b-burst-correcting codes

Schenkeveld, L.N.

Award date:
2011

BACHELOR PROJECT

b-burst-correcting codes

Author:
Loes SCHENKEVELD

Supervisor:
Henk VAN TILBORG

July 12, 2011

Contents

Abstract

If Alice sends a message to Bob, this message can be affected by external factors, which makes it impossible for Bob to read the message. Digital messages are therefore coded. In stead of sending the actual message, Alice sends a coded version of the message. When this coded message is not damaged too much, Bob is able to decode the received message, and therefore he can read the actual message. Obviously the size of the coded message is larger than the size of the original message. The goal is to maximize the number of errors that can be corrected, while keeping the number of added digits as small as possible.
Having this goal in mind, it seems to be a good idea to look at the behavior of errors. In some applications errors tend to come in clusters, like a scratch on a CD. For these cases b-burst-correcting codes are very useful. These codes are able to correct bursts of error up to length b.
This report is about optimum binary b-burst-correcting codes. These codes are binary cyclic codes, generated by generator polynomial $g(x)$. Codes for $b = 3$, $b = 4$ and $b = 5$ are discussed in detail.

1 Introduction

Suppose you got a letter from your friend. While reading it, you make sure that it will not get wet, because otherwise the letter will become unreadable. You want to know what the message is, so you take precautions. And if you are talking to someone with a different accent, you have to listen very carefully, in order to know what he or she is telling you.
In these situation you are the receiver of the message, where you want to make sure that you receive the correct information. If someone talks unclearly, you might not succeed. Or, if you are standing in a crowded room with a lot of noise, you might not be able to hear the message. In digital communication there can be 'noise' as well that affects the message. If a message is affected too much, the receiver will not be able to read the message. To make sure that the receiver gets the exact same message that is sent by the sender, a message is coded.
For example the sender wants to send "1". In stead of sending it once, he sends "111". If the receiver now receives "101" he corrects the 0 in a 1 and decodes it. Because the message consists of more ones then zeros, he assumes that the original message was a "1". However, if two errors would occur, de message would be incorrectly decoded, since there would be more zeros then ones. This is a very expensive way of coding. The message sent is three times longer than the original message, while only one error can be corrected. In order to make this more efficient codes are used which are able to correct as many errors as possible while keeping the message sent as small as possible.

Errors generally occur not by oneself. For example a scratch on a CD. Not just one, but a cluster of digits is affected by this scratch. If an error is detected, the chance is very high that there are more digits affected close to the error. These clusters of affected digits are called bursts. This property of errors makes it able to make more efficient codes. The so-called burst-correcting codes are able to correct bursts up to a certain length b. This report is about optimum binary cyclic b-burst-correcting codes. After some definitions are given, these codes will be discussed in detail.

2 Definitions

This report is about binary cyclic codes of length n that can correct bursts. These codes should be able to correct a single burst of length b or less.
Cyclic codes are linear codes that are invariant under cyclic shifts. This means $\underline{c} = (c_0, c_1, \ldots, c_{n-1})$ is a codeword if and only if $\underline{c}' = (c_{n-1}, c_0, \ldots, c_{n-2})$ is a codeword as well.
Furthermore, a code is said to be linear if and only if any linear combination of codewords is a codeword as well.
A codeword can be seen as a polynomial of degree $n-1$ represented as $c(x) = \sum_{i=0}^{n-1} c_i x^i$.
A cyclic code C is completely characterized by its generator polynomial. This is the (unique) monic polynomial $g(x)$ over $GF(q)$ dividing $x^n - 1$ with the property that $c(x)$ in C if and only if $g(x)$ divides $c(x)$.
Moreover, this report is about binary codes, which means that $c_i \in \{0, 1\}$.

Let C be a $[n, n-r]$ binary cyclic code, with n the length of the code and $n-r$ the length of the original message. The redundancy r is the number of added digits that make it possible to detect and correct the errors.
If $g(x)$ is the generator polynomial of C then G, the $(n-r) \times n$ generator matrix of C, is given by:

$$G = \begin{pmatrix} g_0 & g_1 & \ldots & g_{n-r} & 0 & 0 & \ldots & 0 \\ 0 & g_0 & \ldots & g_{n-r+1} & g_{n-r} & 0 & \ldots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \ldots & g_0 & g_1 & g_2 & \ldots & g_{n-r} \end{pmatrix}$$

Each codeword is a linear combination of the rows of G. This matrix can be used to encode a message $\underline{m}$ into a codeword: $\underline{c} = \underline{m}G$.
There also exists an $n \times r$ parity check matrix H such that:

$$\underline{c} \in C \iff H\underline{c}^\top = \underline{0}^\top$$

The result of $H\underline{w}^\top$ is called the syndrome of word $\underline{w}$. The code C consists of all words having syndrome $\underline{0}$.

Now consider bursts. A burst of length b will be considered as an error pattern starting at position i and ending at position $i+b-1$. The components at positions i and $i+b-1$ are nonzero and the only other nonzero components have coordinate j with $i < j < i+b-1$. Bursts $\underline{b}$ are of the form:

$$\underline{b} = (0, \ldots, 0, b_i, \ldots, b_{i+b-1}, 0, \ldots, 0) = (0, \ldots, 0, 1, *, \ldots, *, 1, 0, \ldots, 0)$$

where the stars can be either zero or one. Similar to codewords bursts can be represented as polynomials. The polynomial representation is of the form $b(x) = \sum_{i=0}^{n-1} b_i x^i$. From now on, bursts will be considered to be cyclic bursts, for which the coordinates of the nonzero elements should be reduced modulo n. If $i+b-1 > n$ the burst $\underline{b}$ is represented the following way:

$$\underline{b} = (\ldots, b_{i+b-1} = b_{i+b-1-n}, 0, \ldots, 0, b_i, \ldots) = (\ldots, *, 1, 0, \ldots, 0, \ldots, 0, 1, *, \ldots)$$

Let $b(x)$ be a burst of length b and consider $B(x) = \sum_{k=0}^{b-1} b_{i+k} x^k$, where i is the starting position of the burst and $i+k$ is reduced modulo n. $B(x)$ stands for the error pattern belonging to the

burst $b(x)$. The error pattern is a monic polynomial of degree $b-1$, while the degree of $b(x)$ does not depend on the length of the burst. Note that shifted versions of bursts have the same error pattern and a burst $\underline{b}$ can therefore be represented as

$$\sum_{i=0}^{n-1} b_i x^i = b(x) = x^i B(x) \pmod{1+x^n}$$

with $0 \leq i \leq n-1$.

A code C that is able to correct these bursts of length less than or equal to b is called an $[n, n-r]$ b-burst-correcting code.

Since bursts have length $\leq b$ the number of possible cyclic bursts is $1 + n2^{b-1}$.
To see this notice that each burst starts with a one followed by $b-1$ entries that can be either zero or one, so there are 2^{b-1} different burst patterns. There are n possible starting positions for these patterns. Together with the $\underline{0}$-vector this results in all distinct cyclic bursts up to length b.

□

Every burst must have a unique syndrome, otherwise the code would not be b-burst-correcting. This leads to constraints on the codes. The number of bursts should be less than or equal to the number of syndromes of a code. From this, a bound for the length of the code can be found:

$$\begin{aligned} 1 + n2^{b-1} &\leq 2^r \\ n &\leq (2^r - 1)2^{-b+1} \\ &= 2^{r-b+1} - 2^{-b+1} \\ n &\leq 2^{r-b+1} - 1 \end{aligned} \tag{1}$$

where the last step follows from the fact that n is an integer.

In this report the codes that are studied are the so-called optimum codes. These are the codes where inequality (1) meets with equality.
Moreover Reiger derived that $r \geq 2b$ in [1]. The interesting codes are therefore the codes of length $n = 2^m - 1$ with $m = r - b + 1 \geq 2b - b + 1 = b + 1$, so

$$m \geq b + 1.$$

Consider the field $GF(2^m)$ generated by a primitive polynomial $p(x)$ which has degree m. $GF(2^m)$ has 2^m elements. These elements can be represented by binary polynomials of degree $< m$:

$$GF(2^m) = \Big\{ \sum_{i=0}^{m-1} f_i x^i \Big| f_i \in \{0, 1\}, 0 \leq i < m \Big\},$$

where addition is binary and multiplication is done modulo $p(x)$.
On the other hand $p(x)$ is a primitive polynomial, which means that the elements $1, x, \ldots, x^{2^m-2}$ are all different, and

$$GF(2^m) = \{0, 1, x, x^2 \ldots, x^{2^m-2}\}$$

Thus, for $p(x)$ a primitive polynomial of degree m and a generator polynomial of $GF(2^m)$ the nonzero elements can be described as

$$x^a \equiv f(x) \pmod{p(x)}$$

where integer a (with $0 \leq a \leq 2^m - 2$)is called the index of the polynomial $f(x) = \sum_{i=0}^{m-1} f_i x^i$.

Now consider error patterns $B(x)$. These polynomials can be represented as bit strings, similarly as bursts and codewords. Error patterns have degree $\leq b - 1$, which is smaller than m. Therefore these polynomials can also be seen as an element of $GF(2^m)$. The code C is generated by $g(x)$ and has length $2^m - 1$, with $m \geq b + 1$. This means that the error patterns can be seen as a power of x:

$$x^a \equiv B(x) \pmod{g(x)}$$

A nonzero error pattern can now be described in three ways: As a bit string, as a polynomial, and as a power of x.

3 Simple cases

In order to find b-burst correcting codes of length $n = 2^m - 1$, the generator polynomials $g(x)$ have to be determined. Elspas and Short have stated necessary conditions for these generator polynomials. First of all, the generator polynomial has to be of the form $g(x) = e(x)p(x)$ where $p(x)$ is a primitive polynomial of degree m and $e(x)$ a polynomial of degree $b-1$. The polynomials $e(x)$ and $p(x)$ have to satisfy some statements, which are stated and proved in the next section. In this section some simple examples are treated.
In case $b = 1$ the b-burst-correcting codes are single-error-correcting codes of length $2^m - 1$. These are simply the Hamming codes. In case $b = 2$ Hamming codes can be used as well. The case $b = 3$ is treated to get an idea of how to determine a generator polynomial $g(x)$ in the general case.

3.1 $b = 1$

The length of the code is $n = 2^m - 1$ and the redundancy $r = m + b - 1$. This means that $r = m$ for $b = 1$. According to the Reiger bound (which says that $m \geq b + 1$), $m \geq 2$.
Let $p(x)$ be a primitive polynomial generating the field $GF(2^m)$. The binary $[n, n-m]$ Hamming code generated by $p(x)$ has parity check matrix

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \end{pmatrix}$$

where α is a primitive element of $GF(2^m)$. The hamming code generated by $g(x) = p(x)$ is a single-error-correcting code, with $\underline{c}$ is a codeword $\Leftrightarrow H\underline{c}^\top = 0$.

To see how an error is corrected, consider a received word $r(x)$ and compute its syndrome.

$$H\underline{r}^\top = \sum_{k=0}^{n-1} r_k \alpha^k = \begin{cases} 0 & r \in C \\ \alpha^i & \text{else} \end{cases}$$

If $H\underline{r}^\top \neq 0$ an error is made. As the code is a single error correcting code, the error made has length 1. This single error occurs at position i and the burst $\underline{b}$ can be represented as $b(x) = x^i$. Now the error can be corrected and

$$c(x) = r(x) - b(x)$$

3.2 $b = 2$

The length of this code is still $n = 2^m - 1$ but the redundancy becomes $r = m+1$. For a two-burst-correcting code m should be ≥ 3 from the fact that $m \geq b + 1$ according to the Reiger bound. Again take $p(x)$ a primitive polynomial of degree m, with α as zero. Consider the code generated by $g(x) = (1 + x)p(x)$. It has parity check matrix

$$H = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \end{pmatrix}$$

The code is a two-burst-correcting code.
Consider a received word $r(x)$. The received word is a combination of a codeword $c(x)$ and a burst $b(x)$ of length 0, 1 or 2, so $r(x) = c(x) + b(x)$. The syndrome is $s = \binom{s_0}{s_1}$ with

$$s_0 = r(\alpha^0) = c(\alpha^0) + b(\alpha^0) = b(\alpha^0),$$

because $c(\alpha^0) = 0$ as α is a zero. Likewise is s_1 defined as

$$s_1 = r(\alpha) = c(\alpha) + b(\alpha) = b(\alpha),$$

using the fact that $c(\alpha) = 0$.
There are three kind of solutions for the syndrome:

$$H\underline{r}^\top = \begin{cases} \binom{0}{0} \\ \binom{1}{\alpha^i} \\ \binom{0}{\alpha^i} \end{cases}$$

If $s = \binom{0}{0} = \binom{b(\alpha^0)}{b(\alpha)}$, no error is made and the received word is a codeword.
If $s = \binom{1}{\alpha^i} = \binom{b(\alpha^0)}{b(\alpha)}$, the burst is of length 1. One error is made at position i, for $s_1 = \alpha^i$.
In the last case where $s = \binom{0}{\alpha^i} = \binom{b(\alpha^0)}{b(\alpha)}$, the burst is of length 2 and of the form $b(x) = x^j(1+x)$. Also $\alpha^i = b(\alpha) = \alpha^j(1 + \alpha)$. So an error of length two is made starting at position j with $\alpha^i = \alpha^j + \alpha^{j+1}$.

The error(s) made can be determined from the syndrome, which enables us to compute the codeword. The error is corrected by

$$c(x) = r(x) - b(x)$$

3.3 $b = 3$

Consider the case $b = 3$. We want to find a generator polynomial of a three-burst-correcting code, which is of the form $e(x)p(x)$. Consider the case where $e(x) = 1 + x + x^2$. In this section we are trying to find a three-burst-correcting code of the form $(1 + x + x^2)p(x)$ where $p(x)$ is a primitive polynomial. Later in this report possible other cases of three-burst-correcting codes will be treated, but for now this example is sufficient.
From the Reigerbound as discussed in Section 2 the degree of $p(x)$ the primitive polynomial $m \geq b + 1 = 4$.
In order to be three-burst-correcting, the syndromes of bursts up to length three should all be different. There are four different error patterns of length ≤ 3: 1, $1 + x$, $1 + x^2$ and $1 + x + x^2$. The length of the code is $n = 2^m - 1$ so the $4 \cdot (2^m - 1)$ bursts are:

$$\begin{aligned} &x^i \\ &x^i(1 + x) \\ &x^i(1 + x^2) \\ &x^i(1 + x + x^2) \end{aligned}$$

where $0 \leq i < 2^m - 1$. These bursts should all be different $\pmod{g(x)}$.
Since the generator polynomial is of the form $(1 + x + x^2)p(x)$, this leads to some restrictions on the generator polynomial. There are cases where two bursts are equal $\pmod{(1 + x + x^2)}$. Therefore these bursts are not allowed to be equal $\pmod{p(x)}$. For example

$$x^2(1 + x) \equiv (1 + x)(1 + x) \equiv x^0(1 + x^2) \pmod{1 + x + x^2}$$

But these two bursts have to have a different syndrome, so

$$x^2(1 + x) \not\equiv x^0(1 + x^2) \pmod{(1 + x + x^2)p(x)}$$

Therefore

$$x^2(1 + x) \not\equiv x^0(1 + x^2) \pmod{p(x)}$$

In general, consider the bursts $\pmod{(1 + x + x^2)}$ to determine these restrictions on $p(x)$. The bursts become:

$$\begin{array}{lcll} x^i & \equiv & x^i & \bmod (1 + x + x^2) \\ x^i(1 + x) & \equiv & x^i(1 + x) & \bmod (1 + x + x^2) \\ x^i(1 + x^2) & \equiv & x^i \cdot x \equiv x^{i+1} & \bmod (1 + x + x^2) \\ x^i(1 + x + x^2) & \equiv & x^i \cdot 0 \equiv 0 & \bmod (1 + x + x^2) \end{array}$$

for all $0 \leq a < 2^m - 1$. The polynomial $(1 + x + x^2)$ is primitive so the upper three types of bursts will never become $0 \pmod{(1 + x + x^2)}$. Furthermore, $x^i \not\equiv x^{i+1} \pmod{(1 + x + x^2)}$. The only possibilities where for bursts that might be equivalent $\pmod{(1 + x + x^2)}$ are:

$$\begin{aligned} x^i &\equiv x^j \\ x^i &\equiv x^j(1 + x) \\ x^i(1 + x) &\equiv x^j(1 + x) \\ x^i(1 + x) &\equiv x^{j+1} \\ x^{i+1} &\equiv x^{j+1} \end{aligned}$$

for all $0 \leq i, j < 2^m - 1$. These possibilities can be reduced to the following equations.

$$x^k \equiv 1 \pmod{(1 + x + x^2)} \tag{2}$$
$$x^k(1 + x) \equiv 1 \pmod{(1 + x + x^2)} \tag{3}$$

where $0 \leq k < 2^m - 1$. The polynomial $p(x)$ is a primitive polynomial so $2^m - 1$ is the first nonzero integer i for which $x^i \equiv 0 \pmod{p(x)}$. Since $k < 2^m - 1$ the first equation will not lead to restrictions on $p(x)$. Now evaluating and $x^k(1 + x)$ gives

$$x^k(1 + x) \equiv \begin{cases} 1 + x & k = 0, 3, 6, \ldots \\ 1 & k = 1, 4, 7, \ldots \\ x & k = 2, 5, 8, \ldots \end{cases} \mod (1 + x + x^2).$$

From this can be concluded that equation (3) holds for $k \equiv 1 \pmod 3$ which implies that

$$x^{1+3l}(1 + x) \not\equiv 1 \pmod{p(x)} \tag{4}$$

In Section 2 is shown that there is an integer a with

$$x^a \equiv 1 + x \pmod{p(x)}$$

Moreover we know that $g(x)|(1 + x^{2^m-1})$ and therefore $p(x)|(1 + x^{2^m-1})$ as well. Now restriction (4) can be reduced.

$$\begin{aligned} x^{1+3l}(1 + x) &\not\equiv 1 \pmod{p(x)} \\ x^{1+3l}x^a &\not\equiv 1 \pmod{p(x)} \\ x^{1+3l+a} &\not\equiv 1 \pmod{p(x)} \\ 1 + 3l + a &\not\equiv 0 \pmod{2^m - 1} \end{aligned}$$

for all l. Since $(1 + x + x^2)\big|1 + x^3$ and $(1 + x + x^2)\big|1 + x^{2^m-1}$ this means that $3|2^m - 1$. This implies that $1 + a \not\equiv 0 \pmod 3$ or

$$a \not\equiv 2 \pmod 3$$

Furthermore, the degree of $p(x)$ has to be even, since $(2^2 - 1 =)3|2^m - 1$, which means that $2|m$.

Now can be concluded that $(1 + x + x^2)p(x)$ generates an optimum three-burst-correcting code with $p(x)$ a primitive polynomial of even degree m with $m \geq 4$ and $a \not\equiv 2 \pmod 3$ for a the index of $(1 + x)$. Examples for polynomials $p(x)$ where $g(x) = (1 + x + x^2)p(x)$ generates a three-burst-correcting code are:

$1 + x + x^4$
$1 + x + x^6$
$1 + x + x^2 + x^3 + x^6 + x^7 + x^8$

Apart from their reciprocals, these are the only polynomials of degree 4 and 6 that are suitable. A complete list of suitable polynomials of degree 8 is given in Appendix A.1.

From now on assume that $b > 2$.

4 Codes

In order to find b-burst-correcting codes of length $n = 2^m - 1$, the generator polynomials $g(x)$ have to be determined. These generator polynomials have to satisfy certain conditions. Elspas and Short have stated necessary conditions for these generator polynomials. First of all, the generator polynomial has to be of the form: $g(x) = e(x)p(x)$ where $p(x)$ is a primitive polynomial of degree m and $e(x)$ a polynomial of degree $b-1$. Moreover the following statements should hold.

1. $e(x)$ is squarefree and not divisible by x.
2. $p(x)$ is a primitive polynomial of degree $m \geq b+1$ and $m_e|m$, defined as the smallest m_e with $e(x)\big|(x^{2^{m_e}-1} - 1)$.

The proof for these statements is as follows.

Let $g(x)$ be a generator polynomial of an optimum b-burst-correcting code of length $n = 2^m - 1$ (where $m = r - b + 1$). Then $g(x)\big|x^{2^m-1} + 1$ and the degree of $g(x) = m + b - 1$. For every polynomial $f(x)$ holds that $GCD(f(x), f'(x)) = 1 \iff f(x)$ is squarefree.
Therefore $x^{2^m-1} + 1$ is squarefree. Hence, $g(x)\big|x^{2^m-1} + 1$ implies that $g(x)$ is a squarefree polynomial which is not divisible by x. This proves the first statement.

Consider the generator polynomial $g(x)$. The degree of $g(x)$ is $m+b-1$. Furthermore $g(x)|(x^{2^m-1}-1)$ hence for l the period of $g(x)$ we know that $l|2^m - 1$. Moreover if $l \neq 2^m - 1$ there is an $u(x)$ with $g(x)u(x) = x^l - 1$. But this would mean that there are two different bursts with the same syndrome. This is forbidden. Therefore, $g(x)$ has period $l = 2^m - 1$.

On the other hand, $g(x)$ can be written as a product of k distinct irreducible polynomials:

$$g(x) = \prod_{i=1}^{k} f_i(x)$$

$g(x)\big|x^{2^m-1} + 1$ implies that $f_i(x)\big|x^{2^m-1} + 1$ for all $i = 1, \ldots, k$. Let d_i be the degree of $f_i(x)$ and l_i the period of $f_i(x)$ (the smallest l such that $f_i(x)|x^l - 1$). Since $g(x)$ has degree $m + b - 1$ and period $2^m - 1$, it follows that:

$$m + b - 1 = deg\,(g(x)) = deg\,(\prod_{i=1}^{k} f_i(x)) = \sum_{i=1}^{k} deg\,(f_i(x)) = \sum_{i=1}^{k} d_i \tag{5}$$

$$2^m - 1 = LCM(l_1, \ldots, l_k) \tag{6}$$

In [2] is shown that the polynomial $(x^{2^m} - x)$ is the product of all monic irreducible polynomials of a degree dividing m. Remember that f_i is an irreducible polynomial. Hence

$$f_i(x)\big|x^{2^m-1} - 1 \Longrightarrow d_i\big|m \text{ thus } d_i \leq m$$

for all $i = 1, \ldots, k$. Similarly

$$f_i(x)\big|x^{l_i} - 1 \Longrightarrow l_i\big|2^{d_i} - 1 \tag{7}$$

Together with equation (6) this implies that

$$2^m - 1 \big| LCM(2^{d_1} - 1, \ldots, 2^{d_k} - 1) \tag{8}$$

From a result in [3] and [4] follows that there always is a j with $1 \leq j \leq k$ where $d_j = m$. This can only occur once.
To proof this, suppose j would not be unique. Then $m + b - 1 = \sum_{i=1}^{k} d_i \geq 2m$ contradicts the Reiger bound, which says that $m \geq b + 1$. Therefore, there is a unique j where $d_j = m$.

□

Without loss of generality, let $j = 1$ and try to determine l_1 the period of $f_1(x)$.
Equations (7) and (8) imply that

$$l_1 \big| 2^{d_1} - 1 \big| 2^m - 1$$

Now the period of $f_1(x)$ is a divisor of the period of $g(x)$. From definition we know that $f_1(x) \big| g(x)$ and $\sum_{i=2}^{k} f_i(x) = \frac{g(x)}{f_1(x)} \big| g(x)$. Together with the fact that $f_1(x) \big| x^{l_1} + 1$ this leads to the following result.

$$(1 + x^{l_1}) \frac{g(x)}{f_1(x)} = \frac{1 + x^{l_1}}{f_1(x)} g(x) \equiv 0 \pmod{g(x)} \tag{9}$$

So

$$\begin{aligned} (1 + x^{l_1}) \frac{g(x)}{f_1(x)} &\equiv 0 \pmod{g(x)} \\ x^{l_1} \frac{g(x)}{f_1(x)} + \frac{g(x)}{f_1(x)} &\equiv 0 \pmod{g(x)} \end{aligned}$$

The degree of $\frac{g(x)}{f_1(x)}$ is $b - 1$. Since $g(x)$ is a b-burst-correcting code, this has to mean that

$$l_1 = 2^m - 1$$

Choose $p(x) = f_1(x)$ and $e(x) = \frac{g(x)}{f_1(x)}$ to finalize the proof for the statements.

□

5 Conditions on $g(x)$

The conditions in Section 4 are the necessary conditions stated by Elspas and Short. If $g(x)$ generates a b-burst-correcting code, it will satisfy the given conditions.

In general a code C is b-burst-correcting if and only if all bursts have different syndromes. Let C be a binary cyclic code generated by $g(x) = e(x)p(x)$ where $e(x)$ and $p(x)$ satisfy the conditions stated in Section 4 which says that $e(x)$ is a squarefree polynomial of degree $b - 1$ and not divisible by x and $p(x)$ is a primitive polynomial of degree $m \geq b - 1$ with $m_e | m$. Now C is a b-burst-correcting code if and only if for all distinct and nonzero error patterns $B_1(x)$ and $B_2(x)$

$$B_1(x) - x^i B_2(x) \equiv 0 \pmod{e(x)} \quad \Rightarrow \quad B_1(x) - x^i B_2(x) \not\equiv 0 \pmod{p(x)} \tag{10}$$

For all $0 \leq i < n$.

5.1 Proof

The code C is a binary cyclic code generated by $g(x) = e(x)p(x)$:

$$C = \{c(x) | c(x) \equiv 0 \pmod{e(x)p(x)}\}$$

Let $B_1(x)$ and $B_2(x)$ be two error patterns belonging to distinct bursts $\underline{b}_1$ respectively $\underline{b}_2$ (the case $\underline{b}_1 = \underline{b}_2$ is trivial, since addition is binary). Now:

$$B_1(x) - x^i B_2(x) \equiv 0 \pmod{e(x)p(x)} \quad \Longleftrightarrow \quad B_1(x) - x^i B_2(x) \in C$$

In the next sections equation 11 is elaborated for some choices for the error patterns.

$$B_1(x) - x^i B_2(x) \equiv 0 \pmod{e(x)p(x)} \tag{11}$$

5.1.1 $B_1(x) = B_2(x)$

For $B_1(x) = B_2(x)$ equation (11) becomes

$$\begin{aligned} B_1(x) - x^i B_2(x) &\equiv 0 \pmod{e(x)p(x)} \\ B_1(x)(1 - x^i) &\equiv 0 \pmod{e(x)p(x)} \end{aligned} \tag{12}$$

where $0 \leq i < 2^m - 1$. Since $p(x)$ is a primitive polynomial and of degree $> b$ equation and $B_1(x)$ is of degree $< b$ equation (12) implies that

$$\begin{aligned} p(x) &\mid 1 - x^i \quad (= x^i + 1) \\ 2^m - 1 &\mid i \end{aligned} \tag{13}$$

Therefore, $i = 0$ and $\underline{b}_1 = \underline{b}_2$.

From now on assume that $B_1(x)$ and $B_2(x)$ are different error patterns.

5.1.2 $B_1(x) = 0$

If one of the error patterns equals 0, without loss of generality let $B_1(x) = 0$. Equation (11) can be rewritten as

$$\begin{aligned} B_1(x) - x^i B_2(x) &\equiv 0 \pmod{e(x)p(x)} \\ x^i B_2(x) &\equiv 0 \pmod{e(x)p(x)} \\ B_2(x) &\equiv 0 \pmod{e(x)p(x)} \end{aligned} \tag{14}$$

for $0 \leq i < 2^m - 1$. This means that $B_2(x)$ corresponds to a codeword as well. The degree of $B_2(x)$ is smaller than b while the degree of $e(x)p(x)$ is larger than b. Therefore $B_2(x) = 0$.
If one of the error patterns equals 0 and their difference is a codeword, the other error pattern is 0 as well. This is quite easy to see, since $B(x) = 0$ means $\underline{b} = 0$.

5.1.3 $B_1(x) = e(x)$

Without loss of generality let $B_1(x) = e(x)$. Equation (11) becomes

$$\begin{aligned} B_1(x) - x^i B_2(x) &\equiv 0 \pmod{e(x)p(x)} \\ e(x) - x^i B_2(x) &\equiv 0 \pmod{e(x)p(x)} \end{aligned} \tag{15}$$

for $0 \leq i < 2^m - 1$. According to Section 5.1.2 $B_2(x)$ is not equal to 0, so $e(x) \big| B_2(x)$. Both $e(x)$ and $B_2(x)$ have degree $b-1$. Therefore $B_2(x) = e(x)$. Now $B_1(x) = B_2(x)$ which means that we are in case 5.1.1.

5.2 Conclusion

At the beginning of this section was stated that a code linear cyclic code C generated by $g(x) = e(x)p(x)$ is b-burst-correcting if and only if all bursts have different syndromes.
Equation (10) now is proven, which says that $e(x)p(x)$ generates a b-burst-correcting code if and only if

$$B_1(x) - x^i B_2(x) \equiv 0 \pmod{e(x)} \quad \Rightarrow \quad B_1(x) - x^i B_2(x) \not\equiv 0 \pmod{p(x)}$$

for all $0 \leq i < 2^m - 1$ and for all nonzero error patterns $B_1(x)$ and $B_2(x)$ with $B_1(x) \neq B_2(x)$ and both not equal to $e(x)$.

6 Constraints on $p(x)$

Condition (10) is very useful to derive generator polynomials. Since $p(x)$ is a primitive polynomial, every nonzero polynomial of degree less than m can be written as $x^a \equiv f(x) \pmod{p(x)}$ as described in Section 2. The degree of error patterns $B(x)$ is less than b, while m is the degree of $p(x)$ and $m \geq b+1$. Therefore $B(x)$ has an unique factorization in irreducible polynomials where each irreducible polynomial can be seen as a power of $x \pmod{p(x)}$ as well. In other words, $B(x) = \prod h_j(x)$ where $h_j(x)$ are irreducible (not necessarily distinct) polynomials. Hence $h_j(x)$ has index a_j and

$$\prod x^{a_j} = x^{\sum a_j} \equiv \prod h_j(x) \equiv B(x) \pmod{g(x)}.$$

Now every error pattern has its own index $\tilde{a}$ with $\tilde{a} = \sum a_j$ the indices of the factors of $B(x)$. Therefore

$$B_1(x) - x^i B_2(x) \not\equiv 0 \pmod{p(x)} \Longleftrightarrow x^{\tilde{a}_1} - x^{i+\tilde{a}_2} \not\equiv 0 \pmod{p(x)} \tag{16}$$

Given the fact that $g(x) = e(x)p(x)$ with $p(x)$ a primitive polynomial, this leads to the condition:

$$\tilde{a}_1 - \tilde{a}_2 \not\equiv i \mod 2^{m_e} - 1 \tag{17}$$

for all indices $\tilde{a}_j$ belonging to error patterns $B_j(x)$ for $j = 1, 2$ and $0 \leq i < 2^m - 1$.
From this condition a set of constraints can be derived for given error polynomial $e(x)$. The constraints are in terms of the indices of the error pattern $B(x)$. Since these error patterns can be written as $B(x) = \prod h_j(x)$ a product of irreducible polynomials, the set of constraints can be seen in terms of the indices of the irreducible polynomials $h(x)$.

6.1 Set of constraints

The constraints depend on the choice of $e(x)$ since equation (17) is $\bmod (2^{m_e} - 1)$, where m_e is defined as the smallest m_e with $e(x) \big| (x^{2^{m_e}-1} - 1)$. For any pair of distinct (and nonzero) error patterns this conditions must hold. This leads to a set of constraints on the indices of irreducible polynomials of degree $< b$. A solution is given as a list where the j-th entry has value $a_j \bmod (2^{m_e} - 1)$ for each index a_j belonging to irreducible polynomial $h_j(x)$.
In Section 3.3 a three-burst-correcting code is wanted with generator polynomial $e(x)p(x) = (1 + x + x^2)p(x)$. The only constraint for $p(x)$ turned out to be

$$a \not\equiv 2 \pmod 3$$

where a is the index for $h(x) = 1 + x$. The set of constraints on $p(x)$ in this case is only one equation.

In general, this set of constraints is larger. Notice that the set of constraints on $p(x)$ is never an inconsistent system, for $\underline{0}$ is always a solution.
To prove this, suppose $\underline{0}$ would not be a solution. $\underline{0}$ stands for $a_j \equiv 0 \mod (2^{m_e} - 1)$ for all j. Moreover, $\tilde{a}_j \equiv 0 \mod (2^{m_e} - 1)$ for all j. Since $\underline{0}$ is not a solution, there are distinct $B_1(x)$ and $B_2(x)$ with $B_1(x) - x^i B_2(x) \equiv 0 \pmod{e(x)}$ and $\tilde{a}_1 - \tilde{a}_2 \equiv i \mod 2^{m_e} - 1$. Since $\tilde{a}_j \equiv 0 \mod 2^{m_e} - 1$ for all j this means that $i = 0$ and therefore $B_1(x) - B_2(x) \equiv 0 \pmod{e(x)}$. $B_1(x)$ and $B_2(x)$ are distinct and of degree less than b, which means that $B_1(x) - B_2(x) = e(x)$. This contradicts the fact that $B_1(x)$ and $B_2(x)$ are nonzero error patterns not divisible by x.

Therefore $\underline{0}$ is always a solution for the set of constraints and the system is never inconsistent.

□

Since the set of constraints is consistent one intuitively can say that if enough polynomials are checked, at one point one would find an appropriate polynomial. In [4] it is shown that for all sufficient large $m \equiv 0 \bmod m_e$ a primitive polynomial does exist such that $e(x)p(x)$ generates an optimum b-burst-correcting code of length $2^m - 1$.

7 Software

The challenge now is to find a generator polynomial $g(x)$ for a b-burst-correcting code. A software program is written in Mathematica to compute suitable polynomials $p(x)$ for which $e(x)p(x)$ is a generator polynomial of a b-burst-correcting code. The program is divided in three pieces. The first part concerns the set of constraints discussed in the Section 6 for given $e(x)$. The second part computes the primitive polynomials of given degree m with $m_e | m$. The last part verifies wether $(a_1, \ldots, a_k)$ of given polynomial $p(x)$, with $x^{a_i} \equiv h_i(x) \pmod{p(x)}$ is a solution for the set constraints determined in the first part.

The program is divided in three parts, but this does not mean that these parts act in sequence. For given $e(x)$ the constraints are derived at first.
Thereafter all primitive polynomials of degree m are computed. For these polynomials $(a_1, \ldots, a_k)$ is determined and checked as possible solution of the set of constraints.

The next sections are used to explain de parts of the program. The code can be found in Appendix D.

7.1 Deriving constraints

First some pre-computation is done. All possible polynomials $e(x)$ are computed for given b. According to Section 4 these are the squarefree polynomials of degree b which are not divisible by x. The next step is to calculate m_e the smallest integer with $e(x) \big| (x^{2^{m_e}-1} - 1)$.
From equation (16) a constraint of the form of equation (17) can be derived. This can be done for each pair of distinct and nonzero error patterns (B_i, B_j). Now the constraints on the indices are computed. The output is therefore a set of inequalities $(\text{mod } (2^{m_e} - 1))$.

7.2 Computing all primitive polynomials of degree m

To find these polynomials consider the field $GF(2^m)$ as a vector space of dimension m. Up to isomorphisms this is a unique field. Let α be a root of a binary primitive polynomial $p(x)$ of degree m. Now α is a generator of $GF(2^m)$. This means that the smallest integer such that $\alpha^k = 1$ is $k = 2^m - 1$.
Consider the element α^j for $1 < j < 2^m - 1$. From a result in [5] it is a generator(i.e. α^j has order $2^m - 1$ as well) if and only if $GCD(j, 2^m - 1) = 1$.
Since α and α^{2^i} are zeros of the same primitive polynomial for $i = 1, \ldots, m-1$, only one of these has to be considered to find the primitive polynomial.

Let L be a list of all integers coprime with $2^m - 1$

$$L = \{j | GCD(j, 2^m - 1) = 1, 1 \leq j < 2^m - 1\}$$

As α^{2^i} are zeros of the same polynomial as α for $i = 1, \ldots, m-1$, all $2^i \mod (2^m - 1)$ can be deleted from L for $i = 1, \ldots, m-1$.
Take $p(x)$ and consider its reciprocal $p^*(x)$. This is a primitive polynomial as well and has α^{-2^i} as zeros with $i = 1, \ldots 2^m - 1$. If $p(x) \neq p^*(x)$ this second primitive polynomial of degree m is given for free. Therefore, in this case the entries $-2^i \mod (2^m - 1)$ for $i = 1, \ldots, m-1$ can be deleted from L as well.

Take j the smallest entry in L. Now α^j is a primitive element of $GF(2^m)$. Consider:

$$GF(2^m) = \langle \{\alpha^0, \alpha^j, \ldots, \alpha^{j \cdot (m-1)}\} \rangle$$

which is a vector space of dimension m. This means that $m+1$ terms are linearly dependant. Moreover, there exist $\lambda_i \in \{0, 1\}$ such that $\sum_{i=0}^{m} \lambda_i \alpha^{j \cdot i} = 0$. Now there exists a primitive polynomial $p_j(x)$ defined as:

$$p_j(x) = \sum_{i=0}^{m} \lambda_i x^{j \cdot i}$$

This polynomial is primitive and of degree m and it is different from $p(x)$ and its reciprocal.
If $p_j(x)$ does not equal its reciprocal, the reciprocal is a primitive polynomial as well, with $\alpha^{-i \cdot j}$

as zeros. Now delete $2^{j \cdot i} \pmod{(2^m - 1)}$ and $-2^{j \cdot i} \pmod{(2^m - 1)}$ for all $i = 1, \ldots, m-1$ from list L and do this calculation again for j' the smallest entry of L. Continue until L is empty.

Once L is empty all primitive polynomials of degree m are found, since there are no integers left that are coprime with $2^m - 1$.

7.3 Finding $g(x) = e(x)p(x)$

The next step is to check wether a primitive polynomial is appropriate for the b-burst-correcting code i.e. if the polynomial satisfies the constraints.
First of all, the constraints on the primitive polynomial $p(x)$ - depending on the choice of $e(x)$ - are derived.
The degree of $p(x)$ depends on the polynomial $e(x)$, as discussed in Section 4. For given degree m all primitive polynomials $p(x)$ are computed as discussed in the previous section.

Once all $p(x)$ are computed, the program calculates the indices for every $p(x)$ and verifies wether it is a suitable polynomial, i.e. if it satisfies the constraints. These constraints are restrictions on a_i the indices of the irreducible polynomials $f_i(x)$ of degree $\leq b$. These indices are derived from $GF(2^m)$ generated by $p(x)$, for $x^{a_i} \equiv h_i(x) \pmod{p(x)}$.
Let $(a_1, a_2, \ldots, a_k) \mod (2^{m_e} - 1)$ be the list of indices belonging to given $p(x)$.

Every equation in the set of constraints is now verified. If all constraint hold for this particular $p(x)$ it is a valid solution and $e(x)p(x)$ generates a b-burst-correcting code. The program gives this polynomial as output and continues with the next polynomial. If one of the constraints fails, the program will not give that $p(x)$ as output and continues with the next polynomial.

8 $b = 3$

In Section 3.3 one example for $b = 3$ is elaborated. In this example $e(x) = 1 + x + x^2$. From Section 4 we now know that $e(x)$ has to be a squarefree polynomial of degree $b - 1 = 2$. The only squarefree polynomial of degree is $1 + x + x^2$. Therefore the only possible three-burst-correcting codes are generated by $(1 + x + x^2)p(x)$, where $p(x)$ is a primitive polynomial of degree m with $m_e | m$. As discussed in Section 3.3 $(1 + x + x^2) | 1 + x^3$ and therefore $2^{m_e} - 1 = 3$ and $m_e = 2$. This means that the degree of $p(x)$ has to be even. In Appendix A.1 some examples for $p(x)$ are given.

9 $b = 4$

Let $b = 4$. There are three possibilities for $e(x)$;

$$e(x) \in \{1 + x^3, 1 + x + x^3, 1 + x^2 + x^3\}$$

The goal is to find $p(x)$ such that $g(x) = e(x)p(x)$ generates an optimum four-burst-correcting code. In order to find $p(x)$ some restrictions on $p(x)$ have to be made. These restrictions follow from constraints on the indices of the irreducible error patterns. To find these constraints, every irreducible error pattern is written as $x^a \pmod{p(x)}$.

$$
\begin{array}{lcll}
1 & = & 1 & = & x^0 \\
1+x & = & 1+x & = & x^{a_1} \\
1+x+x^2 & = & 1+x+x^2 & = & x^{a_2} \\
1+x+x^3 & = & 1+x+x^3 & = & x^{a_3} \\
1+x^2+x^3 & = & 1+x^2+x^3 & = & x^{a_4}
\end{array}
$$

From this can be seen that the variables in the constraints will be the indices of the irreducible polynomials of degree $< b$: $(1, 1+x, 1+x+x^2, 1+x+x^3, 1+x^2+x^3)$ or short $(0, a_1, a_2, a_3, a_4)$. To find the generator polynomials every $e(x)$ has to be considered separately. This is done in the following sections.

9.1 $b = 4$ and $e(x) = 1 + x^3$

Let $b = 4$ and $e(x) = 1 + x^3$. The goal is to find $g(x) = (1 + x^3)p(x)$ such that it generates a four-burst-correcting code.
Consider m the degree of $p(x)$. Determine the smallest m_e such that $1+x^3 | x^{2^{m_e}-1}+1$. Obviously $m_e = 2$, because $2^2 - 1 = 3$. Therefore m the degree of $p(x)$ has to be even. According to the Reiger bound $m \geq b + 1 = 5$. The degree of $p(x)$ is even and ≥ 6.

Now consider the error patterns that lead to constraints on $p(x)$. These are the combinations where $B_1(x) \equiv x^i B_2(x) \pmod{e(x)}$. A table is given with $B(x)$ versus $x^i B(x)$. The error pattern $(1 + x^3)$ is left out as discussed in Section 5, since it is equal to $e(x)$.

B(x)	1	$1+x$	$1+x^2$	$1+x+x^2$	$1+x+x^3$	$1+x^2+x^3$	$1+x+x^2+x^3$
i = 0 mod 3	1	$1+x$	$1+x^2$	$1+x+x^2$	x	x^2	$x+x^2$
i = 1 mod 3	x	$x+x^2$	$1+x$	$1+x+x^2$	x^2	1	$1+x^2$
i = 2 mod 3	x^2	$1+x^2$	$x+x^2$	$1+x+x^2$	1	x	$1+x$

This shows that the combinations of bursts that lead to constraints are the pairs $(1, 1 + x + x^3)$, $(1, 1 + x^2 + x^3)$, $(1 + x + x^3, 1 + x^2 + x^3)$, $(1 + x, 1 + x^2)$, $(1 + x, 1 + x + x^2 + x^3)$ and $(1 + x^2, 1 + x + x^2 + x^3)$. For each pair condition (16) is elaborated.

$$
\begin{array}{rcl}
1 & \not\equiv & x^{2+3l}(1+x+x^3) \pmod{p(x)} \\
1 & \not\equiv & x^{2+3l}x^{a_3} \pmod{p(x)} \\
0 & \not\equiv & 2+3l+a_3 \pmod{2^m-1} \\
a_3 & \not\equiv & 1 \pmod{3}
\end{array} \tag{18}
$$

$$
\begin{aligned}
1 &\not\equiv x^{1+3l}(1+x^2+x^3) \pmod{p(x)} \\
1 &\not\equiv x^{1+3l}x^{a_4} \pmod{p(x)} \\
0 &\not\equiv 1+3l+a_4 \pmod{2^m-1} \\
a_4 &\not\equiv 2 \pmod 3
\end{aligned} \tag{19}
$$

$$
\begin{aligned}
1+x+x^3 &\not\equiv x^{2+3l}(1+x^2+x^3) \pmod{p(x)} \\
x^{a_3} &\not\equiv x^{2+3l}x^{a_4} \pmod{p(x)} \\
a_3 &\not\equiv 2+3l+a_4 \pmod{2^m-1} \\
a_3-a_4 &\not\equiv 2 \pmod 3
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
1+x &\not\equiv x^{1+3l}(1+x^2) \pmod{p(x)} \\
x^{a_1} &\not\equiv x^{1+3l}x^{2a_1} \pmod{p(x)} \\
a_1 &\not\equiv 1+3l+2a_1 \pmod{2^m-1} \\
a_1 &\not\equiv 2 \pmod 3
\end{aligned} \tag{21}
$$

$$
\begin{aligned}
1+x &\not\equiv x^{2+3l}(1+x+x^2+x^3) \pmod{p(x)} \\
x^{a_1} &\not\equiv x^{2+3l}x^{3a_1} \pmod{p(x)} \\
a_1 &\not\equiv 2+3l+3a_1 \pmod{2^m-1} \\
a_1 &\not\equiv 2 \pmod 3
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
1+x^2 &\not\equiv x^{1+3l}(1+x+x^2+x^3) \pmod{p(x)} \\
x^{2a_1} &\not\equiv x^{1+3l}x^{3a_1} \pmod{p(x)} \\
2a_1 &\not\equiv 1+3l+3a_1 \pmod{2^m-1} \\
a_1 &\not\equiv 2 \pmod 3
\end{aligned} \tag{23}
$$

This leads to the constraints on $p(x)$ for $e(x) = 1 + x^3$, where each inequality is (mod 3).

$a_1 \not\equiv 2$
$a_3 \not\equiv 1$
$a_4 \not\equiv 2$
$a_3 - a_4 \not\equiv 2$

From these constraints and the fact that the degree of $p(x) = m$ is even and $m \geq 6$ the appropriate polynomials $p(x)$ can be found. It turns out that there are no primitive polynomials of degree 6 and 8 that satisfy these constraints. Two examples of polynomials that are appropriate are:

$1 + x^2 + x^3 + x^4 + x^5 + x^8 + x^{10}$
$1 + x^3 + x^4 + x^7 + x^{12}$

For degree 10 and 12 all suitable polynomials are listed in Appendix A.2. These polynomials were determined by the Software program. Polynomials of higher degree were not taken into

account (although their existence was shown in [4]).

9.2 $b = 4$ and $e(x) = 1 + x + x^3$

The smallest integer m_e such that $1 + x + x^3 | x^{2^{m_e}-1} + 1$ is satisfied is $m_e = 3$, since $(1 + x + x^3)(1 + x + x^2 + x^4) = x^7 + 1 = x^{2^3-1} + 1$. Therefore the degree of $p(x) = m$ with $3|m$ and $m \geq 5$ hence $m \geq 6$.
Consider the error patterns (mod $e(x)$). Again the error pattern equivalent to $e(x)$ does not have to be considered as a result of Section 5.

B(x)	1	$1+x$	$1+x^2$	$1+x+x^2$	$1+x^3$	$1+x^2+x^3$	$1+x+x^2+x^3$
i = 0 mod 7	1	$1+x$	$1+x^2$	$1+x+x^2$	x	$x+x^2$	x^2
i = 1 mod 7	x	$x+x^2$	1	$1+x^2$	x^2	$1+x+x^2$	$1+x$
i = 2 mod 7	x^2	$1+x+x^2$	x	1	$1+x$	$1+x^2$	$x+x^2$
i = 3 mod 7	$1+x$	$1+x^2$	x^2	x	$x+x^2$	1	$1+x+x^2$
i = 4 mod 7	$x+x^2$	1	$1+x$	x^2	$1+x+x^2$	x	$1+x^2$
i = 5 mod 7	$1+x+x^2$	x	$x+x^2$	$1+x$	$1+x^2$	x^2	1
i = 6 mod 7	$1+x^2$	x^2	$1+x+x^2$	$x+x^2$	1	$1+x$	x

From this table can be seen that every combination of error patterns might lead to a constraint on $p(x)$. In Appendix A.3 the derivation of the constraints are given. The 21 constraints derived from this table result in the following inequalities (mod 7).

$a_1 \not\equiv 3$
$a_2 \not\equiv 5$
$a_1 + a_2 \not\equiv 1$
$a_4 \not\equiv 4$
$a_1 - a_2 \not\equiv 5$
$a_1 - a_4 \not\equiv 6$
$a_1 - 4a_2 \not\equiv 4$
$a_1 - 4a_4 \not\equiv 1$
$a_2 - a_4 \not\equiv 1$
$a_1 - a_2 \not\equiv 4$
$a_1 + a_2 - a_4 \not\equiv 4$
$a_1 - 5a_4 \not\equiv 4$

In order to find a four-burst-correcting code of the form $g(x) = (1 + x + x^3)p(x)$ the primitive polynomial has to have degree $m \geq 6$ and divisible by $m_e = 3$ and it has to fulfill the set of constraints. As mentioned in Section 6 a solution should exist for the degree of $p(x)$ large enough. For $m = 6$ there are no polynomials that fulfill the conditions. For $m = 9$ the program did already find appropriate polynomials. Three examples that are appropriate are:

$1 + x + x^2 + x^3 + x^6 + x^7 + x^9$
$1 + x + x^2 + x^4 + x^{10} + x^{11} + x^{12}$
$1 + x + x^2 + x^4 + x^5 + x^8 + x^{11} + x^{12} + x^{15}$

A list of all polynomials $p(x)$ of degree 9 and 12 are listed in Appendix A.3. The program did not check polynomials of degree higher than 15.

9.3 $b = 4$ and $e(x) = 1 + x^2 + x^3$

The same steps as in the previous section can be taken to derive the constraints which define the primitive polynomial for $g(x) = (1 + x^2 + x^3)p(x)$. However, this primitive polynomial can be retrieved quite easily from the generator polynomial $g(x) = (1 + x + x^3)p(x)$, since $1 + x^2 + x^3$ is the reciprocal of $1 + x + x^3$.

$$(1 + x + x^3)p(x) = (1 + x^2 + x^3)p^*(x)$$

with $p^*(x)$ the reciprocal of $p(x)$. Once $p(x)$ is known the reciprocal can easily be found and the generator polynomial therefore is found as well. Therefore the polynomials $p(x)$ for $e(x) = 1 + x^2 + x^3$ of degree 9 and 12 are exactly the reciprocals of the primitive polynomials belonging to $1 + x + x^3$.

10 $b = 5$

In order for $g(x)$ to correct bursts of length five, $e(x)$ has to have degree four. There are five possible polynomials $e(x)$ of degree four, namely

$$e(x) \in \{1 + x + x^4, 1 + x + x^2 + x^4, 1 + x^3 + x^4, 1 + x^2 + x^3 + x^4, 1 + x + x^2 + x^3 + x^4\}$$

The reciprocals do not have to be considered, since the $p(x)$ belonging to the reciprocal of $e(x)$ is simply the reciprocal of $p(x)$ ($g(x) = e^*(x)p(x) = e(x)p^*(x)$). Therefore there are three possibilities for $e(x)$ that need to be discussed, namely $1+x+x^4$, $1+x+x^2+x^4$ and $1+x^2+x^3+x^4$. Every choice for $e(x)$ is elaborated in the following sections.

10.1 $b = 5$ and $e(x) = 1 + x + x^4$

First consider the smallest m_e such that $1 + x + x^4 | x^{2^{m_e}-1} + 1$. Since $1 + x + x^4 \nmid x^7 + 1$ and $(1 + x + x^4)(1 + x + x^2 + x^3 + x^5 + x^7 + x^8 + x^{11}) = x^{15} + 1$ it can be concluded that $m_e = 4$ and $2^{m_e} - 1 = 15$. Considering the Reiger bound, we see that $m \geq b + 1 = 6$.
So the degree of $p(x) = m$ with $4|m$ and $m \geq 8$.
The program finds forbidden relations for $p(x)$ which can be found in Appendix C.1. Note that these restrictions are (mod 15), since $2^{m_e} - 1 = 15$. Similar to previous sections these constraints are retrieved from the combinations of error patterns.

$$B_1(x) - x^i B_2(x) \equiv 0 \pmod{e(x)} \quad \Rightarrow \quad B_1(x) - x^i B_2(x) \not\equiv 0 \pmod{p(x)}$$

Where $B_i(x)$ are distinct and $B_i(x) \in \{1, 1 + x, 1 + x^2, 1 + x + x^2, 1 + x^3, 1 + x + x^3, 1 + x^2 + x^3, 1 + x + x^2 + x^3, 1 + x^4, 1 + x^2 + x^4, 1 + x + x^2 + x^4, 1 + x^3 + x^4, 1 + x + x^3 + x^4, 1 + x^2 + x^3 + x^4, 1 + x + x^2 + x^3 + x^4\}$. This are the irreducible and squarefree polynomials up to length $b - 1 = 4$ except $e(x)$ as discussed in Section 5.
The second part of the program is able to check the constraints for every primitive polynomial of given degree. For the degree $m = 8$ and 12 there are no primitive polynomials that satisfy all constraints. Polynomials of higher degree were not checked.

10.2 $b = 5$ and $e(x) = 1 + x + x^2 + x^4$

The smallest m_e such that $1+x+x^2+x^4 | x^{2^{m_e}-1}+1$ is 3, because $(1+x+x^2+x^4)(1+x+x^3) = x^7+1$. This result together with the Reiger bound which says that $m > b + 1 = 6$ says that the degree of $p(x)$ $m \geq 9$ with $3|m$.
Again the constraints are obtained from the program. Since $m_e = 3$ these inequalities are (mod 7). The complete set of constraints is given in Appendix C.2.
There are no polynomial $p(x)$ of degree 9 and 12 satisfying all constraints. An example of a primitive polynomial $p(x)$ with $e(x) = 1 + x + x^2 + x^4$ where $(1 + x + x^2 + x^4)p(x)$ generates a 5-burst-correcting code is:

$$1 + x^2 + x^5 + x^6 + x^{10} + x^{12} + x^{13} + x^{14} + x^{15}$$

The program did not check all polynomials of degree 15, but stopped as soon as this polynomial was found. Higher degree polynomials are not checked as well.

10.3 $b = 5$ and $e(x) = 1 + x + x^2 + x^3 + x^4$

For $e(x) = 1+x+x^2+x^3+x^4$ the degree of $p(x)$ should be > 6, according to the Reiger bound. On the other hand, $m_e|m$ with m_e as small as possible such that $1+x+x^2+x^3+x^4 | x^{2^{m_e}-1}+1$. Because $1+x+x^2+x^3+x^4 \nmid x^7+1$ and $(1+x+x^2+x^3+x^4)(1+x+x^5+x^6+x^{10}+x^{11}) = x^{2^4-1}+1 = x^{15}+1$ we see that $m_e = 4$.
In order for $g(x) = (1+x+x^2+x^3+x^4)p(x)$ te be a generator polynomial for a 5-burst-correcting code the degree of $p(x)$ m satisfies $m \geq 8$ and $4|m$.
A complete set of constraint on $p(x)$ can be found in Appendix C.3. These constraints on $p(x)$ are derived by the program.
There are no appropriate primitive polynomials of degree 8 and 12. An example of a polynomial that satisfies the given constraints of degree 16 is:

$$1 + x^2 + x^4 + x^5 + x^6 + x^7 + x^{11} + x^{13} + x^{16}$$

Since $e(x) = 1 + x + x^2 + x^3 + x^4$ is its own reciprocal the reciprocal of $p(x)$ is an appropriate polynomial as well. The program stopped immediately when a suitable polynomial of degree 16 was found, so a complete list of primitive polynomials of degree 16 is not given.

11 Computational time

Unfortunately the program was not able to compute suitable primitive polynomials of degree ≥ 16. This is due to the number of calculations needed to compute these polynomials. The calculations that lead to limitations on capacity of the program are discussed here.

11.1 Deriving primitive polynomials

Deriving the primitive polynomials of degree m takes a lot of effort. The number of irreducible polynomials of degree m is approximately $\frac{2^m}{m}$, which grows exponentially. Therefore computing all primitive polynomials takes a lot of computational time. In order to remedy this, only roughly half of the polynomials are computed, since you get the reciprocal for free. Unfortunately you still

have to compute the elements of the field of the reciprocal to derive the indices a_i for $f_i(x) \equiv x^{a_i} \pmod{p(x)}$. Therefore not much is gained by this. It is only interesting for the polynomials $e(x)$ which are equal to their reciprocal, since for a suitable $p(x)$ its reciprocal is automatically a suitable polynomial as well.

11.2 Computing indices a_i

Unfortunately this does not solve the problem completely. Computing the indices is probably the most consuming part of the program. In order to get the indices of the irreducible polynomials, every element of the field has to be computed. The size of the field is $2^m - 1$, which has to be computed for èvery polynomial $p(x)$. This number grows exponentially as well.
To catch this problem, these indices are not computed for all primitive polynomials at once. After these indices are derived for given polynomials, the constraints are immediately tested. If this polynomial is a feasible solution for the set of constraints, $p(x)$ will directly be given as output. Computing the complete list of suitable polynomials of degree m will not take less time, but you will get intermediate results.

11.3 Limitation on size of b

Moreover, the program does not search for generator polynomials of burst-correcting codes with bursts longer than 5. For these codes the smallest m_e such that $e(x)|(x^{2^{m_e}-1}+1)$ becomes large, which means that the degree of p(x)becomes larger, for $m|m_e$. Therefore it does not make sense to search for b-burst-correcting codes with $b > 5$. Furthermore, the set of constraints will grow for higher degrees. There are approximately $\frac{2^l}{l}$ irreducible polynomials of degree l. Therefore the set of constraints on the indices of the error patterns as described in Section 9.1 increases when b increases. When the set of constraints increases the computational time increases.

11.4 Conclusion

In Section 6.1 is mentioned that for every $e(x)$ an appropriate $p(x)$ can be found, as long as m is large enough. The lower bound for m however is pessimistic, with high probability these polynomials exist for m smaller as well. In this report some appropriate polynomials for m relatively small are derived. This gives the impression that although the statistic lower bound M on m is very high, there exist polynomials with degree much less than M.

A Appropriate generator polynomials

A.1 $e(x) = 1 + x + x^2$

A complete list of feasible polynomials $p(x)$ of degree 4, 6 and 8 such that $g(x) = (1+x+x^2)p(x)$ generates a three-burst-correcting code. Since $e(x) = 1 + x + x^2$ is it's own reciprocal, the reciprocals of these polynomials are feasible as well. These are excluded from the list. Feasible polynomials of higher degree (that do exist) are not included as well.

$1 + x + x^4$
$1 + x + x^6$
$1 + x + x^2 + x^5 + x^6$
$1 + x + x^2 + x^3 + x^6 + x^7 + x^8$
$1 + x^2 + x^3 + x^4 + x^8$
$1 + x + x^6 + x^7 + x^8$
$1 + x^2 + x^3 + x^5 + x^8$
$1 + x^3 + x^5 + x^7 + x^8$

A.2 $e(x) = 1 + x^3$

A complete list of feasible polynomials $p(x)$ of degree 10 and 12 such that $g(x) = (1 + x^3)p(x)$ generates a four-burst-correcting code. Since $e(x) = 1 + x^3$ is it's own reciprocal, the reciprocals of these polynomials are feasible as well. These are excluded from the list. One polynomial of degree 15 is given as well, but not the complete list. Feasible polynomials of higher degree (that do exist) are left out as well.

$1 + x^2 + x^3 + x^4 + x^5 + x^8 + x^{10}$
$1 + x + x^3 + x^4 + x^6 + x^9 + x^{10}$
$1 + x + x^2 + x^3 + x^6 + x^9 + x^{10}$
$1 + x^3 + x^4 + x^6 + x^8 + x^9 + x^{10}$
$1 + x^2 + x^3 + x^5 + x^{10}$
$1 + x^2 + x^3 + x^4 + x^6 + x^7 + x^8 + x^{10} + x^{12}$
$1 + x + x^3 + x^4 + x^5 + x^7 + x^{10} + x^{11} + x^{12}$
$1 + x^2 + x^6 + x^8 + x^9 + x^{10} + x^{12}$
$1 + x + x^6 + x^8 + x^{10} + x^{11} + x^{12}$
$1 + x^3 + x^4 + x^7 + x^{12}$
$1 + x + x^2 + x^8 + x^{+}x^{11} + x^{12}$
$1 + x^3 + x^7 + x^8 + x^{10} + x^{11} + x^{12}$
$1 + x^2 + x^3 + x^4 + x^8 + x^{10} + x^{12}$
$1 + x^2 + x^4 + x^5 + x^8 + x^{11} + x^{12}$
$1 + x + x^2 + x^6 + x^9 + x^{10} + x^{12}$
$1 + x + x^2 + x^3 + x^4 + x^5 + x^8 + x^9 + x^{12}$
$1 + x + x^2 + x^5 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$
$1 + x^3 + x^4 + x^5 + x^8 + x^9 + x^{12}$
$1 + x + x^2 + x^4 + x^5 + x^8 + x^{11} + x^{12} + x^{15}$

A.3 $e(x) = 1 + x + x^3$

A complete list of feasible polynomials $p(x)$ of degree 9 and 12 such that $g(x) = (1+x+x^3)p(x)$ generates a four-burst-correcting code. Polynomials of higher degrees are not checked, and therefore left out. The reciprocals of these polynomials are feasible for $e(x) = 1 + x^2 + x^3$, which is the reciprocal of $1 + x + x^3$.

$1 + x + x^2 + x^3 + x^6 + x^7 + x^9$

$1 + x + x^2 + x^4 + x^{10} + x^{11} + x^{12}$
$1 + x^3 + x^9 + x^{10} + x^{12}$
$1 + x + x^2 + x^3 + x^4 + x^7 + x^{10} + x^{11} + x^{12}$
$1 + x + x^2 + x^3 + x^8 + x^9 + x^{12}$

B Derivation of constraints

B.1 $b = 4$ and $e(x) = 1 + x + x^3$

The derivation of the set of constraints for $b = 4$ and $e(x) = 1 + x + x^3$.

$$\begin{aligned} 1 &\not\equiv x^{4+7l}(1+x) \pmod{p(x)} \\ 1 &\not\equiv x^{4+7l}x^{a_1} \pmod{p(x)} \\ 0 &\not\equiv 4 + 7l + a_1 \pmod{2^m - 1} \\ a_1 &\not\equiv 3 \pmod 7 \end{aligned} \tag{24}$$

$$\begin{aligned} 1 &\not\equiv x^{1+7l}(1+x^2) \pmod{p(x)} \\ 1 &\not\equiv x^{1+7l}x^{2a_1} \pmod{p(x)} \\ 0 &\not\equiv 1 + 7l + 2a_1 \pmod{2^m - 1} \\ 2a_1 &\not\equiv 6 \pmod 7 \\ a_1 &\not\equiv 3 \pmod 7 \end{aligned} \tag{25}$$

$$\begin{aligned} 1 &\not\equiv x^{2+7l}(1+x+x^2) \pmod{p(x)} \\ 1 &\not\equiv x^{2+7l}x^{a_2} \pmod{p(x)} \\ 0 &\not\equiv 2 + 7l + a_2 \pmod{2^m - 1} \\ a_2 &\not\equiv 5 \pmod 7 \end{aligned} \tag{26}$$

$$\begin{aligned} 1 &\not\equiv x^{6+7l}(1+x^3) \pmod{p(x)} \\ 1 &\not\equiv x^{6+7l}x^{a_1+a_2} \pmod{p(x)} \\ 0 &\not\equiv 6 + 7l + a_1 + a_2 \pmod{2^m - 1} \\ a_1 + a_2 &\not\equiv 1 \pmod 7 \end{aligned} \tag{27}$$

$$\begin{aligned} 1 &\not\equiv x^{3+7l}(1+x^2+x^3) \pmod{p(x)} \\ 1 &\not\equiv x^{3+7l}x^{a_4} \pmod{p(x)} \\ 0 &\not\equiv 3 + 7l + a_4 \pmod{2^m - 1} \\ a_4 &\not\equiv 4 \pmod 7 \end{aligned} \tag{28}$$

$$\begin{aligned} 1 &\not\equiv x^{5+7l}(1+x+x^2+x^3) \pmod{p(x)} \\ 1 &\not\equiv x^{5+7l}x^{3a_1} \pmod{p(x)} \\ 0 &\not\equiv 5 + 7l + 3a_1 \pmod{2^m - 1} \\ 3a_1 &\not\equiv 2 \pmod 7 \\ a_1 &\not\equiv 3 \pmod 7 \end{aligned} \tag{29}$$

$$
\begin{aligned}
1+x &\not\equiv x^{4+7l}(1+x^2) \pmod{p(x)} \\
x^{a_1} &\not\equiv x^{4+7l}x^{2a_1} \pmod{p(x)} \\
a_1 &\not\equiv 4+7l+2a_1 \pmod{2^m-1} \\
a_1 &\not\equiv 3 \pmod 7
\end{aligned} \tag{30}
$$

$$
\begin{aligned}
1+x &\not\equiv x^{5+7l}(1+x+x^2) \pmod{p(x)} \\
x^{a_1} &\not\equiv x^{5+7l}x^{a_2} \pmod{p(x)} \\
a_1 &\not\equiv 5+7l+a_2 \pmod{2^m-1} \\
a_1-a_2 &\not\equiv 5 \pmod 7
\end{aligned} \tag{31}
$$

$$
\begin{aligned}
1+x &\not\equiv x^{2+7l}(1+x^3) \pmod{p(x)} \\
x^{a_1} &\not\equiv x^{2+7l}x^{a_1+a_2} \pmod{p(x)} \\
a_1 &\not\equiv 2+7l+a_1+a_2 \pmod{2^m-1} \\
a_2 &\not\equiv 5 \pmod 7
\end{aligned} \tag{32}
$$

$$
\begin{aligned}
1+x &\not\equiv x^{6+7l}(1+x^2+x^3) \pmod{p(x)} \\
x^{a_1} &\not\equiv x^{6+7l}x^{a_4} \pmod{p(x)} \\
a_1 &\not\equiv 6+7l+a_4 \pmod{2^m-1} \\
a_1-a_4 &\not\equiv 6 \pmod 7
\end{aligned} \tag{33}
$$

$$
\begin{aligned}
1+x &\not\equiv x^{1+7l}(1+x+x^2+x^3) \pmod{p(x)} \\
x^{a_1} &\not\equiv x^{1+7l}x^{3a_1} \pmod{p(x)} \\
a_1 &\not\equiv 1+7l+3a_1 \pmod{2^m-1} \\
2a_1 &\not\equiv 6 \pmod 7 \\
a_1 &\not\equiv 3 \pmod 7
\end{aligned} \tag{34}
$$

$$
\begin{aligned}
1+x^2 &\not\equiv x^{1+7l}(1+x+x^2) \pmod{p(x)} \\
x^{2a_1} &\not\equiv x^{1+7l}x^{a_2} \pmod{p(x)} \\
2a_1 &\not\equiv 1+7l+a_2 \pmod{2^m-1} \\
2a_1-a_2 &\not\equiv 1 \pmod 7 \\
a_1-4a_2 &\not\equiv 4 \pmod 7
\end{aligned} \tag{35}
$$

$$
\begin{aligned}
1+x^2 &\not\equiv x^{5+7l}(1+x^3) \pmod{p(x)} \\
x^{2a_1} &\not\equiv x^{5+7l}x^{a_1+a_2} \pmod{p(x)} \\
2a_1 &\not\equiv 5+7l+a_1+a_2 \pmod{2^m-1} \\
a_1-a_2 &\not\equiv 5 \pmod 7
\end{aligned} \tag{36}
$$

$$
\begin{aligned}
1+x^2 &\not\equiv x^{2+7l}(1+x^2+x^3) \pmod{p(x)} \\
x^{2a_1} &\not\equiv x^{2+7l}x^{a_4} \pmod{p(x)} \\
2a_1 &\not\equiv 2+7l+a_4 \pmod{2^m-1} \\
2a_1-a_4 &\not\equiv 2 \pmod 7 \\
a_1-4a_4 &\not\equiv 1 \pmod 7
\end{aligned} \tag{37}
$$

$$
\begin{aligned}
1+x^2 &\not\equiv x^{4+7l}(1+x+x^2+x^3) \pmod{p(x)} \\
x^{2a_1} &\not\equiv x^{4+7l}x^{3a_1} \pmod{p(x)} \\
2a_1 &\not\equiv 4+7l+3a_1 \pmod{2^m-1} \\
a_1 &\not\equiv 3 \pmod 7
\end{aligned} \tag{38}
$$

$$
\begin{aligned}
1+x+x^2 &\not\equiv x^{4+7l}(1+x^3) \pmod{p(x)} \\
x^{a_2} &\not\equiv x^{4+7l}x^{a_1+a_2} \pmod{p(x)} \\
a_2 &\not\equiv 4+7l+a_1+a_2 \pmod{2^m-1} \\
a_1 &\not\equiv 3 \pmod 7
\end{aligned} \tag{39}
$$

$$
\begin{aligned}
1+x+x^2 &\not\equiv x^{1+7l}(1+x^2+x^3) \pmod{p(x)} \\
x^{a_2} &\not\equiv x^{1+7l}x^{a_4} \pmod{p(x)} \\
a_2 &\not\equiv 1+7l+a_4 \pmod{2^m-1} \\
a_2-a_4 &\not\equiv 1 \pmod 7
\end{aligned} \tag{40}
$$

$$
\begin{aligned}
1+x+x^2 &\not\equiv x^{3+7l}(1+x+x^2+x^3) \pmod{p(x)} \\
x^{a_2} &\not\equiv x^{3+7l}x^{3a_1} \pmod{p(x)} \\
a_2 &\not\equiv 3+7l+3a_1 \pmod{2^m-1} \\
a_1-a_2 &\not\equiv 4 \pmod 7
\end{aligned} \tag{41}
$$

$$
\begin{aligned}
1+x^3 &\not\equiv x^{4+7l}(1+x^2+x^3) \pmod{p(x)} \\
x^{a_1+a_2} &\not\equiv x^{4+7l}x^{a_4} \pmod{p(x)} \\
a_1+a_2 &\not\equiv 4+7l+a_4 \pmod{2^m-1} \\
a_1+a_2-a_4 &\not\equiv 4 \pmod 7
\end{aligned} \tag{42}
$$

$$
\begin{aligned}
1+x^3 &\not\equiv x^{6+7l}(1+x+x^2+x^3) \pmod{p(x)} \\
x^{a_1+a_2} &\not\equiv x^{6+7l}x^{3a_1} \pmod{p(x)} \\
a_1+a_2 &\not\equiv 6+7l+3a_1 \pmod{2^m-1} \\
2a_1-a_2 &\not\equiv 1 \pmod 7 \\
a_1-4a_2 &\not\equiv 4 \pmod 7
\end{aligned} \tag{43}
$$

$$
\begin{aligned}
1 + x^2 + x^3 &\not\equiv x^{2+7l}(1 + x + x^2 + x^3) \pmod{p(x)} \\
x^{a_4} &\not\equiv x^{2+7l} x^{3a_1} \pmod{p(x)} \\
a_4 &\not\equiv 2 + 7l + 3a_1 \pmod{2^m - 1} \\
3a_1 - a_4 &\not\equiv 5 \pmod{7} \\
a_1 - 5a_4 &\not\equiv 4 \pmod{7}
\end{aligned} \tag{44}
$$

C Set of constraints for $b = 5$

C.1 $e(x) = 1 + x + x^4$

The set of constraints $\pmod{15}$ on $p(x)$ for $e(x) = 1 + x + x^4$.

$a_1 \not\equiv 4$
$2a_1 - a_2 \not\equiv 13$
$3a_1 - a_2 \not\equiv 2$
$4a_1 - a_2 \not\equiv 6$
$a_2 \not\equiv 10$
$-a_1 + a_2 \not\equiv 6$
$a_1 + a_2 \not\equiv 14$
$2a_1 + a_2 \not\equiv 3$
$-3a_1 + 2a_2 \not\equiv 8$
$-a_1 + 2a_2 \not\equiv 1$
$3a_1 - a_3 \not\equiv 5$
$4a_1 - a_3 \not\equiv 9$
$2a_1 + a_2 - a_3 \not\equiv 11$
$2a_2 - a_3 \not\equiv 13$
$a_3 \not\equiv 7$
$-2a_1 + a_3 \not\equiv 14$
$-a_1 + a_3 \not\equiv 3$
$a_1 + a_3 \not\equiv 11$
$a_1 - 2a_2 + a_3 \not\equiv 6$
$-a_2 + a_3 \not\equiv 12$
$-a_1 - a_2 + a_3 \not\equiv 8$
$a_1 - a_2 + a_3 \not\equiv 1$
$3a_1 - a_4 \not\equiv 14$
$4a_1 - a_4 \not\equiv 3$
$a_1 + a_2 - a_4 \not\equiv 1$
$2a_1 + a_2 - a_4 \not\equiv 5$
$2a_2 - a_4 \not\equiv 7$
$a_3 - a_4 \not\equiv 9$
$a_1 + a_3 - a_4 \not\equiv 13$
$a_4 \not\equiv 13$
$-2a_1 + a_4 \not\equiv 5$
$-a_1 + a_4 \not\equiv 9$
$a_1 + a_4 \not\equiv 2$
$a_1 - 2a_2 + a_4 \not\equiv 12$
$-a_2 + a_4 \not\equiv 3$
$a_1 - a_2 + a_4 \not\equiv 7$
$-a_3 + a_4 \not\equiv 6$
$a_1 - a_3 + a_4 \not\equiv 10$
$2a_1 + a_2 - a_6 \not\equiv 9$
$a_1 + a_3 - a_6 \not\equiv 2$
$a_6 \not\equiv 9$
$-4a_1 + a_6 \not\equiv 8$
$-3a_1 + a_6 \not\equiv 12$
$-2a_1 + a_6 \not\equiv 1$
$-a_1 + a_6 \not\equiv 5$
$-2a_2 + a_6 \not\equiv 4$
$-a_2 + a_6 \not\equiv 14$
$-a_1 - a_2 + a_6 \not\equiv 10$
$-a_3 + a_6 \not\equiv 2$
$-a_4 + a_6 \not\equiv 11$
$-a_1 - a_4 + a_6 \not\equiv 7$
$a_7 \not\equiv 6$
$-4a_1 + a_7 \not\equiv 5$
$-3a_1 + a_7 \not\equiv 9$
$-2a_1 + a_7 \not\equiv 13$
$-a_1 + a_7 \not\equiv 2$
$-2a_2 + a_7 \not\equiv 1$
$-a_2 + a_7 \not\equiv 11$
$-2a_1 - a_2 + a_7 \not\equiv 3$
$-a_1 - a_2 + a_7 \not\equiv 7$
$-a_3 + a_7 \not\equiv 14$
$-a_1 - a_3 + a_7 \not\equiv 10$
$-a_4 + a_7 \not\equiv 8$
$-a_1 - a_4 + a_7 \not\equiv 4$
$-a_6 + a_7 \not\equiv 12$

C.2 $e(x) = 1 + x + x^2 + x^4$

The set of constraints $\pmod 7$ on $p(x)$ for $e(x) = 1 + x + x^2 + x^4$.

$$
\begin{aligned}
&a_1 \not\equiv 5\\
&2a_1 - a_2 \not\equiv 6\\
&3a_1 - a_2 \not\equiv 4\\
&a_2 \not\equiv 4\\
&-a_1 + a_2 \not\equiv 6\\
&a_1 + a_2 \not\equiv 2\\
&2a_2 - a_3 \not\equiv 2\\
&a_3 \not\equiv 6\\
&-3a_1 + a_3 \not\equiv 5\\
&-2a_1 + a_3 \not\equiv 3\\
&-a_1 + a_3 \not\equiv 1\\
&-a_2 + a_3 \not\equiv 2\\
&-a_1 - a_2 + a_3 \not\equiv 4\\
&2a_2 - a_5 \not\equiv 6\\
&a_5 \not\equiv 2\\
&-a_2 + a_5 \not\equiv 5\\
&-a_3 + a_5 \not\equiv 3\\
&a_6 \not\equiv 5\\
&-2a_2 + a_6 \not\equiv 4\\
&-a_2 + a_6 \not\equiv 1\\
&-a_3 + a_6 \not\equiv 6\\
&-a_5 + a_6 \not\equiv 3\\
&a_7 \not\equiv 3\\
&-2a_2 + a_7 \not\equiv 2\\
&-a_2 + a_7 \not\equiv 6\\
&-a_3 + a_7 \not\equiv 4\\
&-a_5 + a_7 \not\equiv 1\\
&-a_6 + a_7 \not\equiv 5
\end{aligned}
$$

C.3 $e(x) = 1 + x + x^2 + x^3 + x^4$

The set of constraints $\pmod{15}$ on $p(x)$ for $e(x) = 1 + x + x^2 + x^3 + x^4$.

$3a_1 \not\equiv 4$
$3a_1 \not\equiv 9$
$3a_1 \not\equiv 14$
$4a_1 - a_2 \not\equiv 1$
$4a_1 - a_2 \not\equiv 6$
$4a_1 - a_2 \not\equiv 11$
$-a_1 + a_2 \not\equiv 3$
$-a_1 + a_2 \not\equiv 8$
$-a_1 + a_2 \not\equiv 13$
$2a_1 + a_2 \not\equiv 2$
$2a_1 + a_2 \not\equiv 7$
$2a_1 + a_2 \not\equiv 12$
$2a_2 - a_3 \not\equiv 4$
$2a_2 - a_3 \not\equiv 9$
$2a_2 - a_3 \not\equiv 14$
$-2a_1 + a_3 \not\equiv 2$
$-2a_1 + a_3 \not\equiv 7$
$-2a_1 + a_3 \not\equiv 12$
$a_1 + a_3 \not\equiv 1$
$a_1 + a_3 \not\equiv 6$
$a_1 + a_3 \not\equiv 11$
$-a_1 - a_2 + a_3 \not\equiv 4$
$-a_1 - a_2 + a_3 \not\equiv 9$
$-a_1 - a_2 + a_3 \not\equiv 14$
$a_1 + a_2 - a_4 \not\equiv 4$
$a_1 + a_2 - a_4 \not\equiv 9$
$a_1 + a_2 - a_4 \not\equiv 14$
$2a_2 - a_4 \not\equiv 2$
$2a_2 - a_4 \not\equiv 7$
$2a_2 - a_4 \not\equiv 12$
$a_3 - a_4 \not\equiv 3$
$a_3 - a_4 \not\equiv 8$
$a_3 - a_4 \not\equiv 13$
$-2a_1 + a_4 \not\equiv 4$
$-2a_1 + a_4 \not\equiv 9$
$-2a_1 + a_4 \not\equiv 14$
$a_1 + a_4 \not\equiv 3$
$a_1 + a_4 \not\equiv 8$
$a_1 + a_4 \not\equiv 13$
$-4a_1 + a_5 \not\equiv 3$
$-4a_1 + a_5 \not\equiv 8$
$-4a_1 + a_5 \not\equiv 13$
$-a_1 + a_5 \not\equiv 2$
$-a_1 + a_5 \not\equiv 7$
$-a_1 + a_5 \not\equiv 12$
$-a_2 + a_5 \not\equiv 4$
$-a_2 + a_5 \not\equiv 9$
$-a_2 + a_5 \not\equiv 14$
$-4a_1 + a_6 \not\equiv 2$
$-4a_1 + a_6 \not\equiv 7$
$-4a_1 + a_6 \not\equiv 12$
$-a_1 + a_6 \not\equiv 1$
$-a_1 + a_6 \not\equiv 6$
$-a_1 + a_6 \not\equiv 11$
$-a_2 + a_6 \not\equiv 3$
$-a_2 + a_6 \not\equiv 8$
$-a_2 + a_6 \not\equiv 13$
$-a_5 + a_6 \not\equiv 4$
$-a_5 + a_6 \not\equiv 9$
$-a_5 + a_6 \not\equiv 14$

D Program

```
<<FiniteFields`

Test[bb_ , ee_] :=
  Module[{RTest = ConstantArray[0, Length[B]]},
   For[i = 1, i <= Length[B], i++,
    RTest[[i]] =
     Cases[Position[bb ,
       PolynomialMod[B[[i]], ee, Modulus -> 2]], {_, _}]];
   RTest
    For[i = 1, i <= Length[B], i++,
     Table[PrependTo[RTest[[i]][[j]], i], {j, Length[RTest[[i]]]}];
     RTest[[i]] = Cases[RTest[[i]], Except[{s, _, _}]]];
   RTest = Flatten[DeleteCases[RTest, {}], 1];
   For[i = 1, i <= Length[RTest], i++,
    If[RTest[[i, 1]] >= RTest[[i, 2]], RTest = Delete[RTest, {i}] ;
     i = i - 1;]];
   RTest];


pol[t_] := Module[{Ref = Map[Prepend[#, 1] &, PowerList[GF[2, t]]]},
  n = t;
  Ref = Prepend[Ref, {1, 0}];
  B = Sort[FieldIrreducible[GF[2, #], x] & /@ Ref];
    (*List of all  (2^n) polynomials up to degree n (burstpatterns)*)
  F = Extract[B, Take[Position[B, x^n, Infinity], All,
      1]]; (*lijst met alle polynomen graad = n*)
  F = Expand[Select[Factor[F, Modulus -> 2], SquareFreeQ],
      Modulus -> 2];
  stringF = (PolynomialToElement[GF[2, n + 1], #] & /@ F)[[All,
      1]]; (*stringrepresentation of F*)
  AA1 = Factor[B, Modulus -> 2];
  Bprim = Select[AA1, IrreduciblePolynomialQ];
    (*irreducible polynomials up to degree n*)
  stringB = (PolynomialToElement[GF[2, n + 1], #] & /@ Bprim)[[All,1]];
    (*stringrepresentation*)
  AA = Replace[AA1, 1 -> 0, 1];
  For[i = 1, i <= Length[Bprim], i++,
    AA = Replace[AA, Bprim[[i]] -> Subscript[a, i], 3]];
    AA = AA /. x_ y_ -> x + y /.
    x_^a_ -> a x;(*power of x for each burst in B*)
  Print[StringForm["All square free polynomials with degree `` are ``", t, F]]
  ]


modver[k_] := Module[{},
```

```
  Subscript[m, e] = Length[DeleteDuplicates[
     PowerList[GF[2, stringF[[k]]]]]];(*size splitting field*)
  burst = Drop[Transpose[
     NestList[PolynomialMod[x #, F[[k]], Modulus -> 2] &,
     PolynomialMod[B, F[[k]], Modulus -> 2], Subscript[m, e] - 1]],
     None, 1];(*burstpatterns x^i*)
  Burst = Test[burst, F[[k]]];
  f[a_] := AA[[a]];
  Bursta = MapAt[f, #, {{1}, {2}}] & /@ Burst;
  Burstmod = DeleteDuplicates[Replace[Bursta, {x_, y_, z_} -> y - x + z, 2]];
  {ll, A} =  Normal[CoefficientArrays[Burstmod,
     Array[Subscript[a, #] &, Length[Bprim]]]];
  ll = Mod[-ll, Subscript[m, e]];
  Print[StringForm["The inequalities mod(``) with e(x)=``", Subscript[
    m, e], F[[k]]]]; Print[Burstmod // TableForm];
        (*prime[a]  a is degree of primitive polynomial*)
  ]


prime[n_] := Module[{Field = GF[2, n]},
  one = Field[{1, 0, 0, 0}];
  alpha = Field[{0, 1, 0, 0}];
  primitives = {FieldIrreducible[Field, x]};
  primeList = {1};
  L = Select[Range[2^n - 1],
    CoprimeQ[#, 2^n - 1] &]; (*lijst met cosets voor primitieve polynomen*)
  j = 1;
  While[Length[L] > 1,
    k = First[L];
    For[i = 0, i < n, i++, L = DeleteCases[L, Mod[k*2^i, 2^n - 1]];
    L = DeleteCases[L, Mod[(-k) 2^ i, 2^n - 1]]];
    v = Array[one*alpha^(k*(# - 1)) &, n + 1][[All, 1]];
    Subscript[g, j] = Flatten[NullSpace[Transpose[v], Modulus -> 2]];
    Subscript[f, j] = FieldIrreducible[GF[2, Subscript[g, j]], x];
    AppendTo[primitives, Subscript[f, j]];
    AppendTo[primeList, CoefficientList[Subscript[f, j], x]];
   j++;
   Subscript[f, j] = FieldIrreducible[GF[2, Reverse[Subscript[g, j - 1]]], x];
   AppendTo[primitives, Subscript[f, j]];
   AppendTo[primeList, CoefficientList[Subscript[f, j], x]];
   j++
   ];
  primitives = Delete[primitives, 1];
  primeList = Delete[primeList, 1];
  ]
```

```
gencheck[n_, q_] := Module[{},
  powers = {1};
  modver[q];
  prime[n];
  If[! Divisible[n, Log2[Subscript[m, e] + 1]],
   Print[StringForm["`` not divisible by ``", n,
     Log2[Subscript[m, e] + 1]]],
   For[i = 76, i <= 200, i++,
    PowerListQ[GF[2, primeList[[i]]]] = True;
    powers = FieldInd[GF[2, primeList[[i]]][#]] & /@ stringB;
    c = Mod[A.powers, Subscript[m, e]];
    If[! MemberQ[Subtract[ll, c], 0], Print[primitives[[i]]]]]]
```

References

[1] S.H.Reiger, "Codes for the correction of 'clustered' errors", *IRE Transactions on Information Theory*, vol IT-6, March 1960

[2] H.C.A. van Tilborg, "Fundamentals of cryptology: a professional reference and interactive tutorial",*Eindhoven University of Technology*,

[3] L.E. Dickson, "On the cyclotomic function", *The American Mathematical Monthly*, vol 12 No. 4, April 1905

[4] Khaled A.S. Abdel-Ghaffar, Robert J. McEliece, Andrew M. Odlyzko, Henk van Tilborg, "On the existence of optimum cyclic burst-correcting codes", *IRE Transactions on Information Theory*, vol. IT-32, Nov 1986

[5] Arjeh M. Cohen, Hans Cuypers, Hans Sterk, "Algebra interactive", *Eindhoven University of Technology*, 1991

[6] J.I. Hall, "Notes on Coding Theory" , *Department of Math. Michigan State University*, 1986

[7] B. Elspas, R.A. Short, "A note on optimum burst-error-correcting codes" ,*IRE Transactions on Information Theory*, vol. IT-8, Jan 1962

[8] Todd K. Moon, "Error correction coding: mathematical methods and algorithms",*John Wiley and Sons*, 2005