3.7 can be seen that this Conjecture is false. However, when the repetition parameters are suitably chosen, it is believed that $E_1$ determines the true asymptotic decoding error behaviour of the code word estimator. Also, when a small loss of rate is acceptable, extra precoding (see e.g. subsection 7.2.5) can make $E_2 = \infty$, in which case the Conjecture holds automatically.

In Chapter 4 the transmission rate of multiple-repetition strategies is computed. It is shown [99, 105] that the transmission rate equals the channel capacity for certain channel error probabilities. This result is based on Theorem 4.1, which is generalized from $yz^{b_x z}$-constrained sequences to sequences with arbitrary constraints in [101]. This generalization actually extends a Theorem by Shannon (Theorem 5.6 in this thesis) from integer to real durations. In [100] is shown how the precoding distribution of a capacity achieving multiple-repetition strategy can be computed.
It is shown [103] in Chapter 5 that for a given DMC with arbitrary channel probabilities, the repetition parameters should be chosen close to the (absolute value of the) logarithm of the channel error probabilities to maximize the transmission rate.

A modification of multiple-repetition block coding strategies is shown [98, 97] in Chapter 6 to be suitable for high channel error probabilities.

In Chapter 7 some additional references are mentioned concerning possible precoding systems of multiple-repetition coding schemes, and some applications.

8.2. Open problems

Although much is known about multiple-repetition feedback strategies, several problems are still open. The open problems offer interesting opportunities for further research. For that purpose some (yet) unsolved questions are mentioned.

Tail completeness. In subsection 2.2.1 four different tail conditions are presented. Whenever a tail is used that satisfies any of these conditions, the error-correcting capability determined by Equation 2.2 is satisfied. It is not clear that whenever a tail is used such that the required error-correcting capability is guaranteed, this tail necessarily has to satisfy (at least) one of the four tail conditions. A candidate that comes to mind is the tail $012012\ldots$, where the repetition parameters are $k_{10} = k_{21} = k_{02} = 3$ and $k_{12} = k_{20} = k_{01} = 2$. One might be able to obtain a number of tail conditions such that the required error-correcting capability is guaranteed if and only if (at least) one of those tail conditions is satisfied. Since a lot of variations can be thought of when constructing tails, it might be difficult to cover all possible tails. Therefore, a different formulation of the tail problem would present a more realistic goal: whenever it is possible to construct a tail that guarantees the required error-correcting capability, it is possible to construct a tail that satisfies (at least) one of the available tail conditions.

Error exponent for fixed-delay coding. Two different types of decoding errors are considered in subsection 3.2.2. The first type of decoding error is caused by an unfinished error correction process, and the second type is due to a special kind of information sequences called flipsequences. The first type of decoding error is conjectured to determine the true decoding error probability. Although a period of extremely many channel errors can affect earlier correctly received symbols during decoding, it is believed that the most common error events consist of a flipsequence followed by an unfinished error correction process. Therefore, the error exponent is assumed to equal the minimum of $E_1$ and $E_2$. Since the exponent $E_1$ is smaller than $E_2$ when the repetition parameters are suitably chosen (in that case $E_2 \approx 1$), the decoding error behavior is determined by $E_1$. It would be nice to prove the assertions mentioned above, perhaps by using the theory of typical sequences.

Left-to-right recursive coding. Recursive coding with fixed-delay is described in Section 3.2. The estimation algorithm scans the received sequence $r_n \ldots r_{n+D}$ from right to left to obtain an estimate for the transmitted symbol $t_n$. For binary channels the estimation can also be done from left to right by using a state diagram. A left to right estimation algorithm provides a generalization to variable-delay coding schemes, and facilitates the computation of the error exponent. However, for channels with more than two inputs, a left-to-right estimation algorithm is not (yet) available.
**Error propagation.** As mentioned in Section 3.3, recursive coding for multiple-repetition schemes suffers from potential error propagation during postcoding, depending on the used (inverse) precoding system. In Section 7.2 several precoding systems are considered that are able to limit the error propagation.

Another related problem is that the inverse encoder sometimes is able to detect a decoding error made by the code word estimator. Namely, when the estimated transmitted sequence is a sequence that could not have been transmitted by a multiple-repetition encoder given the received sequence, it is clear that the code word estimator made a mistake. It could be difficult to recover the synchronization when such a decoding error is detected (and even more difficult when the decoding error is not detected, although this is very unlikely, and probably even impossible for larger delay values).

Since the transmitter is aware of the received sequence, the transmitter is able to detect decoding problems at the receiver's side. The transmitter could therefore be able to prevent or correct such kind of decoding problems, although further research is needed to obtain good solutions.

**Achieving capacity for all channel probabilities.** In Section 4.2 is illustrated that, given the repetition parameters, one DMC exists such that channel capacity can be achieved by the multiple-repetition strategy. Since the repetition parameters are necessarily integers, and the channel error probabilities are real valued numbers, this implies that (many) DMC's exist such that a multiple-repetition strategy can not achieve channel capacity. The simplest example is repetition coding for the BSC as depicted in Figure 4.1. A step in the proper direction can be made when we are allowed to use rational repetition numbers, but it is not clear how such a coding scheme would look like. Going back to Horstein’s scheme [45], and Schalkwijk’s idea of regular median paths (see subsection 1.4.1), it seems promising to consider the generalization of 1-up, 2-down median paths to 2-up, 3-down median paths (and further). The channel error probability \( p \) that makes the median come back after going 2 steps up and 3 steps down, is the solution of

\[
(2p)^2(2q)^3 = 1, \tag{8.1}
\]

where \( q = 1 - p \), which is approximately \( p = 0.3053046 \). The corresponding regular median paths are depicted in Figure 8.1.

**Soft decision.** In this thesis multiple-repetition strategies are generalized such that they can be used on DMC's that have more outputs than inputs. The generalization is done by considering each received output symbol \( y \in Y \) as an erasure when \( y \notin X \) (see subsection 1.4.4). One could imagine a DMC like in Figure 1.15, where hard decision decoding is not appropriate. It is assumed in Figure 1.15 that \( p_0 > p_1 > p_2 > p_3 \). A channel like in Figure 1.15 arises when the output of a continuous channel is quantized in four levels, although more levels are similarly modeled. So when for example the output symbol \( \frac{1}{4} \) is received, it is more likely that a 0 was transmitted than a 1. When the output symbol \( \frac{1}{3} \) is considered as an erasure, information is unnecessarily thrown away. An intermediate solution would be to consider the output symbols 0 and \( \frac{1}{4} \) as being one output symbol, and the output symbols \( \frac{3}{4} \) and 1 as another output symbol, thereby converting the channel into a BSC, but this is not an efficient solution.

Horstein's scheme [45] is easily modified for the channel of Figure 1.15. The receiver's distribution is modified by each received symbol. Assume that the transmitter sends
Figure 8.1: 2-up, 3-down median paths
a 1 when the message point lies in the lower half of the unit interval, otherwise 0 is the transmitted symbol (which is opposite to Horstein’s scheme but more suitable when observing Figure 1.15). Then, by computing the a posteriori receiver’s distribution, the median (or midpoint) moves down upon receiving $\frac{3}{4}$ or 1, and moves up when $\frac{1}{4}$ or 0 is received. In fact, the median moves less upon receiving $\frac{1}{4}$ or $\frac{3}{4}$, then when respectively 0 or 1 are received. It is not clear how the modified Horstein’s scheme is “translated” into an efficient (modified) multiple-repetition strategy.

**More general channel models.** As mentioned in Section 7.1 of this thesis, multiple-repetition feedback coding on discrete memoryless channels with noiseless delayless feedback could be extended to coding for more general channel models. A solution for channels with delayed feedback is already mentioned in subsection 7.3.3. Furthermore, in Chapter 6 some variations on multiple-repetition strategies are presented that are suitable for discrete memoryless channels with high channel error probabilities. An important advantage of the alternative multiple-repetition strategies is that they do not require that all received symbols are feedback. More precisely, the binary strategies with parameter $\nu$ (see Chapter 6) require $1/\nu$ trits feedback per forward transmission. Strategies that require less feedback are useful when the transmitting power at the receiver’s side is limited or when less symbols can be fed back for some reason. The low feedback rate can also be exploited to correct possible channel errors in a noisy feedback channel.

Although the capacity of a DMC is not increased by the availability of a noiseless feedback channel, the capacity of a channel with memory can indeed be increased when feedback is available. Some examples of channels with memory where the channel capacity is increased by feedback are given by Dobrushin [27]. The subject of coding for channels with memory and feedback is difficult. Butman [20] considers coding schemes for arbitrary Gaussian channels with (noisy) feedback. Tiernan [93] presents a coding scheme for autoregressive Gaussian channels with noiseless feedback. Ahlswede [3] provides a constructive proof of the coding Theorem for channels with arbitrarily varying channel probability functions in the presence of a noiseless feedback channel. Most known results (see e.g. Tiernan and Schalkwijk [92], and more recently Alajaji [5]) on channels with memory and feedback are about determining the capacity of such channels. It would be nice to have a simple and efficient coding scheme for channels with memory; perhaps an adaptive modification of multiple-repetition feedback strategies.

### 8.3 Conclusions

Much is known about multiple-repetition feedback coding. Many existing and new results are found in this thesis: starting with a description of the ideas from which multiple-repetition strategies originated, through a presentation of new results on block coding, recursive coding, achieving capacity, properly choosing the parameters, generalizations, and finally overviewing the possible applications. On the other hand, there is still room for further research on multiple-repetition feedback coding. Several directions for future research were suggested.
Bibliography


Appendix A

Asymptotic error exponents

In this Chapter some asymptotic exponents are calculated that are used in subsection 3.2.2. Remember that all logarithms are to the base $|X|$. 

A.1 Asymptotic exponent of $P_n^s$

To simplify the notation we eliminated the $x$. Let $\mathcal{Y} = \{0, 1, \ldots, Y-1\}$. Let $k_y$ $(1 \leq y < Y)$ be positive integers. Let $k_0 = 0$. Let $p_y$ $(y \in \mathcal{Y})$ be probabilities with $\sum_{y \in \mathcal{Y}} p_y = 1$ and $p_0 > 0$. (In the back of our mind we have $k_y = k_{xy}$ and $p_y = p_{xy}$.) Consider the infinite markov chain with states $0, 1, 2, \ldots$, and transition probabilities $\Pr(s + k_y - 1 | s) = p_y$ for $y \in \mathcal{Y}$ and $s > 0$. Denote the transition probability matrix by $P$. Let $Q$ be the matrix $I - tP$ with complex variable $t$. Then $Q_{ab}^{-1}(t) = \sum_{n \geq 0} p_{abn} t^n$, where $p_{abn}$ is the probability of going from state $a$ to state $b$ in $n$ steps ($a, b \geq 0$). Note that $p_{n0}$ is also denoted by $p_n$. Let $Q^{r}(t)$ denote the square submatrix of $Q(t)$ with rows and columns $0, 1, \ldots r - 1$, then

$$
(Q^r)_{ab}^{-1}(t) = \frac{(b, a) \text{ minor of } Q^r(t)}{\det Q^r(t)} \quad (A.1)
$$

for $0 \leq a, b < r$. Let $s > 0$ be an arbitrary state. Denote the determinant of $Q^r(t)$ by $\alpha_r$, and the $(0, s)$ subdeterminant of $Q^r(t)$ (the determinant of the matrix $Q^r(t)$ with row $0$ and column $s$ deleted) by $\beta^r_s$, then one can derive the relations

$$
\alpha_r = \begin{cases} 
1 & \text{ if } 1 \leq r \leq \min_{0 < y < Y} k_y, \\
\alpha_{r-1} - \sum_{0 < y < Y} tp_y (tp_0)^{k_y-1} \alpha_{r-k_y} & \text{ if } r > \min_{0 < y < Y} k_y,
\end{cases} \quad (A.2)
$$

where $\alpha_r$ is defined as $0$ for $r \leq 0$, and

$$
\beta^r_s = \begin{cases} 
\alpha_r & \text{ if } 1 \leq r \leq s, \\
(tp_0)^s \alpha_{r-s} & \text{ if } r > s.
\end{cases} \quad (A.3)
$$

Consequently $Q_{s0}^{-1}(t) = \lim_{r \to \infty} (-tp_0)^r \frac{\alpha_{r-s}}{\alpha_r}$. Let $\alpha'_r = (tp_0)^{-r} \alpha_r$ $(r \geq 0)$, then $\alpha'_r$ satisfies $(tp_0)\alpha'_r = \alpha'_{r-1} - \sum_{0 < y < Y} (tp_y)\alpha'_{r-k_y}$ for sufficiently large $r$ and $Q_{s0}^{-1}(t) = (-1)^{s} \lim_{r \to \infty} \frac{\alpha'_{r-s}}{\alpha'_r}$. Let $\gamma(t)$ be the (complex) solution of

$$
tp_0 = \gamma - \sum_{0 < y < Y} (tp_y) \gamma^{k_y} \quad (A.4)
$$

131
with the smallest absolute value, then it follows that

\[ Q_{a_0}^{-1}(t) = (-1)^k \gamma(t)^k. \quad (A.5) \]

Our next investigations will concern the radius of convergence \( R \) for the Taylor series of \( Q_{a_0}^{-1}(t) \), because \( R \) determines the exponent of \( p_n^* \) for large \( n \) (see Equation A.6). From Equation A.5 it follows that the Taylor series of \( \gamma(t) \) has the same radius of convergence, so we are interested in the singularity of \( \gamma(t) \) with smallest absolute value. Since \( \gamma \) is an algebraic function of \( t \), the singularities are branch points and solutions of \( \frac{d \gamma}{dt} = 0 \). Combining this with Equation A.4 one obtains

\[ p_0 = \sum_{0 < y < Y} p_y (k_y - 1) \gamma^k_y \quad \text{and} \quad t = (\sum_{0 < y < Y} p_y k_y \gamma^{k_y - 1})^{-1}, \]

so we look for the complex solution \( \gamma \) of \( p_0 = \sum_{0 < y < Y} p_y (k_y - 1) \gamma^k_y \) with smallest absolute value. Assume w.l.o.g. there is a \( y, 0 < y < Y \), such that \( p_y > 0 \) and \( k_y > 1 \). Since \( \sum_{0 < y < Y} p_y (k_y - 1) \gamma^k_y \) is strictly increasing for real positive \( \gamma \), and \( p_0 > 0 \), there is a unique real positive solution \( \gamma \) of \( p_0 = \sum_{0 < y < Y} p_y (k_y - 1) \gamma^k_y \). Call this solution \( \gamma_R \). Let \( \gamma \) be a complex solution of \( p_0 = \sum_{0 < y < Y} p_y (k_y - 1) \gamma^k_y \). Since \( p_y \geq 0 \) and \( k_y \geq 1 \) for \( 0 < y < Y \), we obtain \( p_0 \leq \sum_{0 < y < Y} p_y (k_y - 1) |\gamma|^{k_y} \), so \( |\gamma| \geq \gamma_R \). It follows that \( R = (\sum_{0 < y < Y} p_y k_y \gamma_{R}^{k_y - 1})^{-1} \).

The number \( R \) can be (numerically) computed, and satisfies

\[ E_1(x) = \lim_{n \to \infty} \frac{-\log p_n^*}{n} = \log R. \quad (A.6) \]

### A.2 Asymptotic exponent of the alternative estimation error

In this Section we use the ideas and notation of Section 3.2. Fix \( x \in \mathcal{X} \). A sequence of the form \( y_1 x^{k_{y_1}-1} y_2 x^{k_{y_2}-1} \ldots y_\nu x^{k_{y_\nu}-1} \) where \( \nu \geq 1 \) and \( y_n \neq x \) \((1 \leq n \leq \nu)\) is called an \( x \)-flip sequence.

Let \( n > 0 \). We compute the asymptotic \((D \to \infty)\) exponent of the probability

\[ P_e^2 = \Pr(r_n \ldots r_{n+D-1} = t_n \ldots t_{n+D-1} \text{ is an } x \text{-flip sequence (and } r_{n+D} = x)). \quad (A.7) \]

Note that the extra condition \( r_{n+D} = x \) does not influence the probability in an asymptotic sense. Let \( \log c \) be the amount of information per precoded symbol. In Section 4.1 is shown how \( c \) can be computed. Let \( f \) be an \( x \)-flip sequence of length \( D \). Since there are approximately \( c^D \) precoded sequences of length \( D \), the probability that the appropriate subsequence of the precoded sequence \( p \) (i.e. the subsequence that is transmitted at transmissions \( n \) up to \( n + D - 1 \)) equals \( f \) is close to \( c^{-D} \). Let \( s_y \) \((y \in \mathcal{Y}, y \neq x)\) be the number of subsequences \( y x^{k_{y_1}-1} \) in \( f \). Then \( \Pr(r_n \ldots r_{n+D-1} = f \mid t_n \ldots t_{n+D-1} = f) = \prod_{y \neq x} (p_{yy} p_{xx}^{-1})^{s_y} \). Therefore

\[ P_e^2 \approx (p_{xx}/c)^D \cdot \sum_{s_y(y \neq x) : \sum_y s_y = D} \left( \sum_{y \neq x} s_y \right) \prod_{y \neq x} (p_{yy}/p_{xx})^{s_y}. \quad (A.8) \]

As for the asymptotic exponent, the two sides of Equation A.8 are equal. Fix \( s_y = \frac{s_y}{D} \) \((y \in \mathcal{Y}, y \neq x)\) as \( D \to \infty \), then

\[ \lim_{D \to \infty} \frac{1}{D} \log \left\{ \left( \frac{\sum_{y \neq x} s_y}{s_y(y \neq x)} \right) \prod_{y \neq x} (p_{yy}/p_{xx})^{s_y} \right\} = \sum_{y \neq x} \frac{s_y}{s_y(y \neq x)} \log \frac{p_{yy}}{s_y(y \neq x)} + \left( \sum_{y \neq x} s_y \right) \log \left( \sum_{y \neq x} s_y \right). \quad (A.9) \]
When the right-hand side of Equation A.9 is maximized under the constraint \( \sum_{y \neq x} k_{xy} s_y = 1 \) using Lagrange we obtain as a maximum \(- \log \gamma_x\), where \( \gamma_x \) is the solution \( \gamma > 0 \) of \( \sum_{y \neq x} p_{yy} \gamma^{k_{xy}} = p_{xx} \). Since the number of \( s_y(y \neq x) \) that satisfy \( \sum_{y \neq x} k_{xy} s_y = D \) is upperbounded by \( D^{D/1} \), we obtain

\[
E_2(x) = \lim_{D \to \infty} - \frac{\log D^2}{D} = \log \frac{\gamma_x}{p_{xx}}.
\]  

(A.10)
Appendix B

Binary precoding rate for increasing repetition parameters

In this Chapter a more detailed proof is presented for the binary case of Lemma 5.3.

**Lemma B.1** Let \( \mathcal{X} = \{0, 1\} \). Let \( q \) be a symbol precoding distribution. Then the maximal precoding rate as a function of the repetition parameters satisfies

\[
\tilde{R}_p(q) \triangleq h(q) - (1/\ln 2)(q_0 q_1^{k_{10}} + q_1 q_0^{k_{01}})
\]

for increasing repetition parameters.

**Proof:** Consider the binary precoded sequences of length \( L \). Split each precoded sequence into subsequences \( 0^i1^j \) \((1 \leq i < k_{01}, 1 \leq j < k_{10})\), hereby neglecting w.l.o.g. precoded sequences starting with \( 1 \) or \( 0^i \) for \( i \geq k_{01} \) and precoded sequences ending with \( 0 \). Let \( a_{ij}, 1 \leq i < k_{01}, 1 \leq j < k_{10} \) be the number of subsequences \( 0^i1^j \). The total number \( M_L \) of precoded sequences then equals the multinomial coefficient \( \sum_{i,j} a_{ij} \) choose \( a_0, ..., a_{k_{01}-1}, k_{10} \)-1. Let \( L \) increase and fix the fractions \( \alpha_{ij} \) that go to \( \alpha \). We obtain a precoding rate \( \log_2 M_L / L \) that goes to

\[
\alpha = \sum_{i,j} \alpha_{ij} \log_2 \left( \sum_{i,j} \alpha_{ij} \right) - \sum_{i,j} \alpha_{ij} \log_2 \alpha_{ij}.
\]

(B.2)

The fractions of zeros and ones are \( q_0 = \sum_{i,j} \alpha_{ij} \) and \( q_1 = \sum_{i,j} \alpha_{ij} \) respectively. In order to maximize the precoding rate for a given symbol precoding distribution, we define the Lagrange function

\[
\mathcal{L}(\alpha_{11}, ..., \alpha_{k_{01}-1,k_{10}-1}, \lambda_0, \lambda_1) = \alpha + \lambda_0(q_0 - \sum_{i,j} \alpha_{ij}) + \lambda_1(q_1 - \sum_{i,j} \alpha_{ij}).
\]

(B.3)

The partial derivative of \( \mathcal{L} \) to \( \alpha_{ab} \) equals \( \log_2 (\sum_{i,j} \alpha_{ij}) - \log_2 \alpha_{ab} - \lambda_0 a - \lambda_1 b \). Let \( x_i = 2^{-\lambda_i} \) \((i = 0, 1)\). By setting all partial derivatives to zero, we derive \( \alpha = \lambda_0 q_0 + \lambda_1 q_1, \sum_{i,j} x_i^a x_j^b = 1 \), and \( q_1 \sum_{i,j} x_i^a x_j^b = q_0 \sum_{i,j} x_i^a x_j^b \). The last two Equations reduce to \( (x_0 - x_{01}^k)(x_1 - x_{10}^k) = (x_0 - 1)(x_1 - 1) \) and \( q_1(1 - x_1)(1 - x_1^{k_{10} - 1})(1 - k_{01} x_0^{k_{10} - 1} + (k_{01} - 1)x_0^{k_{10}}) = q_0(1 - x_0)(1 - x_0^{k_{10} - 1})(1 - k_{10} x_1^{k_{10} - 1} + (k_{10} - 1)x_1^{k_{01}}) \). Let \( (\chi_0, \chi_1), 0 < \chi_i < 1 \((i = 0, 1)\), be the limiting value of \((x_0, x_1)\) for increasing \( k_{01} \) and \( k_{10} \), then \( \chi_0 \chi_1 = (\chi_0 - 1)(\chi_1 - 1) \) and \( q_1(1 - \chi_1) = q_0(1 - \chi_0) \), so \( \chi_i = q_i \((i = 0, 1)\). Note that this coincides with the limiting value of \( \alpha \), which is \( -q_0 \log_2 \chi_0 - q_1 \log_2 \chi_1 = h(q) \). Define \( \varepsilon_i = x_i - q_i \((i = 0, 1)\), then we derive \( x_0 x_1^{k_{10}} + x_1 x_0^{k_{01}} = \chi_0^{k_{10} - 1} + \chi_1^{k_{01} - 1} \).
Since $\varepsilon_0 + \varepsilon_1$ gets exponentially small, we assume $|\varepsilon_0| \ll 1/k_{10}$ and $|\varepsilon_1| \ll 1/k_{10}$. It follows that $\varepsilon_0 + \varepsilon_1 \approx q_0 k_{10} + q_1 k_{10}$. Finally, $R_p(q) = -q_0 \log_2 x_0 - q_1 \log_2 x_1 = h(q) - q_0 \log_2 (1 + \varepsilon_0/q_0) - q_1 \log_2 (1 + \varepsilon_1/q_1) = h(q) - (1/\ln 2)(\varepsilon_0 + \varepsilon_1) = h(q) - (1/\ln 2)(q_0 k_{10} + q_1 k_{10})$. \hfill \blacksquare
Appendix C

Decreasing capacities

In this Chapter a first order approximation of $C^\nu(p)$ (as defined by Equation 6.3) is derived, which, together with figure 6.3, suggests that $C^\nu(p)$ is decreasing in $\nu$ with equality only for $p = 1/2$. We recapitulate the definition of $C^\nu(p)$:

**Definition C.1** The capacity $C^\nu(p)$ of the $\nu^{th}$ meta channel in bits equals

$$C^\nu(p) = \frac{p^\nu + (1-p)^\nu}{\nu} \cdot C\left(\frac{p^\nu}{p^\nu + (1-p)^\nu}\right),$$

where $C(p) = 1 - h(p)$ is the capacity of the BSC with channel error probability $p$.

Equation C.1 can, by using $1 - h(p) = p \log_2(2p) + (1 - p) \log_2(2(1-p))$, be rewritten as

$$C^\nu(p) = \frac{p^\nu}{\nu} \log \frac{2p^\nu}{(1-p)^\nu + p^\nu} + \frac{(1-p)^\nu}{\nu} \log \frac{2(1-p)^\nu}{(1-p)^\nu + p^\nu}.$$  \hspace{1cm} (C.2)

It is clear that $C^\nu(1/2) = 0$ for all $\nu$. For $p < 1/2$ a first order approximation for $C^\nu(p)$ is derived. The approximation is good for small $p$, and for large $\nu$. Since $\log[1 + (p/(1-p))^\nu]$ is close to $(1/\ln 2)(p/(1-p))^\nu$, we obtain

$$C^\nu(p) \approx \frac{p^\nu}{\nu} \left[1 + \nu \log \frac{p}{1-p} - \frac{1}{\ln 2} \left(\frac{p}{1-p}\right)^\nu\right] + \frac{(1-p)^\nu}{\nu} \left[1 - \frac{1}{\ln 2} \left(\frac{p}{1-p}\right)^\nu\right]$$

$$\approx p^\nu \log \frac{p}{1-p} + \frac{(1-p)^\nu}{\nu}.$$  \hspace{1cm} (C.3)

Although $\log(p/(1-p)) < 0$, the dominating factor in Equation C.4 is $(1-p)^\nu / \nu$ which is clearly decreasing in $\nu$. This suggests that $C^\nu(p)$ is also decreasing in $\nu$. 

136
Acknowledgements

I would like to thank all the members of the Information and Communication Theory group of the Eindhoven University of Technology for their pleasant company during the last years, especially my promotor Prof.dr.ir. J.P.M. Schalkwijk, who never lost his positive attitude while giving me the opportunity to carry out my doctoral research.

Special thanks go to Frans Willems, Tjalling Tjalkens, Jan Rooyackers, Erik Meeuwissen, and Mortaza Shao Bargh for careful reading and reviewing of my thesis.

Furthermore, I would like to thank all the members of my committee for reviewing and finally approving my thesis. Also I would like to thank Karel Post for his functional theoretical advice, and Erik Meeuwissen for being a sociable room-mate.

Finally, I wish to thank all of my family and friends for their support.
Notation

An overview is given of the notation that is used throughout this thesis. Notation that is only used locally is not mentioned. When a variable is underlined, it denotes a sequence or vector ($\underline{v}$ is the vector with components $v_x, x \in X$). Superscript is used to denote a sequence of equal symbols, e.g. $0^n$ is a sequence consisting of $n$ symbols 0, but is also used as ordinary exponentiation of real numbers. Indexing is done by using subscripts and/or superscripts.

<table>
<thead>
<tr>
<th>Symbol(s)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSC</td>
<td>Binary Symmetric Channel</td>
</tr>
<tr>
<td>$c$</td>
<td>code word (the precoded sequence concatenated with a tail)</td>
</tr>
<tr>
<td>$c_k$</td>
<td>the solution $1 &lt; c &lt; 2$ of $c^k = 2c^{k-1} - 1$</td>
</tr>
<tr>
<td>$C$</td>
<td>channel capacity</td>
</tr>
<tr>
<td>$d$</td>
<td>the decoded sequence</td>
</tr>
<tr>
<td>$D$</td>
<td>decoding delay</td>
</tr>
<tr>
<td>DMC</td>
<td>Discrete Memoryless Channel</td>
</tr>
<tr>
<td>DMCF</td>
<td>Discrete Memoryless Channel with complete Feedback</td>
</tr>
<tr>
<td>$D(p</td>
<td></td>
</tr>
<tr>
<td>$D_X(p</td>
<td></td>
</tr>
<tr>
<td>$e$</td>
<td>the number of channel errors in a block</td>
</tr>
<tr>
<td>$e_{xy}$</td>
<td>the number of $x \rightarrow y$ channel errors in a block</td>
</tr>
<tr>
<td>$E$</td>
<td>error exponent</td>
</tr>
<tr>
<td>$f$</td>
<td>the correctable error fraction</td>
</tr>
<tr>
<td>$h$</td>
<td>the binary entropy function</td>
</tr>
<tr>
<td>$H_X(q)$</td>
<td>$X$-ary entropy ($-\sum_{x \in X} q_x \log q_x$)</td>
</tr>
<tr>
<td>$I(X;Y)$</td>
<td>the mutual information of random variables $X$ and $Y$</td>
</tr>
<tr>
<td>$k$</td>
<td>the binary symmetric repetition parameter</td>
</tr>
<tr>
<td>$k_{xy}$</td>
<td>the number of repetitions caused by an $x \rightarrow y$ channel error</td>
</tr>
<tr>
<td>Symbol(s)</td>
<td>Meaning</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>L</td>
<td>the number of information symbols in a block</td>
</tr>
<tr>
<td>ln</td>
<td>natural logarithm (to the base $e$)</td>
</tr>
<tr>
<td>log</td>
<td>logarithm to the base $</td>
</tr>
<tr>
<td>$m$</td>
<td>a message</td>
</tr>
<tr>
<td>$\hat{m}$</td>
<td>the estimated message</td>
</tr>
<tr>
<td>$M$</td>
<td>the number of messages</td>
</tr>
<tr>
<td>$\mathcal{M}$</td>
<td>the set of messages</td>
</tr>
<tr>
<td>$n$</td>
<td>transmission number</td>
</tr>
<tr>
<td>$N$</td>
<td>block length</td>
</tr>
<tr>
<td>$p$</td>
<td>the error probability of a BSC</td>
</tr>
<tr>
<td>$\hat{p}$</td>
<td>the precoded sequence</td>
</tr>
<tr>
<td>$\hat{\hat{p}}$</td>
<td>the estimated precoded sequence</td>
</tr>
<tr>
<td>$P_e$</td>
<td>probability of erroneous decoding</td>
</tr>
<tr>
<td>$p_k$</td>
<td>the solution $0 &lt; p &lt; 1/2$ of $2p(2 - 2p)^k - 1 = 1$</td>
</tr>
<tr>
<td>$\Pr$</td>
<td>Probability</td>
</tr>
<tr>
<td>$p_{xy}$</td>
<td>$\Pr{\text{output} = y \mid \text{input} = x}$</td>
</tr>
<tr>
<td>$\pi$</td>
<td>the capacity achieving channel output distribution</td>
</tr>
<tr>
<td>$q$</td>
<td>$1 - p$</td>
</tr>
<tr>
<td>$\tilde{q}$</td>
<td>the symbol precoding distribution</td>
</tr>
<tr>
<td>$Q$</td>
<td>the set of symbol precoding distributions</td>
</tr>
<tr>
<td>$r$</td>
<td>the received sequence</td>
</tr>
<tr>
<td>$R$</td>
<td>transmission rate</td>
</tr>
<tr>
<td>$R_{\text{crit}}$</td>
<td>the critical rate</td>
</tr>
<tr>
<td>$R_p$</td>
<td>precoding rate</td>
</tr>
<tr>
<td>$t$</td>
<td>the transmitted sequence</td>
</tr>
<tr>
<td>$\hat{t}$</td>
<td>estimated transmitted sequence</td>
</tr>
<tr>
<td>$\tau_x$</td>
<td>the expected number of transmissions needed to send an $x$-symbol and correct all occurring channel errors</td>
</tr>
<tr>
<td>$x$</td>
<td>input symbol</td>
</tr>
<tr>
<td>$\mathcal{X}$</td>
<td>input alphabet</td>
</tr>
<tr>
<td>$y$</td>
<td>output symbol</td>
</tr>
<tr>
<td>$\mathcal{Y}$</td>
<td>output alphabet</td>
</tr>
<tr>
<td>$\mathcal{Y} \setminus \mathcal{X}$</td>
<td>the set ${y \in \mathcal{Y} \mid y \notin \mathcal{X}}$</td>
</tr>
<tr>
<td>$\omega$</td>
<td>the fixed ratio of $L$ and $N$</td>
</tr>
</tbody>
</table>

$\hat{\nu}_1, \ldots, \hat{\nu}_n$ same order of magnitude

$\binom{\nu_1, \ldots, \nu_n}{(\nu_1 \ldots, \nu_n)}$ the multinomial coefficient $\nu!/(\nu_1! \cdots \nu_n)!$

$(\nu - \sum_{1 \leq i \leq n} \nu_i)!$ there exists a constant $c > 0$ such that $|f(p)| \leq c \cdot p$ for $p$ sufficiently small

$f(p) = O(p) (p \to 0)$
<table>
<thead>
<tr>
<th>Symbol(s)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>x</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{X}</td>
</tr>
</tbody>
</table>
## Index

<table>
<thead>
<tr>
<th>Term</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSL</td>
<td>116</td>
</tr>
<tr>
<td>Ahlswede</td>
<td>18, 20, 115, 122</td>
</tr>
<tr>
<td>alternating tail</td>
<td>43, 48</td>
</tr>
<tr>
<td>alternative strategies</td>
<td>94</td>
</tr>
<tr>
<td>army scouts</td>
<td>117</td>
</tr>
<tr>
<td>Becker</td>
<td>38, 44, 47, 48, 54, 79, 100</td>
</tr>
<tr>
<td>Berlekamp</td>
<td>11, 16–18, 28, 88</td>
</tr>
<tr>
<td>block</td>
<td></td>
</tr>
<tr>
<td>coding</td>
<td>5</td>
</tr>
<tr>
<td>decoding</td>
<td>42, 47</td>
</tr>
<tr>
<td>encoding</td>
<td>42, 47</td>
</tr>
<tr>
<td>length</td>
<td>5, 42, 65, 88</td>
</tr>
<tr>
<td>Burnašev</td>
<td>18, 19, 115</td>
</tr>
<tr>
<td>channel</td>
<td></td>
</tr>
<tr>
<td>&quot;super&quot;</td>
<td>74</td>
</tr>
<tr>
<td>asymmetric</td>
<td>39</td>
</tr>
<tr>
<td>binary defect</td>
<td>110</td>
</tr>
<tr>
<td>binary symmetric</td>
<td>4</td>
</tr>
<tr>
<td>capacity</td>
<td>5, 40, 65, 69, 80, 120, 136</td>
</tr>
<tr>
<td>discrete memoryless</td>
<td>4</td>
</tr>
<tr>
<td>duplex</td>
<td>113</td>
</tr>
<tr>
<td>induced channel input distribution</td>
<td>70</td>
</tr>
<tr>
<td>meta</td>
<td>91, 136</td>
</tr>
<tr>
<td>strict-sense symmetric</td>
<td>39</td>
</tr>
<tr>
<td>symmetric</td>
<td>79</td>
</tr>
<tr>
<td>uniform</td>
<td>79</td>
</tr>
<tr>
<td>wide-sense symmetric</td>
<td>39, 79</td>
</tr>
<tr>
<td>with side state information</td>
<td>111</td>
</tr>
<tr>
<td>Z-channel</td>
<td>14, 73, 108</td>
</tr>
<tr>
<td>zero error capacity</td>
<td>6</td>
</tr>
<tr>
<td>Chernoff bound</td>
<td>66, 93</td>
</tr>
<tr>
<td>code word</td>
<td>5, 43</td>
</tr>
<tr>
<td>code word estimator</td>
<td>34, 55</td>
</tr>
<tr>
<td>coding</td>
<td></td>
</tr>
<tr>
<td>block</td>
<td>5, 42, 44, 46</td>
</tr>
<tr>
<td>convolutional</td>
<td>5</td>
</tr>
<tr>
<td>enumerative</td>
<td>100</td>
</tr>
<tr>
<td>one-way</td>
<td>5, 98, 113</td>
</tr>
<tr>
<td>random</td>
<td>5</td>
</tr>
<tr>
<td>recursive</td>
<td>55</td>
</tr>
<tr>
<td>sequential</td>
<td>8</td>
</tr>
<tr>
<td>variable length</td>
<td>7</td>
</tr>
<tr>
<td>concatenation construction</td>
<td>8</td>
</tr>
<tr>
<td>inverse</td>
<td>9</td>
</tr>
<tr>
<td>constraint length</td>
<td>6</td>
</tr>
<tr>
<td>correctable error fraction</td>
<td>see error</td>
</tr>
<tr>
<td>decision error</td>
<td>91</td>
</tr>
<tr>
<td>decoding</td>
<td></td>
</tr>
<tr>
<td>erasure</td>
<td>13</td>
</tr>
<tr>
<td>list</td>
<td>13</td>
</tr>
<tr>
<td>maximum likelihood</td>
<td>13</td>
</tr>
<tr>
<td>delay</td>
<td>8, 55, 117</td>
</tr>
<tr>
<td>fixed</td>
<td>56, 119</td>
</tr>
<tr>
<td>variable</td>
<td>56</td>
</tr>
<tr>
<td>Dobrušin</td>
<td>16, 18, 122</td>
</tr>
<tr>
<td>dummy reversal</td>
<td>106</td>
</tr>
<tr>
<td>economic market</td>
<td>108</td>
</tr>
<tr>
<td>Elias</td>
<td>5, 7, 10, 115</td>
</tr>
<tr>
<td>encoder</td>
<td></td>
</tr>
<tr>
<td>inverse</td>
<td>34, 55</td>
</tr>
<tr>
<td>enumerative coding</td>
<td>see coding</td>
</tr>
<tr>
<td>erasure</td>
<td>40, 91, 120</td>
</tr>
<tr>
<td>erasure decoding</td>
<td>13</td>
</tr>
<tr>
<td>error</td>
<td></td>
</tr>
<tr>
<td>alternative estimation error</td>
<td>62, 132</td>
</tr>
<tr>
<td>average probability of error</td>
<td>5, 65</td>
</tr>
<tr>
<td>correctable error fraction</td>
<td>88</td>
</tr>
<tr>
<td>correcting capability</td>
<td>44, 48</td>
</tr>
<tr>
<td>decision</td>
<td>94</td>
</tr>
<tr>
<td>exponent</td>
<td>7, 61, 131</td>
</tr>
<tr>
<td>maximal correctable error fraction</td>
<td>16, 88</td>
</tr>
<tr>
<td>propagation</td>
<td>64, 107, 120</td>
</tr>
<tr>
<td>estimation algorithm</td>
<td>55</td>
</tr>
<tr>
<td>Fano</td>
<td>10</td>
</tr>
<tr>
<td>feedback</td>
<td></td>
</tr>
<tr>
<td>decision</td>
<td>8, 117</td>
</tr>
</tbody>
</table>
delayed, 110
information, 8
flip sequence, 62, 106, 132
Forney, 8, 13, 15, 19
Gallager, 10, 11, 14, 79
generalized strategy, 89
generating polynomial, 67
hard decision, 79
Horstein, 21, 36, 109, 115
information symbol, 42, 65, 98
informational divergence, 56, 62, 78, 80
interleaving, 110
interval
  bit, 24
  error, 25
  message, 24
  transmission, 23
inverse concatenation construction, 9
Kudryashov, 15, 19
left-to-right decoding, 55, 119
list decoding, 13
maximum likelihood decoding, 13
median, 28, 56, 69, 101, 122
  path, 28
memory with known defects, 74, 110
message point, 22
midpoint, 23
Post, 27, 34, 55
precoding, 33, 98
  inverse precoder, 39, 98
  maximal achievable precoding rate, 104
  maximal precoding rate, 82
precoded sequence, 42, 55, 59
precoder, 39, 98
precoding rate, 103
rate, 66, 67, 134
  symbol precoding distribution, 47, 59, 67, 69, 81, 84
prefix-synchronized code, 104
quantile, 34
random coding bound, 10
rate, 5
achievable, 5, 65
computation cutoff, 9
critical, 9
maximal achievable precoding rate, 104
maximal achievable transmission rate, 67, 77
maximal precoding rate, 82
precoding rate, 67, 103, 134
transmission rate, 65, 84, 89
receiver's distribution, 22
recursive
decoding, 59
encoding, 59
regular grammar, 102
reliability, 115
reliability function, 7
repeat request, 8
right-to-left decoding, 60
ROM, 108
runlength-limited sequences, 106
satellite, 114, 117
Schalkwijk, 7, 18, 27, 28, 34, 38, 55, 101, 103, 106, 113
self-information, 83
Shannon, 4–6, 10, 11, 16, 17, 20, 68, 84, 111
sliding block code, 64
sliding block coding, 107
soft decision, 40, 120
sphere packing bound, 10
state diagram, 36, 57, 102
state threshold, 56
statistical experiments, 115
symbol precoding distribution, see precoding
tail, 42, 48
Tail Condition, 48, 119
Tangent bound, 17, 88
Telatar, 14
Tiernan, 122
transmission rate, see rate
twenty questions with a liar, 20
Ulam's game, 20
Vinck, 104, 114
Viterbi, 7, 11, 14, 15, 114
Viterbi algorithm, 11
Volume bound, 17, 88

Weldon, 7, 13, 16, 28

Yamamoto and Itoh, 15, 18

Z-channel, 14, 73, 108
zero error capacity, 6

Zigangirov, 17, 18, 20, 28, 38, 101, 115
Samenvatting


Wanneer informatie m.b.v. repetitie-strategieën in blokken van vaste lengte wordt overgezonden is het noodzakelijk om binnen het blok een staart aan het bericht toe te voegen om fouten te kunnen corrigeren. Het foutencorrigerende vermogen per blok is slechts dan gegarandeerd wanneer de staart op een geschikte manier is geconstrueerd. De mogelijkheid om een geschikte staart te construeren hangt af van de repetitie-parameters van de strategie. Er wordt een nieuwe staartconstructie bepaald en bewezen dat daarmee het foutencorrigerende vermogen voor een grotere klasse van repetitie-parameters gegarandeerd is.

Repetitie-strategieën kunnen ook gebruikt worden om een continue stroom van informatie over te sturen. Een ontvangen symbool wordt dan met een bepaalde vertraging gedeencodeerd. Voor deze manier van informatie overdracht was de kans dat een symbool fout wordt gedeencodeerd alleen bekend voor binaire geheugenloze kanalen. De kans op een fout decodering wordt uitgerekend voor alle discrete geheugenloze kanalen.

De transmissie snelheid van een repetitie-strategie is in bepaalde gevallen zelfs gelijk aan de capaciteit van het kanaal. Voor symmetrische kanalen was dit al bekend, maar in het proefschrift wordt gedemonstreerd dat dit ook geldt voor asymmetrische kanalen. Bovendien wordt aangetoond hoe voor een willekeurig kanaal de repetitie-parameters gekozen moeten worden om de transmissie snelheid te maximaliseren.

Wanneer het kanaal zo slecht is dat meer dan één derde van alle symbolen fout wordt ontvangen is het onmogelijk om een dergelijke foutenfractie altijd te corrigeren. Er worden een aantal varianten op repetitie-strategieën gepresenteerd die op zulke kanalen een redelijk hoge transmissie snelheid bereiken in combinatie met een lage foutenkans.

Repetitie-strategieën kunnen voor diverse doeleinden gebruikt worden. Enkele voorbeelden zijn satelliet communicatie, opslag van gegevens in halfgeleidersgeheugens, opslag van gegevens in geheugens met bekende fouten, en het (theoretisch) bereiken van de evenwichtsprijs op economische markten. Hoewel al veel bekend is over repetitie-strategieën, zijn er helaas/gelukkig nog steeds een aantal open problemen.
Curriculum Vitae

Thijs Veugen was born on September 25th, 1969 in Weert, the Netherlands.

In 1987, he obtained his VWO diploma at the Philips van Horne Scholengemeenschap in Weert.

After secondary school, he studied Mathematics as well as Computer Science at the Eindhoven University of Technology. His graduation work was performed at the Center for Mathematics and Computer Science in Amsterdam, where he studied electronic cash systems. In 1991 he received his Master of Science degrees, both with distinction, and passed examination in several additional subjects.

In 1992 he started his PhD studies at the Group on Information and Communication Theory at the Electrical Engineering department of the Eindhoven University of Technology. The results of his information theoretical research are presented in this thesis.

In October 1996 he began working at Statistics Netherlands (CBS in Dutch), division Research and Development, sector Statistical Informatics, in Heerlen.
Stellingen behorende bij het proefschrift

*multiple-repetition coding for channels with feedback*

Thijs Veugen
1. Hoewel de symbooldistributie tijdens het precoderen asymptotisch een belangrijke rol speelt voor het behalen van een zo hoog mogelijke transmissie snelheid, kan het voor eindige bloklengeten voordeliger zijn om geen restricties te stellen aan de symbooldistributie tijdens het precoderen.

2. Hoewel een schattingsalgoritme dat de invoergegevens van links naar rechts (dat wil zeggen op volgorde van binnenkomst) verwerkt beter lijkt dan een schattingsalgoritme dat van rechts naar links werkt, hoeft dat niet altijd zo te zijn [1].


4. De eerste stelling van Shannon in [1] is vermeld als Stelling 5.6 van dit proefschrift. Uit het bewijs in Sectie 4.3 van dit proefschrift volgt dat de stelling niet alleen geldt voor geheeltallige duurtijden, maar ook voor reële, wat de toepasbaarheid van de stelling aanzienlijk verruimt.


5. Het geheugengebruik van het encodeeralgoritme en het decodeeralgoritme uit [1] is kwadratisch in de lengte van de codewoorden, maar kan eenvoudig worden teruggebracht zodat het geheugengebruik slechts lineair is in de lengte van de codewoorden [2].


6. De volgende elegante ongelijkheid uit de informatietheorie voor de binaire entropie functie \( h(x) = -x \log_2 x - (1 - x) \log_2 (1 - x) \) geldt voor alle reële getallen \( a \) en \( b \) tussen 0 en 1:

\[
ah(b) + bh(a) \leq (a + b)h \left( \frac{a + b}{2} \right).
\]

Naast het originele (ongepubliceerde) bewijs van Ineke van Overveld en Frans Willems is een bewijs voor deze ongelijkheid door Thijs Veugen vermeld in [1].

7. Het protocol uit het elektronisch-geld-systeem van [1], waarmee een gebruiker geld kan opnemen bij de bank is niet zo veilig als wordt verondersteld. Een slimme bedrieger kan met name de pakkans voor het verkrijgen van een volledig valse munt aanzienlijk verkleinen [2].


8. Samen met Erik Meeuwissen is met gebruik van een computer gezocht naar een nieuwe ondergrens voor het capaciteitagebied van het binair vermenigvuldigingskanaal met behulp van de methode van Te Sun Han [1]. Dit heeft geen verbetering ten opzichte van Shannon's ondergrens [2] opgeleverd.


9. Een bridge probleem:

\[
\begin{array}{c}
\diamondsuit 864 \\
\heartsuit HV5 \\
\spadesuit A983 \\
\clubsuit A75
\end{array}
\begin{array}{c}
\spadesuit 753 \\
\heartsuit 104 \\
\spadesuit V1075 \\
\clubsuit VB104
\end{array}
\begin{array}{ccc}
N & W & O \\
Z
\end{array}
\begin{array}{c}
\spadesuit AHV102 \\
\heartsuit A632 \\
\spadesuit H2 \\
\clubsuit 9
\end{array}
\]

Zuid speelt 7\spadesuit en west komt uit met \spadesuit V. Zuid kan het contract maken bij optimaal tegenspel van oost en west.

10. De uitdrukking "letterlijk en figuurlijk" wordt vaak misbruikt om iets te benadrukken terwijl feitelijk alleen de figuurlijke betekenis van toepassing is.