Coding for Channels with Multiple Localized Burst Errors

A. Mardjuadi

DELFT UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering

Telecommunications and Traffic Control Systems Group

Title: Coding for Channels with Multiple Localized Burst Errors

Author: A. Mardjuadi

Type: Master Thesis

Size: 60 pages

Date: February 1996

Mentor: Dr. Ir. J.H. Weber

Assignment nr: A 744

Period: August 1995 - February 1996

In the concept of localized burst errors, it is assumed that the encoder knows where the burst may occur. On the other hand, the decoder does not have this extra information. It has been proved that the localized-burst-error-correcting-code will produce a higher coderate than any other conventional burst-error-correcting-code. The goal of this thesis is to construct a new code that can correct localized burst errors. In both cases, single and multiple localized bursts, the coderate is indeed improved. Furthermore, a new concept of partially localized bursts will also be introduced.

Key words: Localized burst error, Error correction

SUMMARY

The main purpose of any communication system is to transmit a message from the source through a channel to the destination. An error-correcting-code is usually used to reduce a risk that the information is defected when passing through the channel.

A new concept where the encoder is assumed to know the locations where errors may occur is introduced by L.A. Bassalygo et. al. and is called as localized-errors-correcting-code. At this moment, there are two known construction methods for correcting localized burst errors.

This thesis introduces a new construction method for correcting localized burst errors. A new rule, called the FBI rule, is specially created for the encoder to inform the decoder about the possible locations of the bursts.

From the comparison with other codes, the new construction method proposed in this thesis produces indeed a higher coderate, both for single and multiple localized bursts. This important result is achieved while the complexity of the encoding and decoding procedures remains unchanged.

There is only one drawback for the new technique namely the decrease of the maximum burstlength.

SAMENVATTING

Het voornaamste doel van elk communicatie systeem is om de boodschap vanuit een bron via een kanaal naar een bestemming over te brengen. Een foutenverbeterende code wordt meestal gebruikt om het risico te verkleinen dat de informatie wordt beschadigd wanneer het via een kanaal wordt verzonden.

Een nieuw concept, waarin wordt verondersteld dat de encoder de locaties weet, waar fouten kunnen optreden, is geïntroduceerd door L.A. Bassalygo, e.a. en staat bekend als localized errors correcting code. Sinds kort, bestaan er twee constructies voor het corrigeren van localized burst fouten.

Deze afstudeeropdracht introduceert een nieuwe constructie voor het corrigeren van localized burst fouten. Een nieuwe regel, genoemd als de FBI regel, is speciaal ontworpen voor de encoder om de mogelijke locaties van de bursts aan de decoder te informeren.

Uit de vergelijkingen met andere codes volgt dat de nieuwe constructie inderdaad een hogere coderate produceert, zowel voor single als voor multiple localized bursts. Dit belangrijk resultaat is bereikt terwijl de complexiteit van de codeer- en decodeer-procedures gelijk blijven.

Er is echter één nadeel van de nieuwe constructie namelijk een afname in de maximale burstlengte.

CONTENTS

LIST OF SYMBOLS

$\lceil x \rceil$	:	the ceiling of x, is the least integer greater than or equal to x.
$\lfloor x \rfloor$	:	the floor of x, is the greatest integer less than or equal to x.
α	:	the number of unreliable parts in one codeword.
λ	:	the number of bursts in one codeword.
b	:	the length of one burst.
b_{max}	:	maximum correctable burstlength.
g	:	the length of one unreliable part.
n	:	the length of a codeword.
p	:	the number of parts in one codeword.
q	:	the size of an alphabet.
r	:	the number of symbols used in I-block.
M_j	:	the jth sub-part or block in one part.
P_i	:	the ith part in a codeword.

Chapter 1

INTRODUCTION

The main objective of any communication system is to transmit information from a source through a channel to reach the destination. Unfortunately, the channel (or the transmission medium) introduces a number of effects such as attenuation, distortion, interference and noise. All of these effects create one major problem, namely it is impossible to guarantee an error-free transmission. In other words, we can assume that there is always a risk that the information is defected when passing through the channel.

One possible way to achieve a more reliable communication system is to use an error-correcting code. This system allows an encoder in the transmitter to transform an actual message into a certain codeword by adding redundancy to that message. The codeword is then transmitted over the channel to the receiver. If errors really occur, this codeword becomes a certain vector. At the receiver's side, a decoder is used to retransform the incoming vector (codeword + errors) into the original message. A general communication system is shown in Figure 1.1.

The choice of which code will be used, of course, depends on the situation. We have to realize that one particular code cannot correct all types of error. It is apparent that in some channels, errors have a tendency to appear clustered (burst). In this case, a code that is especially designed to correct burst would always have an extra advantage.

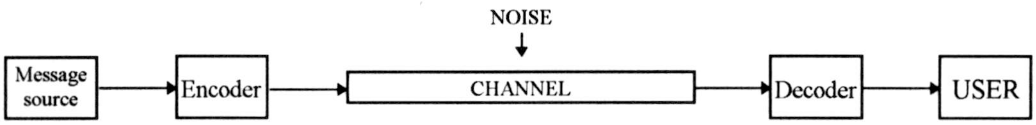

Figure 1.1 : General communication system.

So far, we have assumed that the encoder has no a priori knowledge about the error values or error locations. This kind of codes are usually called conventional error-correcting codes. Another concept, where the values and possible location of errors are known to the encoder, is introduced in [5] and is known as coding for channel with defects. In both methods it is assumed that the decoder has no a priori information about the locations and values of errors.

A new concept where the encoder knows the possible locations of errors but not the error values, is introduced in [1] and is called a code for correction of localized errors. Obviously a conventional error-correcting-code can also be used to correct localized errors, but the important question is : can we improve the performance of the code if the encoder has extra information about the location of the errors.

The goal of this thesis is to construct a new code for correction of multiple localized burst errors. A special construction is also developed for correction of single localized burst errors. Furthermore, we will introduce the concept of partially localized burst errors.

This thesis is divided into seven chapters. The first two chapters introduce the concept of localized (burst) errors. Chapter 3 contains the known results both for codes correcting conventional and localized burst errors. The new construction (Construction A) together with the encoding and decoding procedures are discussed in Chapter 4. A special construction (Construction B) designed for correction of single localized burst is found in Chapter 5. Chapter 6 describes the concept of multiple partially localized burst errors. In the last chapter the conclusions and recommendations are given.

Chapter 2

LOCALIZED ERRORS

2.1 Concept

As mentioned in the previous chapter, the concept of localized errors might be thought of as an intermediate between the conventional error-correcting-codes and coding for channels with defects. In this new concept it is assumed that the encoder knows the positions where errors can (but not necessarily will) occur. This important fact could be useful to enlarge the size of the code. We have to remember that the decoder still doesn't have any extra information about errors in the channel, it only knows the parameters of the channel.

2.2 Definitions

The formal definitions of the described model, referred to as a model for the channel with localized errors, will now be given. We will apply the same notations proposed by Larsson in [2]. These notations will be used throughout this thesis.

The transmitter transmits messages from some message set M through a q-ary channel, where each channel symbol can take values from $\{0,1,\dots, q-1\}$. For this purpose we consider block codes of length n over an alphabet with q elements. The components of a vector x of length n are numbered from 1 to n in the following way : $x = (x_1,\dots, x_n)$. Let E be a subset of $\{1,\dots, n\}$. The set E is the set of positions where errors may occur during transmission over the channel and it is known a priori to the encoder but not to the decoder. Usually E is referred to as the configuration of unreliable positions. In a situation where bursts are considered, we have one more restriction on the set E, namely it consists of consecutive positions.

Denote by $e(E)$ a possible error vector for a certain set E and by $e_i(E)$ the ith component of $e(E)$. The encoder still doesn't know the error values but it knows that there will be no errors in the positions not in E. In other words, the encoder knows that $e_i(E) = 0$ for $i \notin E$. Notice also that the set of unreliable positions may change from one block to another.

The encoding function f is a mapping,

$$f \; : M \times E \rightarrow \{0, 1,\dots, q-1\}^n,$$

i.e. the output from the encoder, $f(m, E)$, depends both on a message $m \in M$ and on the set E of unreliable positions. The output of the channel will be the component-wise addition modulo q of $f(m, E)$ and $e(E)$, denoted $f(m, E) \oplus e(E)$.

The decoder function g is a mapping,

$$g : \{0, 1,\dots, q-1\}^n \rightarrow M \cup \{*\},$$

where $*$ is a special symbol used to indicate that the decoder could not make a proper decision.

Usually a code will be designed to correct a certain number of t localized errors. Then, for every message m, every configuration E such that $|E| \leq t$, and any error vector $e(E)$ with zeros in the position outside E, the following relation must hold,

$$g(f(m, E) \oplus e(E)) = m.$$

This is a general description of the model that is used in this thesis and it is illustrated in Figure 2.1.

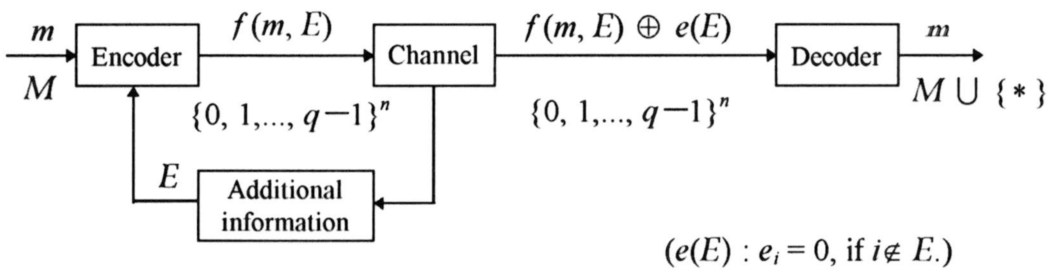

Figure 2.1 : The coding system for correction of localized errors on a q-ary channel.

2.3 Localized burst

On telephone lines, a stroke of lightening or a man-made electrical disturbance frequently affects many adjacent transmitted digits which indicates that errors occur in a form of bursts. This phenomenon can also be found in certain digital communication media such as magnetic tapes, disks and memories in computer storage systems. Therefore, it is very desirable to design codes specially for correcting burst errors. These kind of codes are called burst-error-correcting codes. There are many examples of these codes such as : Fire codes or Reed-Solomon codes (used in CD player).

In the concept of localized burst, it is assumed that the encoder has the ability of knowing the possible locations where the bursts may occur. In other words, the encoder knows all symbols which are not defected by the possible burst, will be transmitted without errors while symbols within the location of the possible burst might suffer from errors. Notice that the decoder does not have any a priori information about the possible locations of the burst. The term multiple localized bursts is used if one codeword contains more than one burst.

2.4 Applications

This section briefly describes two situations where the use of codes correcting localized errors could be advantageous. The first application was actually the original motivation for Larsson to study the problem of localized errors. The second situation is an example of a mobile communication system which could motivate further study of localized burst errors.

2.4.1 Storage medium

We consider some sort of storage medium, for example a magnetic disc. It is possible for the encoder to determine (prior to storage) parts on the storage medium that are unreliable due to irregularities in the material or to uncertainties in the production process.

The decoder is assumed to not know this additional information. Two important reasons why the decoder cannot get the same reliable information as the encoder are that the decoder should be small and cheap.

The storage medium can be considered to be divided into blocks of positions. The extra information about the unreliable parts can be regarded as unreliable positions in a block. The decoder is required to correct a certain number of errors in a block. It is often possible to improve performance by using the additional information about unreliable parts (localized-error-correcting-code).

2.4.2 Mobile radio

We will consider a mobile radio system described by Afanasiev et. al. [10] and show the need of codes correcting (multiple) localized burst errors for this application.

The system consists of one base station communicating with a number of T mobiles in a time division multiple-access system. Each mobile is given a time-slot for communication with the base station. A multi carrier system is assumed, where the mobiles and the base station use the same set of n carrier-frequencies. At each frequency a suitable modulation form is used to transmit one symbol from a set of q symbols.

The carrier-frequencies can be regarded as positions in a codeword of length n over an alphabet with q symbols. If the base station will transmit a message to a mobile, a conventional error-correcting-code is used over the n positions. The mobile decodes the received codeword and will find at what frequencies errors have occurred. Some errors appear independently but others are likely to appear at the same frequencies for several adjacent timeslots. These non-independent errors are often due to slowly varying fading.

If a mobile can listen to and decode messages sent to the preceding k mobiles $(1 \leq k \leq T-1)$, it should be able to distinguish non-independent errors (due to fading) from dependent ones. The mobile can calculate the average number of errors over the n frequencies (averaged by the k observed timeslots) and find at which frequencies fading occurs. The decoder has now knowledge about unreliable carrier-frequencies (unreliable positions). Since the channel is assumed to be the same in both directions within a short period and the non-independent errors typically appear at adjacent carrier-frequencies, the mobile can transmit back to the base station using a (multiple) localized-burst-error-correcting code.

Chapter 3

KNOWN RESULTS

3.1 Conventional burst error correcting codes

Before looking at codes specially designed to correct localized bursts, it is appropriate to review some known results for conventional burst-error-correcting-codes. A burst error of length b will be defined as a sequence of b error symbols with the first and last of which are nonzero. Denote by $S(n,b)$ the maximum size of a code of length n that can correct burst of length b or less. There are two theorems for the upper bound of $S(n,b)$. The first upper bound is the Hamming bound for burst and is derived as follows :

$$S(n,b) \le \frac{q^{n-b+1}}{(q-1)(n-b+1)+1} \tag{3.1}$$

The second bound was originally formulated by Reiger, see [6] :

$$S(n,b) \le q^{n-2b} \tag{3.2}$$

Denote by $S(n,b,\lambda)$ the maximum size of a conventional code of length n that can correct λ bursts, each of length b or less. The following upper bound for $S(n,b,\lambda)$ follows directly from (3.2) :

$$S(n, b, \lambda) \leq q^{n - 2b\lambda} \tag{3.3}$$

One of the most powerful of the known classes of multiple-burst-errors-correcting-codes is the Reed-Solomon code [8, p. 370-371]. This code uses a so called Galois Field (GF(q)) and is designed to correct up to t errors in GF(q) where each t consist of exactly m bits. Consider a t-error-correcting Reed-Solomon code with code symbols from GF(q) with $q = 2^m$. We get a binary linear code with the following parameters :

$$n = m\ (2^m - 1) \tag{3.4}$$

$$n - k = 2mt \tag{3.5}$$

This code is capable to correct any error pattern that affects t or fewer m-bit symbols. Binary codes derived from Reed-Solomon codes are more effective against burst errors than random errors because burst errors usually involve several bit-errors per symbol and thus relatively few symbol-errors. In general, a t-symbol-correcting binary Reed-Solomon code is capable to correct any combination of :

$$\lambda = \left\lfloor \frac{t}{1 + \lfloor (b + m - 2)/m \rfloor} \right\rfloor \tag{3.6}$$

or fewer bursts (each of length b), or correcting any single burst of maximum length $m(t-1)+1$. It can also correct any combination of t or fewer random errors. For details see [7].

3.2 Localized burst error correcting codes

There are only a few known codes for localized burst errors correction. We will discuss two of such codes namely Kaag code (was found by Guus Kaag, see [3]) and Larsson code (was found by Per Larsson, see [2]).

First, we will derive the upper bound for multiple localized burst errors correcting codes. Denote by $B_M(n,b,\lambda)$ the maximum size of a code of length n that can correct λ localized bursts, each of length b or less, over an alphabet of size q. A trivial upper bound is given by the following inequality,

$$B_M(n, b, \lambda) \leq q^{n - b\lambda} \tag{3.7}$$

The stronger upper bound can be found in [2]. However, the upper bound will primarily be used to show the difference in coderate between the upper bound and the new construction code. In this case, (3.7) is sufficient.

3.2.1 Kaag code

The Kaag code is especially designed for correcting single localized burst errors. Like every localized burst error correcting codes, Kaag code uses also part of the codeword to inform the decoder where a burst may be located. We present one construction method where the encoder, using only two symbols, can tell the decoder where the information about the possible location of the burst is located.

First we will divide the codeword of length n into q parts. Part I consists of the first $\lfloor (n-b)/q \rfloor$ symbols plus the last b symbols. Each other part consists of at least $\lfloor (n-b)/q \rfloor$ successive symbols. Using up all symbols we have to add one more symbol to each of $((n-b)-q \cdot \lfloor (n-b)/q \rfloor)$ parts after Part I. Denote by A_1 the first symbol in a codeword while by A_4 the last symbol. Let the r symbols after A_1 be denoted by A_2 and the last r symbols of Part q by A_3.

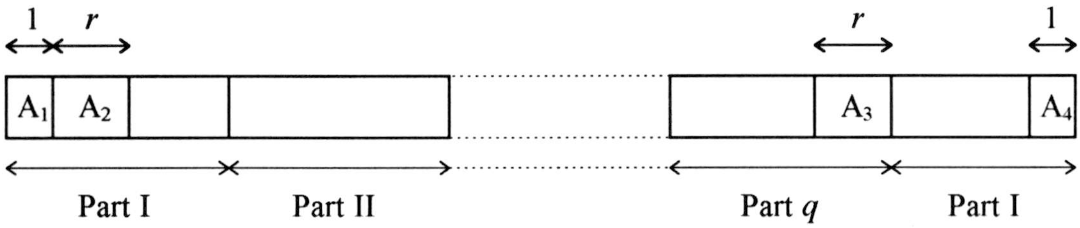

Figure 3.1 : The layout of a codeword using Kaag code.

The combination of A_1 and A_4 are used to tell the decoder which part of the codeword (A_2 or A_3) contains the information about the location of the burst. If the burst occurs in Part I, the information about the location of the burst is transported in A_3. If the burst takes place in another part, that information will be located in A_2.

There are $n-b+1$ possible positions for a burst of length b to start in a codeword of length n. The r symbols in A_2 or A_3 must be able to code those possible positions. Therefore r is chosen as follows,

$$r = \left\lceil \log_q \left\lceil \frac{(n-b+1)}{q} \right\rceil \right\rceil \tag{3.8}$$

A restriction that must be made is that both areas A_1 and A_2 must be in the first $\lfloor (n-b)/q \rfloor$ positions. That means that the following equation must hold :

$$r + 1 \leq \left\lfloor \frac{n-b}{q} \right\rfloor \tag{3.9}$$

The maximum allowable burstlength (b_{max}) is given by :

$$b_{max} \approx \frac{q-1}{2q-1} \cdot n \tag{3.10}$$

For binary case ($q = 2$) this means that a burst of about one third of the length of the codeword can be corrected. Larger values of q give a larger b_{max}, it goes to half the length of n.

From n symbols in one codeword, b symbols are assumed to be defected by the burst, one symbol is used both in A_1 and A_4 while r symbols are used in A_2 or A_3. The size of Kaag code can now be derived as follows,

$$q^{n-(b+2+r)} \tag{3.11}$$

3.2.2 Larsson code

The Larsson code is one of the first codes designed to correct (multiple) localized burst errors. As was mentioned in Chapter 2, the encoder is assumed to have the ability of knowing where all λ bursts may occur, i.e. the encoder knows a set of exactly λ disjoint bursts, each of b consecutive positions. The Larsson code is designed to correct up to λ

bursts, each of length b or less. Denote by $T(n,\lambda,b)$ the number of configurations with precisely λ disjoint blocks of b adjacent positions in a block of length n. We have,

$$T(n,\lambda,b) = \binom{n-\lambda b+\lambda}{\lambda} \tag{3.12}$$

A proof is given in [2, p.72]. The idea is to use only $\lambda+1$ sub-blocks (each of length r) from a total number of $2\lambda+1$ sub-blocks, separated as much as possible in a codeword of length n. In each of $\lambda+1$ sub-blocks, the encoder will transmit the first positions of all λ disjoint bursts (i.e. in each sub-block one of $T(n,\lambda,b)$ message is transmitted).

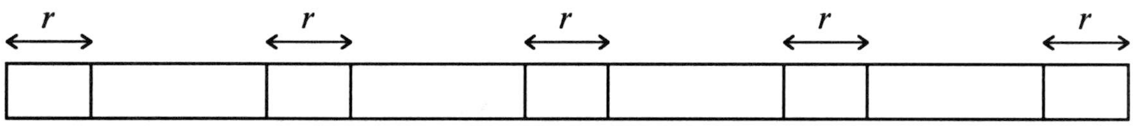

Figure 3.2 : The layout of a codeword using Larsson code with $\lambda = 2$.

Because there are at most λ defected sub-blocks in each codeword, the decoder can obtain the locations of the λ possible bursts by reading the content of all $2\lambda+1$ sub-blocks and makes a majority decision. r is chosen as follows,

$$r = \left\lceil \log_q T(n,\lambda,b) \right\rceil \tag{3.13}$$

It is assumed that b is limited in such a way that the decoder will find $\lambda + 1$ error-free sub-blocks. Because all sub-blocks are separated as much as possible, there will be at least

$$\left\lfloor \frac{n - (2\lambda + 1)r}{2\lambda} \right\rfloor \tag{3.14}$$

positions between any two sub-blocks. Therefore the maximum burstlength (b_{max}) must not exceed (3.14) by more than one, i.e. :

$$b_{max} = \left\lfloor \frac{n - (2\lambda + 1)r}{2\lambda} \right\rfloor + 1 \tag{3.15}$$

If we consider n symbols in one codeword, (λb) symbols will be defected by λ bursts and a total of $(\lambda + 1)r$ symbols are used in the first $(\lambda + 1)$ error-free sub-blocks to inform the decoder about the locations of the bursts. The size of Larsson code becomes :

$$q^{n - \lambda b - (\lambda + 1)r} \tag{3.16}$$

A slightly different construction for Larsson code is defined for single localized burst $(\lambda = 1)$. Although this code produces a larger size than (3.16), it was proven in [3] that Kaag code (especially designed for single localized burst) is always better than that code.

Chapter 4

THE NEW CONSTRUCTION (CONSTRUCTION A)

By Larsson code described in Section 3.2.2, the idea is to use $2\lambda+1$ sub-blocks / parts of the codeword each of length *r*, separated as much as possible to inform the decoder where the locations of λ bursts may be found. It is assumed that the decoder will receive minimum $\lambda+1$ error-free sub-blocks. This means that those $\lambda+1$ sub-blocks must contain the same information about the location of λ bursts.

Construction A

The new construction (Construction A) is designed to send that information only one time. A new rule, called the FBI rule, is used for this purpose. Like by Kaag code and Larsson code, we assume a fixed burstlength (b) for each codeword. This means that λb transmitted symbols will always be discarded even if no burst occurs, or when the total number of defected symbols are fewer than λb.

4.1 Definitions

Construction A is designed to correct up to λ localized bursts, each of length b or less. First, we will divide the codeword of length n into p as equal as possible parts,

$$p = \begin{cases} 3, & \text{for } \lambda = 1,2 \\ \lambda, & \text{for } \lambda > 2 \end{cases}$$

so that each part consists minimum $\lfloor n/p \rfloor$ symbols and maximum $\lceil n/p \rceil$ symbols. After that, we also have to divide each part into λ as equal as possible sub-parts so that each sub-part has a minimum length of $\lfloor n/p\lambda \rfloor$ symbols and a maximum length of $\lceil n/p\lambda \rceil$ symbols. All sub-parts with smaller length will be placed in front of the larger sub-parts. The situation is illustrated in Figure 4.1. From now on, each sub-part will be called a block. Denote by P_i ($i = 0, 1, 2,..., p-1$) the consecutive parts in one codeword and by M_j ($j = 0, 1, 2,...,\lambda-1$) the consecutive sub-parts / blocks in one part.

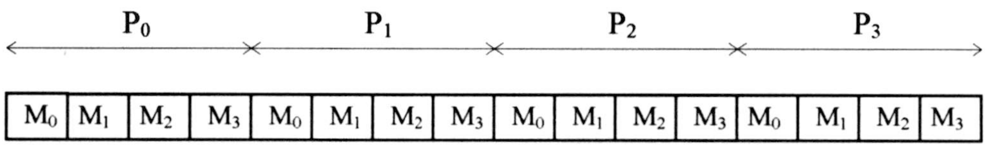

Figure 4.1 : The layout of one codeword from Construction A for $\lambda = 4$.

4.2 FBI rule

FBI rule is specially designed for the encoder to inform the decoder which block in the codeword contains the information about the location of possible bursts. In fact, the FBI rule is used to indicate whether one block in a codeword will be define as the F(fixed)-block, B(burst)-block or I(information)-block (that's why we call it as the FBI rule). The use of the F,B,I-blocks will be explained in the next section. From $p\lambda$ blocks (M_j's) in one codeword, λ blocks will be define as the B-blocks, another λ blocks as the I-blocks and the remaining $(p-2)\cdot\lambda$ blocks as the F-blocks. This implies that at the end of the encoding procedure, all blocks in a codeword will always be classified into three different types of blocks (F, B, I-blocks).

4.2.1 Encoding

As mentioned before, the encoder knows where all λ possible bursts may occur. The encoder has to inform the decoder about those locations. The FBI rule is used for this purpose. First, the encoder must determine which blocks will be define as the B-blocks. Once all B-blocks are discovered, the encoder is able to assign a number of $(p-2)\cdot\lambda$ blocks as the F-blocks and the remaining λ blocks as the I-blocks. The procedure to determine one block as the B-block will now be given.

For each burst, we have two possibilities (see Figure 4.1):

1) The burst starts and ends in M_j :
   - If M_j is not yet used as the B-block, we use M_j as the B-block.
   - If M_j is already used as the B-block, we use the first unused block as the B-block.

2) The burst starts from M_j and ends in $M_{j+1(\mathrm{mod}\ \lambda)}$:
   - If $M_{j+1(\mathrm{mod}\ \lambda)}$ is not yet used as the B-block, we use $M_{j+1(\mathrm{mod}\ \lambda)}$ as the B-block.
   - If $M_{j+1(\mathrm{mod}\ \lambda)}$ is already used as the B-block, we use the first unused block as the B-block.

We use this procedure for the first burst and repeat it until all burst are investigated. At the end, we should get exactly λ B-blocks.

After discovering all B-blocks, the encoder is now ready to determine the F and the I-blocks. We consider three possible situations :

1) All B-blocks appear together only in one part (all M_j's in that part are the B-blocks), say in P_i. In this case we use every block within $P_{i+1(\mathrm{mod}\ p)}$ as I-block and the remaining blocks as F-blocks.

F	F	F	F	B	B	B	B	I	I	I	I	F	F	F	F

I	I	I	I	F	F	F	F	F	F	F	F	B	B	B	B

Figure 4.2 : Two examples of situation 1 for $\lambda = 4$.

2) The B-blocks appear in more than one part but not in all parts. It means that there is a minimum of one part that has no B-block within that particular part, so called B-block-free part. Therefore, we use every block within the first B-block-free part as I-block and the remaining blocks in the codeword as F-blocks.

B	B	F	B	I	I	I	I	F	F	B	F	F	F	F	F

I	I	I	I	F	F	B	B	F	F	F	F	B	B	F	F

F	B	F	F	F	F	B	F	I	I	I	I	F	B	F	B

Figure 4.3 : Three examples of situation 2 for $\lambda = 4$.

3) The B-blocks appear in all parts. In this case, each part contains exactly one B-block (only one M_j in each part is B-block). For each part we determine where the B-block is, say in M_j. We use the $M_{j+1(\text{mod } \lambda)}$ block in that part as I-block. Any other blocks in the codeword are reserved for F-blocks.

B	I	F	F	F	B	I	F	F	F	B	I	I	F	F	B

Figure 4.4 : One example of situation 3 for $\lambda = 4$.

The encoder has now a complete formation of the F,B,I-blocks. We choose one symbol from the alphabet and denote it by the fixed symbol. We put this fixed symbol in the first position of all F-blocks. In every I-block, we put one symbol (except the fixed symbol) in the first position and any symbol in r consecutive positions started from the second position. The combination of the first $(r+1)$ symbols of all I-blocks will give the locations of all bursts (how to fill these r positions will be described in Section 4.3). Finally, the actual message can be placed in the remaining positions of the codeword (except the positions of possible bursts).

4.2.2 Decoding

The primary task of the decoder is to find where λ bursts are located. The decoder knows that the information about those locations can be found in certain blocks (I-blocks) in a received vector. A good method is then needed to ensure that the decoder "looks" at the right blocks. Once all I-blocks are determined, the decoder is able to find the actual location of all bursts.

The first step in the decoding procedure is to find all F-blocks and memorize in which part they occur. Because the decoder knows that each F-block always begins with one fixed symbol, all it has to do is to scan every first symbol of all blocks (all M_j's) of the received vector. The very important thing, at this point, is that since the decoder does not know the locations of the bursts yet, the decoder can not recognize the B-block, so every block starts with fixed symbol will be automatically defined as the F-block. We are now dealing with three possible situations :

1) There is only one part that does not contains F-block. In this case, we are sure that that particular part contains all λ I-blocks.

	F			F	F	F		I	I	I	I	F	F	F	F

I	I	I	I	F	F	F		F	F	F	F			F	F

Figure 4.5 : Two possible received vectors of situation 1 for λ = 4.

2) Two parts do not contain any F-block. If they are two adjacent parts, the decoder knows that all blocks within the last part are the I-blocks. If they are not two adjacent parts, P_0 would be the part that contents all I-blocks.

F	F	F	F					I	I	I	I	F	F	F	F

I	I	I	I	F	F	F	F	F	F	F	F				

Figure 4.6 : Two possible received vectors of situation 2 for $\lambda = 4$.

3) All parts contain F-block. We investigate each part (P_i, for $i = 0,1,2,\dots p-1$) and discuss two different possibilities :
   a) There is only one block left outside the F-blocks which has to be the I-block.
   b) There are two blocks left outside the F-blocks. If they are two adjacent blocks, the last block must be the I-block, if not, M_0 is the I-block.

I	F	F	F	F	I	F	F	F		I	F	I	F	F	

Figure 4.7 : One possible received vector of situation 3 for $\lambda = 4$.

It is clear that the main purpose of the decoding procedure is to find all I-blocks in a received vector, because those blocks contain the information about the locations of the bursts. Once the bursts are located, the decoder is able to discard all symbols which are defected by the bursts. The transmitted message can then be found in the rest of the received vector.

4.3 The I-block

As described in the Section 4.2.1, the encoder has to fill all I-blocks with the information about the possible locations of the bursts. Let T(n,λ,b) be the number of all configurations with exactly λ disjoint bursts, each of b consecutive symbols in a codeword of length n. From (3.12) we get,

$$\mathrm{T}(n,\lambda,b)=\binom{n-\lambda b+\lambda}{\lambda} \tag{4.1}$$

At the moment that the decoder can identify all λ I-blocks, there is a possibility that the decoder still does not know the right locations of the original F-blocks and B-blocks. Nevertheless, the decoder must now determine where the bursts are. One important thing is that the decoder is convinced that no burst can end in those λ I-blocks (each with minimum length of $\lfloor n/p\lambda \rfloor$ symbols). That is the reason why we can replace n in (4.1) with $n-\lfloor n/p\lambda \rfloor\cdot\lambda$. Therefore, T($n$,$\lambda$,$b$) is reduced and becomes A(n,λ,b), where :

$$\mathrm{A}(n,\lambda,b)=\binom{n-\lfloor n/p\lambda \rfloor\cdot\lambda-\lambda b+\lambda}{\lambda}$$

or

$$\mathrm{A}(n,\lambda,b)=\binom{n-\lambda\cdot\left(\lfloor n/p\lambda \rfloor+b-1\right)}{\lambda} \tag{4.2}$$

From q symbols in an alphabet, one symbol is chosen as a fixed symbol (used in the first position of F-block), leaving $(q-1)$ possible symbols for the first position of I-block. This limitation is valid only for the first position of I-block. It means that on the second (and other) position, the encoder is free to use all q symbols of the alphabet.

The encoder uses the combinations of *all* $(r+1)$ positions of *all* λ I-blocks to determine the locations of λ bursts. Therefore, r is chosen as follows :

$$(q-1)^{\lambda} \cdot q^{r\lambda} \geq \mathrm{A}(n,\lambda,b) \tag{4.3}$$

or

$$r = \left\lceil \frac{\log \mathrm{A}(n,\lambda,b) - \lambda \cdot \log (q-1)}{\lambda \cdot \log q} \right\rceil \tag{4.4}$$

4.4 Maximum burstlength

To determine the maximum burstlength, we have to investigate the "worst" combination of the F,B,I-blocks in one codeword. That combination appears if one I-block is located directly in front of one B-block. In that case, there must be always at least :

$$\left\lfloor \frac{n}{p\lambda} \right\rfloor - (r+1) \tag{4.5}$$

remaining positions in that I-block to guarantee the error-free transmission, otherwise the information about the location of the bursts in that I-block could be defected by the bursts themselves. The maximum burstlength $b_{\max}$ is now derived as :

$$b_{\max} = \left\lfloor \frac{n}{p\lambda} \right\rfloor - r \tag{4.6}$$

4.5 The size of Construction A

The encoder has a total of $(q-1)^{\lambda} \cdot q^{\lambda r}$ combinations from the first $(r+1)$ positions of all I-blocks to cover a number of $A(n,\lambda,b)$ configurations for possible locations of the bursts. Denote by x the number of sets that the encoder can choose from, with each set containing at least $A(n,\lambda,b)$ possible locations of all bursts. Hence we have,

$$x = \left\lfloor \frac{(q-1)^{\lambda} \cdot q^{r\lambda}}{A(n,\lambda,b)} \right\rfloor \tag{4.7}$$

From n positions in one codeword, a number of λb positions are assumed being defected by the bursts, a total number of $(p-2)\cdot\lambda$ positions are used by the F-blocks and finally a total number of $(r+1)\cdot\lambda$ positions in the I-blocks. The remaining positions can be used for the message. The size of Construction A becomes :

$$x \cdot \left(q^{n-\lambda b-(p-2)\lambda-(r+1)\lambda}\right)$$

or

$$x \cdot \left(q^{n-(b+p+r-1)\lambda}\right) \tag{4.8}$$

4.6 Comparison of Construction A with the Larsson code

In this section, we will compare Construction A with the Larsson code (described in Section 3.2.2). Notice that Larsson code was constructed mainly for proving the asymptotic result [4]. The upper bound for multiple localized burst (3.7) and the upper bound for conventional multiple-burst-error-correcting-code (3.3) are also included. Figure 4.8, 4.9, 4.10 and 4.11 illustrate the comparison for different parameters.

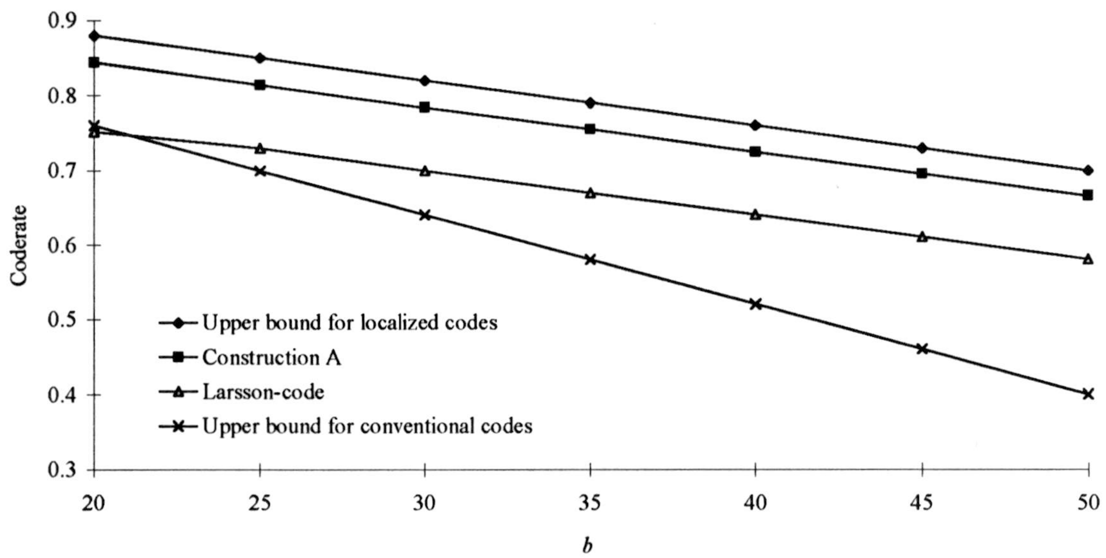

Figure 4.8 : Comparison between Construction A and Larsson code with the following parameters : $n = 500$, $q = 3$, $\lambda = 3$.

From Figure 4.8, it can be concluded that the difference in the coderate between Construction A and Larsson code becomes smaller if the burstlength increases. Using (3.15) and (4.6) we can see that Construction A has a smaller maximum burstlength than Larsson code.

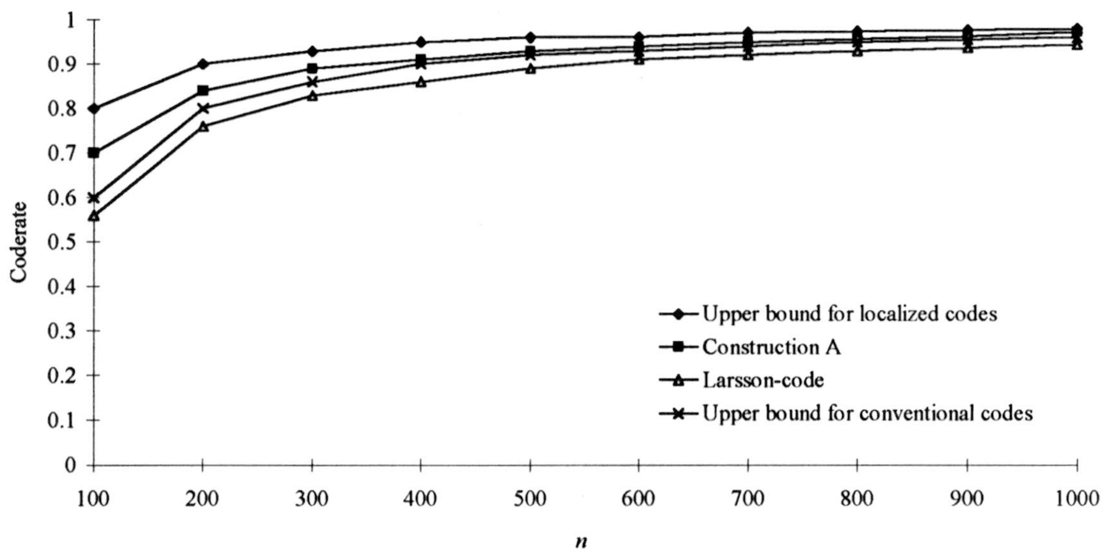

Figure 4.9 : Comparison between Construction A and Larsson code with the following parameters : $q = 3$, $\lambda = 2$, $b = 10$.

Figure 4.9 shows that if n goes to infinity, the difference in coderate between Construction A and Larsson code is negligible. This is understandable because for a large value of n, the total number of redundancy symbols (used in F and I-blocks of Construction A or in $(\lambda+1)$ sub-blocks of Larsson code) are very small compare to n.

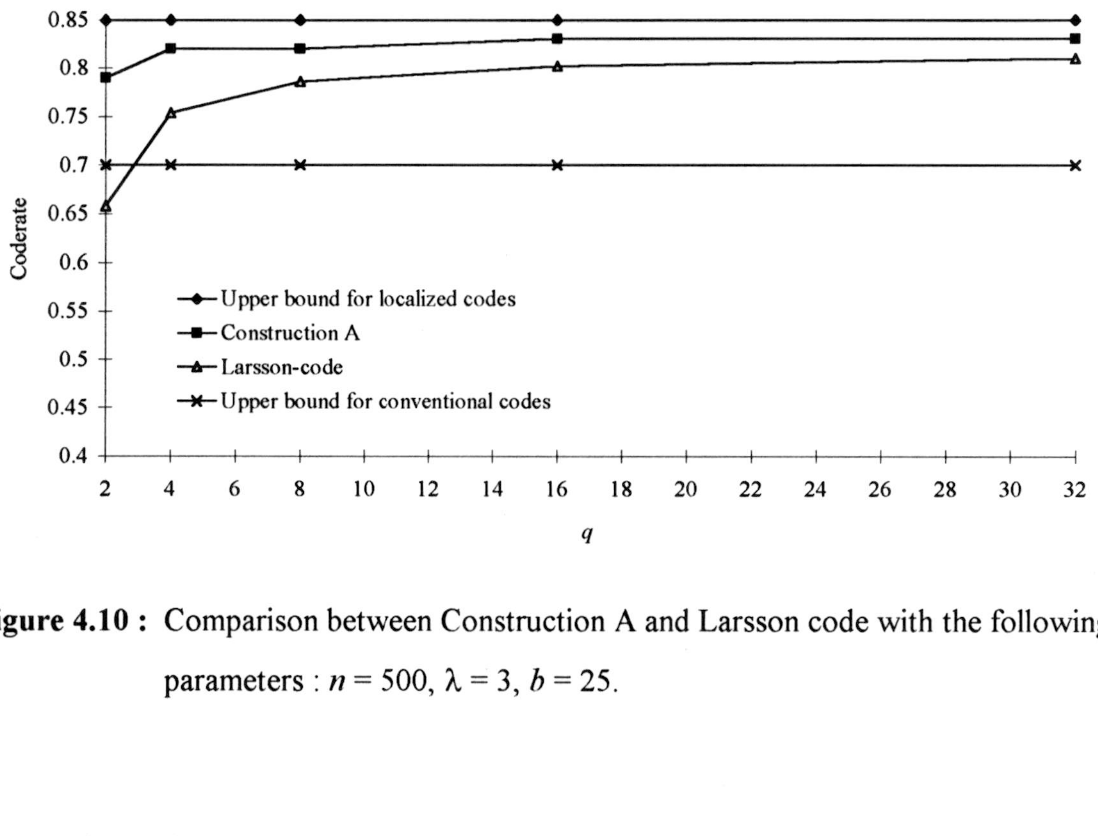

Figure 4.10 : Comparison between Construction A and Larsson code with the following parameters : $n = 500$, $\lambda = 3$, $b = 25$.

We see from Figure 4.10 that for relative small value of q, Construction A produces a much higher coderate than Larsson code. This advantage will disappear if the size of alphabet increases since the length of one sub-block of Larsson code is reduced for a large value of q. Notice that both upper bounds have a constant coderate.

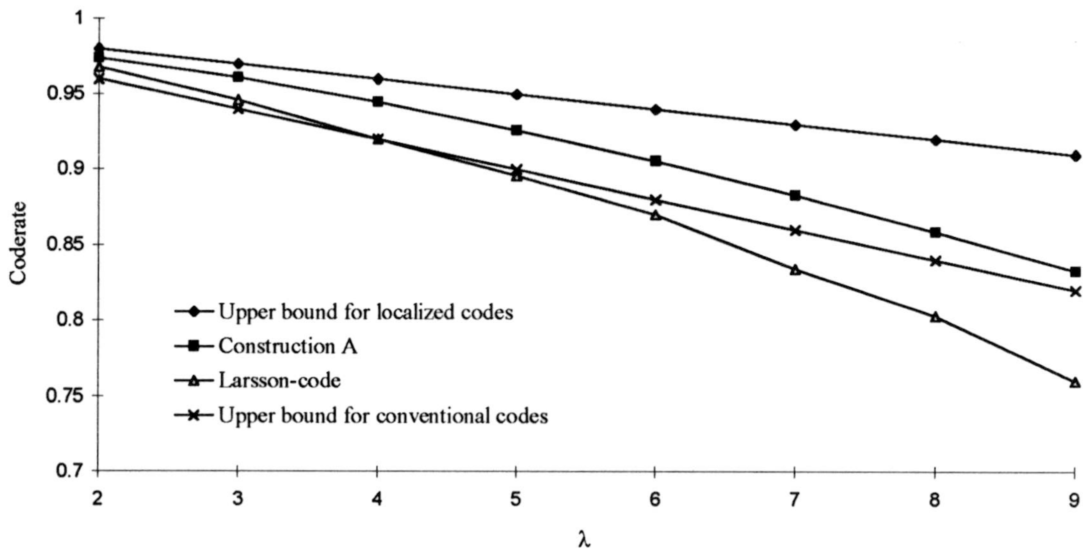

Figure 4.11 : Comparison between Construction A and Larsson code with the following parameters : $n = 1000$, $q = 32$, $b = 10$.

Figure 4.11 shows that the difference in coderate between Construction A and Larsson code becomes significant, especially for larger values of λ.The main reason is that the total number of redundancy symbols used in Larsson code is increased.

Chapter 5

SPECIAL CONSTRUCTION FOR SINGLE LOCALIZED BURST ERROR

A special construction (Construction B) is designed for correction of a single localized burst error. The basis of this construction is still Construction A (described in the previous chapter). The encoding and decoding procedure are the same as mentioned in Chapter 4. In the last section, a comparison with Kaag code that is especially designed for a single localized burst, will be given.

5.1 Construction B

First we will also divide the codeword of length n into three equal parts and denote them from left to right by P_i ($i = 0,1,2$). Each P_i has a minimum length of $\lfloor n/3 \rfloor$ symbols and a maximum length of $\lceil n/3 \rceil$ symbols.

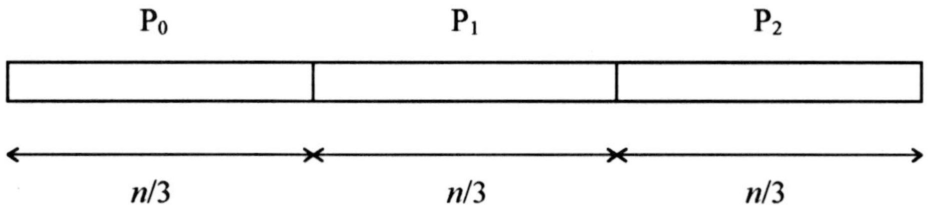

Figure 5.1 : Codeword for single localised burst error ($p = 3$, $\lambda = 1$).

There is only one possible burst and automatically only one B-block in the codeword. The encoder can always determine in which part that B-block is, so the number of possible starting positions of the burst becomes :

$$A(n,\lambda,b) = \left\lceil \frac{n}{3} \right\rceil \qquad (5.1)$$

From the inequality (4.3) follows :

$$(q-1)\cdot q^{r} \geq \left\lceil \frac{n}{3} \right\rceil \qquad (5.2)$$

or

$$r = \left\lceil \frac{\log\left\lceil \frac{n}{3} \right\rceil - \log(q-1)}{\log q} \right\rceil \qquad (5.3)$$

Since the encoder can always put the I-block after (mod 3) the B block (see Section encoding), the maximum burstlength b_{max} becomes :

$$b_{max} = \left\lfloor \frac{n}{3} \right\rfloor \qquad (5.4)$$

The size of Construction B is given by (4.8) :

$$x \cdot \left(q^{n-b-r-2}\right) \tag{5.5}$$

where

$$x = \left\lfloor \frac{(q-1)\cdot q^{r}}{\left\lceil \frac{n}{3} \right\rceil} \right\rfloor \tag{5.6}$$

As an example we will consider the case of $n = 20$, $b = 5$ and $q = 2$. The codeword is divided into three parts which is illustrated in Figure 5.2.

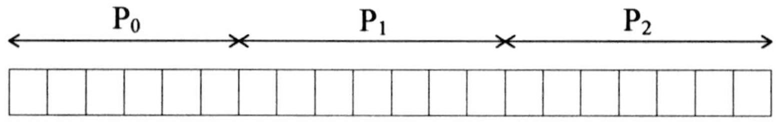

Figure 5.2 : A codeword for $n = 20$.

Using (5.3) we get :

$$r = \left\lceil \frac{\log \left\lceil \frac{20}{3} \right\rceil - \log\,(2-1)}{\log 2} \right\rceil = 3$$

From (5.4) we see that the maximum burstlength is 6 bits, so error free transmission is indeed possible. The size of the code is given by (5.5) and (5.6). This means that there are maximum 10 bits available for the message. Remember that the combination of the first $(r+1)$ bits of the I-block gives the position where the burst ends in the B-block. We choose (for example) 1 as the fixed symbol and for the first $(r+1)$ bits of the I-block (to indicate where the burst ends) : 0001 => 1, 0010 => 2, 0011 => 3, ... 0111 => 7

There are three different situations for the encoding (M for message, X for burst-error) :

1) P_0 is the B-block.

B						I							F						
X	X	X	X	X	M	0	1	0	1	M	M	M	1	M	M	M	M	M	M

2) P_1 is the B-block.

F						B							I						
1	M	M	M	X	X	X	X	X	M	M	M	M	0	0	1	1	M	M	M

1	M	M	M	M	M	M	M	X	X	X	X	X	0	1	1	1	M	M	M

3) P_2 is the B-block.

I						F							B						
0	1	0	0	M	M	1	M	M	M	M	M	X	X	X	X	X	M	M	M

0	1	1	1	M	M	1	M	M	M	M	M	M	M	M	X	X	X	X	X

The decoding procedure is as follows : the decoder scans every first bit of all P_i's (i = 0,1,2) of the received vector. There are two possible situations :

1) Two P_i's begin with 1 (fixed symbol).

   Denote by P_a the block that does not start with 1. The decoder can easily determine the I-block (P_a), the B-block ($P_{a-1 \pmod 3}$) and the F-block ($P_{a-2 \pmod 3}$).

2) Only one P_i starts with 1.

   Denote by P_b the block that begins with 1. In this case, P_b is automatically the F-block, $P_{b+1 \pmod 3}$ is the B-block while $P_{b+2 \pmod 3}$ is the I-block.

Once the F,B,I-blocks are determined, the decoder has to verify the contents of the first $(r+1)$ bits of I-block, locate where the burst ends and discard all bits defected by the burst. The rest of the received vector now contains the transmitted message.

5.2 Comparison with other code

We will compare Construction B with Kaag code (described in section 3.2.1). Although Larsson code can also be used for correcting single localized burst, it is proven in [3] that the Kaag code always produces a higher coderate. The upper bound for localized burst (3.7) is also included.

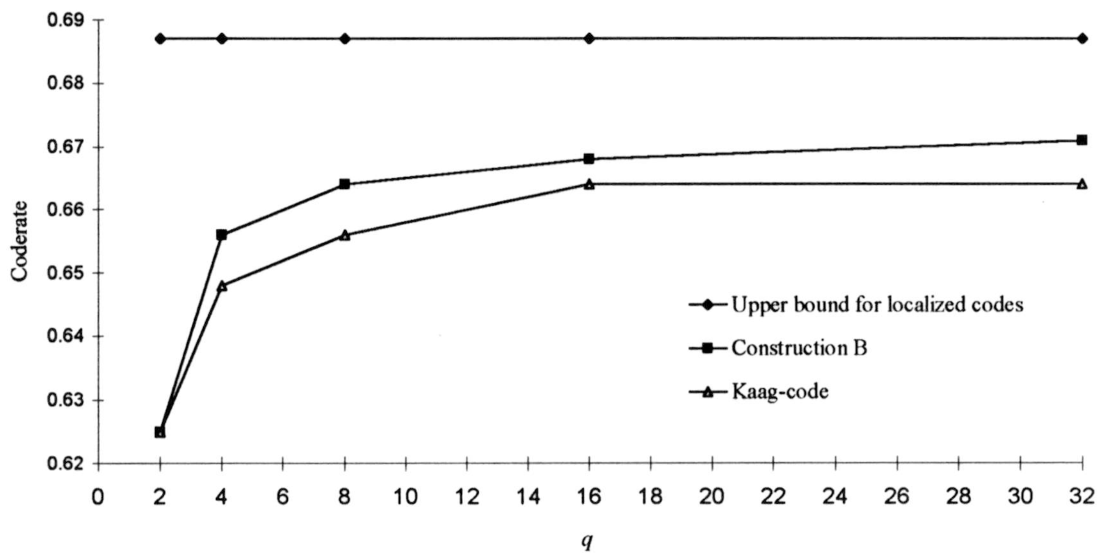

Figure 5.3 : Comparison between Construction B with Kaag code with the following parameters : $n = 128$, $b = 40$.

Figure 5.3 shows that Construction B produces a higher coderate than the Kaag code. In fact, for any given parameters, the size of Construction B cannot be smaller than Kaag code. A proof is given in Appendix A.

Chapter 6

MULTIPLE PARTIALLY LOCALIZED BURST ERRORS

6.1 The concept

The concept of partially localized errors was introduced in [9]. In this chapter we consider the corresponding case for multiple bursts. The assumption is that the encoder knows parts of the codeword of length n where bursts may occur. These parts are called as the unreliable parts.

Denote by G the set of α unreliable parts in a codeword, each of length g. Denote by B the set of λ bursts in a codeword, each of length b. It is assumed that the encoder and the decoder know α, g, λ, and b. G is known only to the encoder while B is not known neither to the encoder nor to the decoder. Although this will not necessarily happen, in this model we assume that there must be at least one burst in each unreliable part and the total number of defected symbols in that part must not exceed g. The following relations hold, $0 \leq b \leq g \leq n$ and $0 \leq \alpha \leq \lambda$.

There are three different possibilities for α :

1) $\alpha = 1$, so all λ bursts are located inside one unreliable part.

   The case $g = n$ corresponds to the case of conventional burst-error-correcting-codes.

   Example : $n = 100$, $\alpha = 1$, $g = 20$, $\lambda = 3$, $b = 3$.

   $G = \{(11,12,...,30)\}$.

   $B = \{(12,13,14);(17,18,19);(23,24,25)\}$.

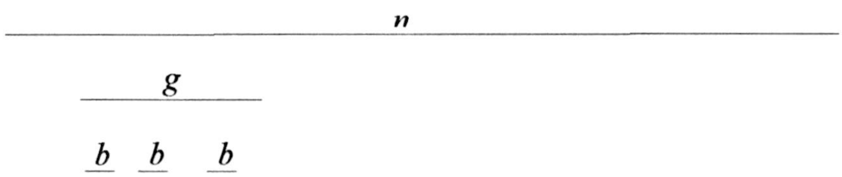

Figure 6.1 : An example of partially localized burst errors with $\alpha = 1$.

2) $1 < \alpha < \lambda$.

   This means that at least one unreliable part contains more than one burst.

   Example : $n = 100$, $\alpha = 2$, $g = 10$, $\lambda = 3$, $b = 3$.

   $G = \{(11,...,20);(71,...,80)\}$

   $B = \{(12,13,14);(17,18,19);(75,76,77)\}$

Figure 6.2 : An example of multiple partially localized burst errors with $1 < \alpha < \lambda$.

3) $\alpha = \lambda$, so each unreliable part contains exactly one burst.

The case $g = b$ corresponds to the case of multiple localized burst errors.

Example : $n = 100$, $\alpha = 3$, $g = 10$, $\lambda = 3$, $b = 3$.

G = {(11,...,20);(51,...,60);(91,...,100)}.

B = {(12,13,14);(57,58,59);(95,96,97)}.

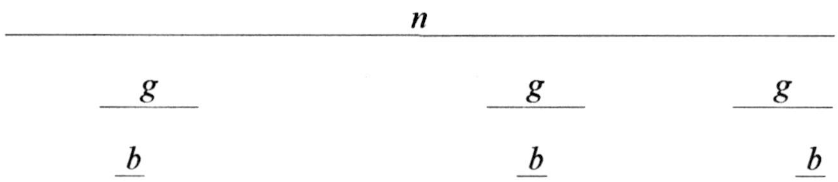

Figure 6.3 : An example of multiple partially localized burst errors with $\alpha = \lambda$.

Let λ_{max} be the maximum number of bursts that can occur in one unreliable part. λ_{max} can be determined when each of $(\alpha - 1)$ unreliable parts contains exactly one burst. Therefore, the only remaining unreliable part has to contain precisely $\lambda - (\alpha - 1)$ bursts. Hence we have,

$$\lambda_{max} = \lambda - (\alpha - 1) \tag{6.1}$$

6.2 Construction of the code

The code is designed to correct λ bursts each of length b in α unreliable parts each of length g. The idea is to encode the information about the location of all unreliable parts in a codeword. Construction A introduced in Chapter 4 is used for this purpose. We have to replace λ with α and b with g while the other parameters remain unchanged. For α = 1, Construction B introduced in Chapter 5 will produce a better performance.

Construction C

Let $G(n,\alpha,g)$ the number of all configurations with exactly α unreliable parts, each of g consecutive positions in a codeword of length n. From (5.1) or (4.2) we get :

$\alpha = 1$: $$G(n,\alpha,g) = \left\lceil \frac{n}{3} \right\rceil \qquad (6.2)$$

$\alpha > 1$: $$G(n,\alpha,g) = \binom{n-\alpha\cdot\left(\lfloor n/p\alpha\rfloor + g - 1\right)}{\alpha} \qquad (6.3)$$

The code uses the combination of all $(r+1)$ positions of all I-blocks to determine the location of all unreliable parts. Therefore r is chosen from (4.4) or (5.3) as follows :

$$r = \left\lceil \frac{\log G(n,\alpha,g) - \alpha\cdot\log(q-1)}{\alpha\cdot\log q} \right\rceil \qquad (6.4)$$

Using (5.4) or (4.6) we get the maximum length of one unreliable part $g_{\max}$:

$\alpha = 1$: $$g_{\max} = \left\lfloor \frac{n}{3} \right\rfloor \qquad (6.5)$$

$\alpha > 1$: $$g_{\max} = \left\lfloor \frac{n}{p\alpha} \right\rfloor - r \qquad (6.6)$$

From (4.7) or (5.5) we can also get an extra advantage in the parameter x :

$$x = \left\lfloor \frac{(q-1)^{\alpha} \cdot q^{\alpha r}}{\mathrm{G}(n,\alpha,g)} \right\rfloor \tag{6.7}$$

In each unreliable part, we apply the best conventional λ_{max} burst-error-correcting-code, i.e. a code that can correct up to λ_{max} burst errors. Denote by g_{min} the minimum length of that code and by $\mathrm{S}(g,b,\lambda_{max})$ the size of a conventional code of length g that can correct λ_{max} bursts, each of length b or less (see Section 3.1). The size of Construction C is now derived as :

$$x \cdot \left(q^{n-(g+p+r-1)\alpha}\right) \cdot \left(\mathrm{S}(g, b, \lambda_{max})\right)^{\alpha} \tag{6.8}$$

6.3 Comparison with other code

We will now compare Construction C with the best conventional multiple-burst-error-correcting-code, namely the Reed-Solomon code (described in Section 3.1). The parameters are $n = 2040$, $q = 2$, $\lambda = 3$, $b = 20$.

<u>Reed-Solomon code</u> :

Substituting n into (3.4) and (3.6) yields $m = 8$ and $t = 12$. Using (3.5) we have a coderate ≈ 0.9058.

Construction C :

We will also use a Reed-Solomon code in every unreliable part to correct the possible burst(s) in that part.

1) $\alpha = 1$.

Using (6.1) we have $\lambda_{max} = 3$. Two situations are considered :

1) $g = 155$. In an unreliable part we use the Reed-Solomon code with the following parameters : $n' = g_{min} = 155$, $m = 5$, S(155,20,3) = 5. Substituting all parameters into (6.2), (6.4), (6.5), (6.7) and (6.8) yields $g_{max} = 680$ and a coderate ≈ 0.9205.
2) $g = 680$. In an unreliable part we use the Reed-Solomon code with the following parameters : $n' = 680$, $m = 7$, S(680,20,3) = 511. Substituting all parameters into (6.2), (6.4), (6.5), (6.7) and (6.8) yields $g_{max} = 680$ and a coderate ≈ 0.9112.

The situation is illustrated in Figure 6.4.

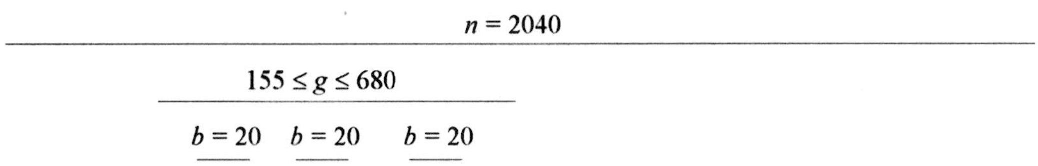

Figure 6.4 : Construction C with $n = 2040$, $q = 2$, $\lambda = 3$, $b = 20$, $\alpha = 1$.

2) $\alpha = 2$.

Using (6.1) we have $\lambda_{max} = 2$. Two situations are considered :

1) g = 105. In each unreliable part we use the Reed-Solomon code with the following parameters : $n' = g_{min} = 105$, $m = 5$, S(105,20,2) = 5. Substituting all parameters into (6.3), (6.4), (6.6), (6.7) and (6.8) yields g_{max} = 330 and a coderate ≈ 0.8901.

2) g = 330. In each unreliable part we use the Reed-Solomon code with the following parameters : n' = 330, m = 5, S(330,20,2) = 230. Substituting all parameters into (6.3), (6.4), (6.6), (6.7) and (6.8) yields g_{max} = 330 and a coderate ≈ 0.8911.

Figure 6.5 illustrates the situation.

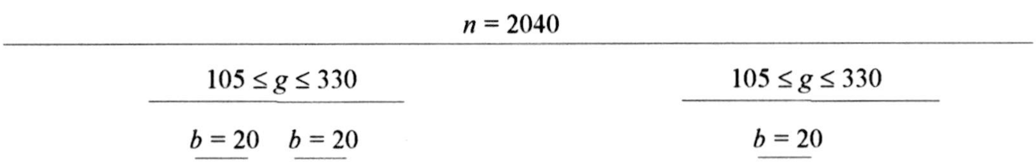

Figure 6.5 : Construction C with $n = 2040$, $q = 2$, $\lambda = 3$, $b = 20$, $\alpha = 2$.

3) $\alpha = 3$.

Using (6.1) we have $\lambda_{max} = 1$. Two situations are considered :

1) $g = 52$. In each unreliable part we use the Reed-Solomon code with the following parameters : $n' = g_{min} = 52$, $m = 4$, S(52,20,1) = 4. Substituting all parameters into (6.3), (6.4), (6.6), (6.7) and (6.8) yields $g_{max} = 216$ and a coderate ≈ 0.9125.

2) g = 216. In each unreliable part we use the Reed-Solomon code with the following parameters : n' = 216, m = 5, S(216,20,1) =165. Substituting all

parameters into (6.3), (6.4), (6.6), (6.7) and (6.8) yields $g_{\max} = 216$ and a coderate ≈ 0.9093.

See Figure 6.6 for illustration.

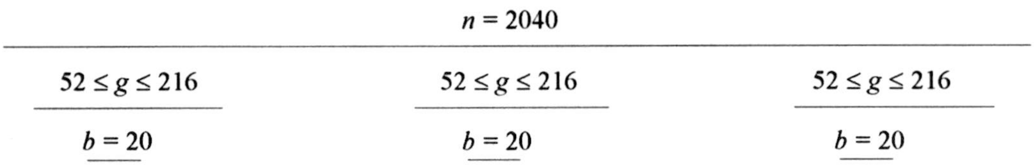

Figure 6.6 : Construction C with $n = 2040$, $q = 2$, $\lambda = 3$, $b = 20$, $\alpha = 3$.

It is remarkable that Construction C only produces a slightly higher (or even lower) coderate than Reed-Solomon code. The main reason is because in every unreliable part, we must use a conventional burst-error-correcting-code to correct a possible burst in that particular part. Another loss can be found in the extra symbols used to tell the decoder where the locations of the unreliable parts are.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

This thesis introduces a new construction method for correction of multiple (partially) localized burst errors. As mentioned in the first chapter, it is assumed that only the encoder knows the possible locations of the bursts. Any other positions outside those locations are then assumed error-free. The following conclusions can be made for three construction methods described in Chapter 4, 5 and 6.

Construction A :

From (3.3) and (4.8) we see that the coderate of Construction A will exceed the upper bound of the conventional multiple burst-errors-correcting-code if (let $x = 1$) :

$$b > p + r - 1 \qquad (7.1)$$

We must also remember that in most cases the upper bound can not always be achieved by any code at all.

The encoding and decoding procedures are constructed only from adding and stripping the check symbols, while the conventional codes uses a more complex mathematical calculation.

It is clear that Construction A always produces a higher coderate than the Larsson code. One important reason is because the information about the locations of the bursts is coded only once in a codeword of Construction A. We also have to remember that in some cases, Construction A has an extra advantage in parameter x, which Larsson code never has.

One disadvantage of Construction A is the reduce of the maximum burstlength, especially for larger values of λ. This can happen because the codeword is divided into λp blocks and one burst may not occupy more than the length of one block minus r positions, otherwise in case where one I-block is located directly in front of one B-block, the information about the locations of the bursts could be defected by the bursts themselves.

Construction B :

As mentioned in Section 5.2, we can conclude that the size of Construction B cannot be smaller than the size of Kaag code. A proof is given in Appendix A.

Like with Construction A, the maximum burstlength of Construction B is also reduced and becomes approximately one third of the codeword while Kaag code can still correct a burst of length 1/2 n if q is sufficient large (see (3.10)).

Construction C :

For 1< α < λ, Construction C will always yield a lower coderate, because in each unreliable part, we have to use a conventional multiple-burst-errors-correcting-code that can correct up to λ_{max} bursts. This implies that an extra loss is created in every unreliable part containing less than λ_{max} bursts.

Because Construction C uses a conventional multiple-burst-errors-correcting-code in every unreliable part, there is a possibility that no such code would exist for a given value of g, i.e. if $g < g_{min}$. Therefore, an extra disadvantage is introduced namely the minimum length of one unreliable part (g_{min}).

For smaller value of g_{min}, it may be worth considering to use Construction A or Construction B. The assumption that must be made is that all positions in every unreliable part are erroneous. Therefore, we have to neglect the parameter b and λ in Construction C. For Construction A or B we consider g as the burstlength and α as the number of bursts.

7.2 Recommendations

For further research, the following recommendations can be considered.

In this thesis we assume that there are no random errors in a codeword. This assumption is not always true, in fact in a certain application, random errors do always occur. Therefore, it would be useful to extend the error correcting capability of a code to a combination of localized burst errors and random errors.

Another assumption is that the burstlength b is always constant. This means that λb transmitted symbols will always be discarded even if no burst occurs, or when the total number of defected symbols are fewer than λb. If bursts do not likely occur in every codeword and/or if the burstlength is quite varying, the use of a code that describes not only the location of the burst but its length as well can be more favourable.

Although Construction A produces a higher coderate than Larsson code, its maximum burstlength is limited, especially for a large number of bursts. Therefore, it may be worth to consider a different construction for correcting a longer localized burst than the maximum correctable burst of Construction A.

The concept of partially localized burst errors introduced in Chapter 6 is the very first concept for multiple bursts. The result is not very satisfactory because the difference in coderate between Construction C and Reed-Solomon code is negligible. Therefore, a whole different concept that yields a better performance than Construction C would be appreciated. No extensive research on this subject has been done yet, so this is still an open area.

APPENDIX A

A COMPARISON BETWEEN CONSTRUCTION B AND KAAG CODE

A proof that the size of Construction B cannot be smaller than the size of Kaag code is now be given. Using (5.5) and (3.11) we get,

Construction B :

Size : $x \cdot q^{n-b-2-r_b}$

Kaag code :

Size : $q^{n-b-2-r_k}$

where

$$r_b = \left\lceil \frac{\log\left\lceil \frac{n}{3} \right\rceil - \log\ (q-1)}{\log q} \right\rceil \qquad r_k = \left\lceil \frac{\log\left\lceil \frac{n-b+1}{q} \right\rceil}{\log q} \right\rceil$$

Let $x = 1$ and

$$R_b = \frac{\log\left\lceil \frac{n}{3} \right\rceil - \log\ (q-1)}{\log\ q} \quad \text{and} \quad R_k = \frac{\log\left\lceil \frac{n-b+1}{q} \right\rceil}{\log\ q}$$

The following relation always holds,

$$R_b \le R_k \ \Rightarrow \ r_b \le r_k \tag{A.1}$$

Assume that $R_b \le R_k$, hence we have

$$\frac{\log\left\lceil \frac{n}{3} \right\rceil - \log\ (q-1)}{\log\ q} \le \frac{\log\left\lceil \frac{n-b+1}{q} \right\rceil}{\log q}$$

$$\left\lceil \frac{n}{3} \right\rceil \div (q-1) \le \left\lceil \frac{n-b+1}{q} \right\rceil \tag{A.2}$$

Since $\left\lceil \frac{n}{3} \right\rceil \div (q-1) \le \left\lceil \frac{n}{3\ (q-1)} \right\rceil$ and $\left\lceil \frac{n-b}{q} \right\rceil \le \left\lceil \frac{n-b+1}{q} \right\rceil$, the inequality (A.2) can be replaced with

$$\left\lceil \frac{n}{3\ (q-1)} \right\rceil \le \left\lceil \frac{n-b}{q} \right\rceil \tag{A.3}$$

We investigate the worst case for Construction B, i.e. a case where $b = b_{max} = \lfloor n/3 \rfloor$. Substituting b in (A.3) gives,

$$\left\lceil \frac{n}{3\,(q-1)} \right\rceil \quad \leq \quad \left\lceil \frac{n - \frac{1}{3}n}{q} \right\rceil$$

or

$$\left\lceil \frac{2\,n}{6\,(q-1)} \right\rceil \quad \leq \quad \left\lceil \frac{2\,n}{3\,q} \right\rceil \qquad \text{(A.4)}$$

The inequality (A.4) is always satisfied for $q \geq 2$. Using the relation (A.1), we can conclude that the size of Construction B cannot be smaller than the size of Kaag code. Remember that Construction B might also get an extra advantage in the parameter x, (see (5.6)) that Kaag code never has.

BIBLIOGRAPHY

[1] L.A. Bassalygo, S.I. Gelfand and M.S. Pinsker,
"Coding for Channels With Localized Errors",
Proc.Fourth Joint Swedish-Soviet Int. Workshop on Information Theory, pp.95-99, Gotland, Sweden, August 1989.

[2] P.Larsson,
Codes for Correction of Localized Errors,
Linköping Studies in Science and Technology, Dissertations No. 374, Sweden, April 1995.

[3] G.Kaag,
"A New Construction Method for Codes Correcting Single Localized Burst Errors",
Master Thesis TU-Delft, Telecommunications and Traffic Control Systems Group, August 1995.

[4] P.Larsson and J.H.Weber,
"Asymptotically Optimal Rate for Codes Correcting Partially Localized Burst Errors",
Proc.of the 1994 International Symposium on Information Theory, pp.250, Trondheim, Norway, June 1994.

[5] A.V. Kuznetsov and B.S. Tsybakov,
"Coding for memories with defects",
Probl. Inform. Transmission, Vol. 10(2), pp. 52-60, 1974.

[6] S.H. Reiger,
"Codes for the correction of clustered errors",
IRE, IT 6 : 16-21, 1960.

[7] S. Lin, D.J. Costello. Jr,
Error Control Coding : Fundamentals and Applications,
New Jersey: Prentice-Hall, New Jersey, 1983.

[8] W.W. Peterson, E.J Weldon. Jr,
Error-Correcting Codes, second edition.
The Colonial Press, 1972.

[9] L.A. Bassalygo, S.I. Gelfand and M.S. Pinsker,
"Coding for Partially Localized Errors",
IEEE Trans.Inform.Theory, Vol. 37(2), pp. 880-884, 1991.

[10] V.A. Afanasiev, A. Barg, V. Sidorenko, V. Zyablov, U. Dettmar, B. Dorsch, U. Sorger,
"An Estimation of the Performance of Multi-frequency Data Transmission in Mobile Radio Communications",
Preprint, 1994.