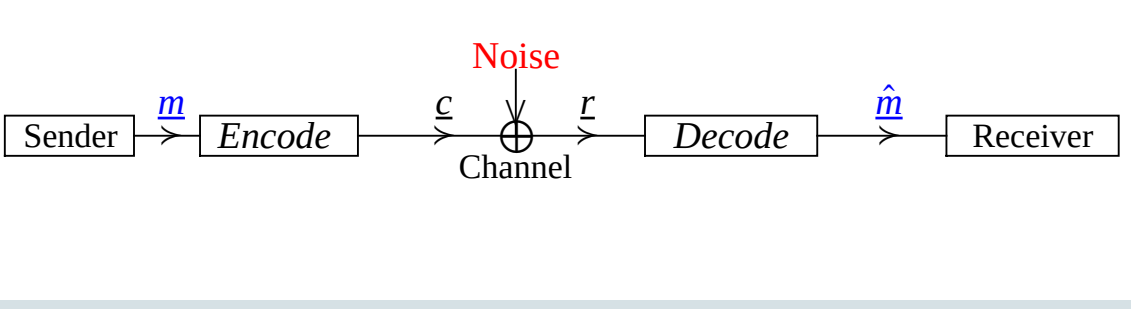
Old and New(er) Results in the Theory of Burst-Correcting Codes

Henk van Tilborg

SPCodingSchool
Campinas, BRASIL
January, 2015

I Error-correcting codes



Codes (here always binary and linear) are mostly used to correct *random* errors, each one independent of the others.

But not always.

A code is a subset of $\{0,1\}^n$. The elements are called codewords. They are said to have length n.

Suppose that each two codewords differ in at least d coordinates (the distance between them is at least d) and put $e = \lfloor \frac{d-1}{2} \rfloor$.

Then the code C is said to be e-error-correcting, because if you transmit (or store) a codeword and not more than e errors have occurred upon reception (or read out) due of noise or damage, then the received word will still be closer to the original codeword than to any other.

Here the codes will be linear, meaning that C is a linear subspace of $\{0,1\}^n$. We use the notation $[n,k,d]$ codes, where k denotes the dimension of the code C.

The quantity $r = n - k$ is called the redundancy of the code. This is the number of additional coordinates (apart from the actual information being transmitted) that make error-correction possible.

A $[7, 4, 3]$ code is given by

$\underline{m}_0$	0	0	0	0	0	0	0	$\underline{c}_0$
$\underline{m}_1$	0	0	0	1	1	1	1	$\underline{c}_1$
$\underline{m}_2$	0	0	1	0	0	1	1	$\underline{c}_2$
$\underline{m}_3$	0	0	1	1	1	0	0	$\underline{c}_3$
$\underline{m}_4$	0	1	0	0	1	0	1	$\underline{c}_4$
$\underline{m}_5$	0	1	0	1	0	1	0	$\underline{c}_5$
$\underline{m}_6$	0	1	1	0	1	1	0	$\underline{c}_6$
$\underline{m}_7$	0	1	1	1	0	0	1	$\underline{c}_7$
$\underline{m}_8$	1	0	0	0	1	1	0	$\underline{c}_8$
$\underline{m}_9$	1	0	0	1	0	0	1	$\underline{c}_9$
$\underline{m}_{10}$	1	0	1	0	1	0	1	$\underline{c}_{10}$
$\underline{m}_{11}$	1	0	1	1	0	1	0	$\underline{c}_{11}$
$\underline{m}_{12}$	1	1	0	0	0	1	1	$\underline{c}_{12}$
$\underline{m}_{13}$	1	1	0	1	1	0	0	$\underline{c}_{13}$
$\underline{m}_{14}$	1	1	1	0	0	0	0	$\underline{c}_{14}$
$\underline{m}_{15}$	1	1	1	1	1	1	1	$\underline{c}_{15}$

A linear code C, say of dimension k, is often described by a generator matrix: a matrix G of size $k \times n$ such that its rows form a basis of C. So,

$$C = \{\underline{u}G \mid \underline{u} \in \{0,1\}^k\}.$$

So, C consists of all linear combinations of the rows of G.

If k is large compared to n, it is often advantageous to describe C as the null-space of a $(n-k) \times n$ matrix H called a parity check matrix:

$$C = \{\underline{x} \in \{0,1\}^n \mid H\underline{x}^T = \underline{0}^T\}.$$

Typically, you transmit a codeword $\underline{c}$ and you receive $\underline{c} \oplus \underline{e} = \underline{r}$, where $\underline{e}$ is called the error vector and is caused by the noise. The decoder looks for the closest codeword to $\underline{r}$, i.e. for $\underline{e}$ of lowest weight such that $\underline{r} - \underline{e} \in C$.

Note that $H\underline{r}^T = H\underline{c}^T \oplus H\underline{e}^T = H\underline{e}^T$. The value $H\underline{r}^T$ is called the syndrome of the received word.

Example: The matrix

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

is the parity check matrix of a linear code of length 7 and dimension 4. Moreover, this code can correct a single error ($e = 1,\ d = 3$). We give a decoding algorithm.

Let $\underline{r}$ be a received word.

1) Compute its syndrome $\underline{s}$, i.e. compute $\underline{s}^T = H\underline{r}^T$.

2a) If $\underline{s} = \underline{0}$ one has that $\underline{r} \in C$, so (most likely) no error occurred.

2b) If $\underline{s} \neq \underline{0}$ then $\underline{s}$ is equal to one of the 7 columns in H, say the j-th. Invert the j-th coordinate in $\underline{r}$ to get a word with syndrome $\underline{0}$, i.e. a codeword.

Example continued: Suppose you receive

$$\underline{r} = (1\ 0\ 0\ 0\ 1\ 1\ 1)$$

Its syndrome with

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

is $\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$, which is the 5-th column. The most likely transmitted codeword is

$$\underline{c} = (1\ 0\ 0\ 0\ 0\ 1\ 1)$$

Most codes in practice have the additional property of being cyclic:

$$(c_0, c_1, \ldots, c_{n-1}) \text{ in } C \quad \Rightarrow \quad (c_{n-1}, c_0, \ldots, c_{n-2}) \text{ in } C$$

It makes them a lot easier to use and it is also easier to find good codes. We associate

$$\underline{c} = (c_0, c_1, \ldots, c_{n-1}) \text{ with the polynomial } c(x) = \sum_{i=0}^{n-1} c_i x^i$$

Theorem: For each cyclic code there exists a unique generator polynomial $g(x)$ that divides $x^n - 1$ with the property:

$$c(x) \text{ in } C \quad \text{if and only if} \quad g(x) \text{ divides } c(x),$$
$$\text{if and only if} \quad c(x) \equiv 0 \pmod{g(x)}.$$

The art is to choose $g(x)$ properly to get a code with good properties.

For a cyclic code, the syndrome of a received word $r(x)$ (corresponding to $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$) is given by

$$s(x) \equiv r(x) \pmod{g(x)}$$

For a codeword $c(x)$ the syndrome is 0.

II Burst-correcting codes

Errors in some applications tend to occur in clusters.
With higher transmission rates or higher storage densities this may even be more so in the future.

Most publications about burst-correcting codes go back to the sixties (Fire '59, Abramson '60, Elspas and Short '62, Bahl and Chien Jr. '69, Peterson and Weldon '72.).

But there has been a revived interest in the late eighties (Blaum, Farrell, and van Tilborg '86, Abdel-Ghaffar, McEliece, Odlyzko, and van Tilborg '86, Zhang and Wolf '88).

A burst of length b starting at position i is an error pattern of the form:

$$(0, \ldots\ldots\ldots, 0, \overset{i}{1}, *, \ldots, *, \overset{i+b-1}{1}, 0, \ldots\ldots\ldots, 0)$$

So, the errors are confined to b or less consecutive places.

A cyclic burst of length b is a burst where the coordinates now have to be viewed cyclically, so it may have the form:

$$(\ldots, *, \overset{i+b-1}{1}, 0, \ldots\ldots\ldots\ldots, 0, \overset{i}{1}, *, \ldots).$$

A code C is a (cyclic) b-burst-correcting code if it is capable of correcting all (cyclic) bursts of length up to b.

Theorem A code C is b-burst-correcting if and only if all different bursts of length up to b have different syndromes.

Proof: Suppose that $\underline{b}_1$ and $\underline{b}_2$ are two different burst with the same syndrome and let H denote the parity check matrix of C. Then

$$H\underline{b}_1 = H\underline{b}_2$$

and thus $H(\underline{b}_1 - \underline{b}_2) = \underline{0}$. In other words, $\underline{c} = \underline{b}_1 - \underline{b}_2$ is a codeword! Now note that

$$\begin{aligned} \underline{b}_1 &= \underline{0} + \underline{b}_1 \\ \underline{b}_1 &= \underline{c} + \underline{b}_2 \end{aligned}$$

So, when you receive $\underline{b}_1$, maybe

$\underline{0}$ was the transmitted codeword and the burst that occurred was $\underline{b}_1$, or
$\underline{c}$ was the transmitted codeword and the burst that occurred was $\underline{b}_2$.

There is no way to tell what happened.

Theorem: Let C be a linear code and H its parity check matrix. Then C is a b-burst-correcting code if and only if every two sets of b consecutive columns in H consists of $2b$ linearly independent vectors.

Proof: Just look at such a linear dependency:

$$(\,0\cdots 0\;\; u_1\;\cdots\; u_b\;\; 0\cdots 0\;\; v_1\;\cdots\; v_b\;\; 0\cdots 0\,)$$

$$\left(\,\underline{h}_1\cdots\underline{h}_i\;\; \underline{h}_{i+1}\cdots\underline{h}_{i+b}\;\; \underline{h}_{i+b+1}\cdots\underline{h}_j\;\; \underline{h}_{j+1}\cdots\underline{h}_{j+b}\;\; \underline{h}_{j+b+1}\cdots\underline{h}_n\,\right) = \underline{0}$$

This means that the two different bursts

$$(\,0\cdots 0\;\; u_1\;\cdots\; u_b\;\; 0\qquad\qquad\cdots\qquad\qquad 0\,)$$

and minus

$$(0\qquad\qquad\cdots\qquad\qquad 0\;\; v_1\;\cdots\; v_b\;\;\cdots\qquad\qquad 0\,)$$

have the same syndrome and vice-versa.

□

Corollary (Reiger): Let C be a (cyclic) b-burst-correcting code with redundancy r. Then $r \geq 2b$, i.e. $\mid C \mid \leq 2^{n-2b}$.

There is a further inequality for cyclic b-burst-correcting codes.

Theorem (Abramson): Let C be a cyclic b-burst-correcting code of length n and redundancy r. Then

$$n \leq 2^{r-b+1} - 1.$$

Proof: The number of cyclic bursts of length up to b is given by $1 + n2^{b-1}$.

The number of different syndromes is 2^r.
So,

$$1 + n2^{b-1} \leq 2^r$$

The theorem now follows from

$$\begin{aligned} 1 + n2^{b-1} &\leq 2^r \\ n2^{b-1} &\leq 2^r - 1 \\ n &\leq 2^{r-b+1} - \frac{1}{2^{b-1}} \\ n &\leq 2^{r-b+1} - 1 \end{aligned}$$

where we have used that n in an integer in the last step.

Later on we stall discuss codes that meet this inequality with equality, so-called "optimal codes".

III Some old constructions

The techniques that are most commonly used are

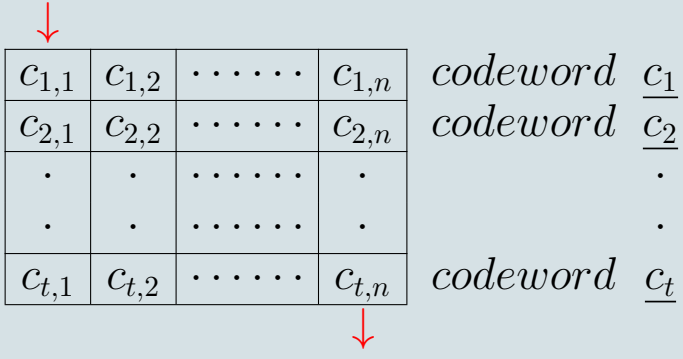
- Interleaving block codes
- Concatenated codes
- Fire codes

We shall discuss these methods here briefly.

III.1 Interleaving a block code

Let C be an $[n, k, d]$ code. Fix a parameter t, the depth of the interleaving process. Let $\underline{c_i}, i \geq 0$, be the list of codewords, that one wants to transmit.

The first tn bits, that are transmitted, are obtained by reading out the columns of the matrix below from top to bottom, starting from the left-most column and continuing to the right. Then transmit the next group, etc.



Now look at:

$r_{1,1}$	$r_{1,2}$	$r_{1,3}$	$r_{1,4}$	$\cdots\cdots$	$r_{1,n}$	$codeword$	$\underline{c_1}$
$r_{2,1}$	$r_{2,2}$	$r_{2,3}$	$r_{2,4}$	$\cdots\cdots$	$r_{2,n}$	$codeword$	$\underline{c_2}$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdots\cdots$	$\cdot$		$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdots\cdots$	$\cdot$		$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdots\cdots$	$\cdot$		$\cdot$
$r_{t,1}$	$r_{t,2}$	$r_{t,3}$	$r_{t,4}$	$\cdots\cdots$	$r_{t,n}$	$codeword$	$\underline{c_t}$

Clearly a burst of length b in the transmitted sequence will affect any particular codeword at most $\lceil b/t \rceil$ times.

So if $\lceil b/t \rceil \leq e$, where C can correct e errors, then this burst pattern can be correctly decoded.

Note that it is even sufficient that C is $\lceil b/t \rceil$-burst-correcting.

In fact, this trick is trivial.

Take for C the binary $[7, 4, 3]$ code. Since $d = 3$, this code can correct a single error. Interleaving at depth 5 gives:

$r_{1,1}$	$r_{1,2}$	$r_{1,3}$	$r_{1,4}$	$r_{1,5}$	$r_{1,6}$	$r_{1,7}$	$codeword\ \underline{c_1}$
$r_{2,1}$	$r_{2,2}$	$r_{2,3}$	$r_{2,4}$	$r_{2,5}$	$r_{2,6}$	$r_{2,7}$	$codeword\ \underline{c_2}$
$r_{3,1}$	$r_{3,2}$	$r_{3,3}$	$r_{3,4}$	$r_{3,5}$	$r_{3,6}$	$r_{3,7}$	$codeword\ \underline{c_3}$
$r_{4,1}$	$r_{4,2}$	$r_{4,3}$	$r_{4,4}$	$r_{4,5}$	$r_{4,6}$	$r_{4,7}$	$codeword\ \underline{c_4}$
$r_{5,1}$	$r_{5,2}$	$r_{5,3}$	$r_{5,4}$	$r_{5,5}$	$r_{5,6}$	$r_{5,7}$	$codeword\ \underline{c_5}$

So, a burst of length up to 5 affects every row at most once and thus can be corrected.

III.2 The concatenated code construction

Let C be an $[n, k, d]$ code but not binary, but over an alphabet with 2^m symbols.

Very good codes of this kind exist because one can equip a set with 2^m elements with a field structure: this means that addition, subtraction, multiplication and division are all possible.

Now write each 2^m-ary symbol as a binary m-tuple. In that way a binary $[nm, km]$ code is obtained.

$$\begin{array}{ccccccccc} (& c_1 & , & c_2 & , & \cdots & , & c_n &) \quad 2^m\text{-ary} \\ & \downarrow & & \downarrow & & & & \downarrow & \\ (& a_1, \ldots, a_m & , & a_{m+1} \ldots, a_{2m} & , & \cdots & , & a_{(n-1)m+1}, \ldots, a_{nm} &) \quad \text{binary} \end{array}$$

On a CD and DVD they use a $[28, 24, 5]$ code over 2^8. With this trick, you get a binary code of length 28×8 and cardinality $(2^8)^{24}$, *i.e.* dimension 24×8.

A burst of length b will affect at most $1+\lceil (b-1)/m \rceil$ consecutive symbols in the 2^m-ary code C. If C is capable of correcting those, we have thus obtained a b-burst-correcting $[nm, km]$ code.

On the CD and DVD, $m = 8$ and the 2^8-code can correct 2 errors (since the $[28, 24, 5]$ has minimum distance 5).

$$\begin{array}{ccccccccc} (& c_1 & , & c_2 & , & \dots & , & c_{28} &) \quad 2^8\text{-ary} \\ & \downarrow & & \downarrow & & & & \downarrow & \\ (& a_1, \dots, a_8 & , & a_9 \dots, a_{16} & , & \dots & , & a_{217}, \dots, a_{224} &) \quad \text{binary} \end{array}$$

So, every burst of length up to 9 can be corrected, but many more patterns can also be corrected.

In fact, they use this trick in combination with some kind of more advanced interleaving.

III.3 Fire codes

Definition: A Fire code '59 is a cyclic code with generator polynomial

$$g(x) = (x^{2b-1} - 1)p(x),$$

where $p(x)$ is an irreducible polynomial of degree m, $m \geq b$, that does not divide $x^{2b-1} - 1$. The block length of the Fire code is the smallest integer n such that $g(x)$ divides $x^n - 1$.

Theorem: The Fire code, as defined above, has block length $\text{lcm}[l, 2b-1]$, where l is the period of $p(x)$. It is b-burst-correcting.

Recall that the redundancy r of a cyclic code satisfies $r = \deg(g(x))$, so the redundancy of the Fire code is given by $r = 2b - 1 + m \geq 3b - 1$. Compare this with the Reiger bound $r \geq 2b$.

The *period* of $p(x)$ is the smallest positive l for which $p(x)|(x^l - 1)$.

Proof by example:

Consider the binary Fire code with $b = 4$, generated by $g(x) = (x^7 - 1)(1 + x + x^4)$.

Since $1+x+x^4$ has period 15, this Fire code has length $n = \text{lcm}[7, 15] = 105$, redundancy $7 + 4 = 11$, and dimension $k = 105 - 7 - 4 = 94$.

Now, let $r(x) = \sum_{i=0}^{104} r_i x^i$ be a received word and let its syndrome (given by $s(x) \equiv r(x) \pmod{g(x)}$) have value

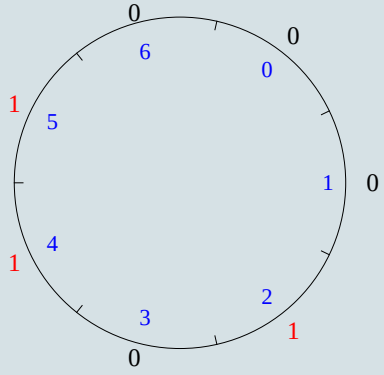
$$x^9 + x^7 + x^5 + x^4 + 1.$$

We reduce this further modulo the two factors of $g(x)$: $x^7 - 1$ and $x^4 + x + 1$ and get

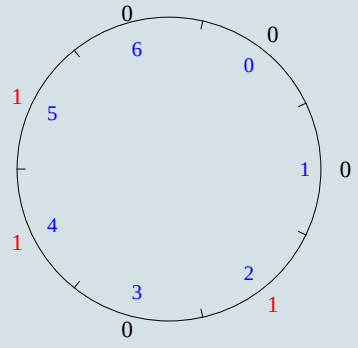
$$\begin{aligned} s_1(x) &\equiv x^2 + x^4 + x^5 \pmod{x^7 - 1}, \\ s_2(x) &\equiv 1 + x^2 \pmod{x^4 + x + 1}. \end{aligned}$$

Remember that $r(x) = c(x) + x^i B(x)$, where $x^i B(x)$ is the burst, $B(0) = B_0 = 1$, $\deg(B(x)) < 4$, $0 \leq i < 105$ and where $c(x)$ is a codeword (so it has syndrome 0). The syndrome comes from $x^i B(x)$.

How can $x^i B(x) \equiv x^2 + x^4 + x^5 \pmod{x^7 - 1}$, when $\deg(B(x)) < 4$?



Since the burst has length at most 4, there will be a gap along the circle of length at least 3. Note that there can not be two gaps of length at least 3 (separated by ones) along the circle.



The gap starts at position 6 and ends at position 1. So the burst starts at position 2. We conclude that

$$B(x) = 1 + x^2 + x^3 \quad \text{and} \quad i \equiv 2 \pmod 7.$$

From the second syndrome

$$(1 + x^2 + x^3)x^{2+7u} \equiv 1 + x^2 \pmod{x^4 + x + 1}$$

one finds $u = 14$. So, the actual burst that occurred was: $(1 + x^2 + x^3)x^{100}$.

IV New(er) developments

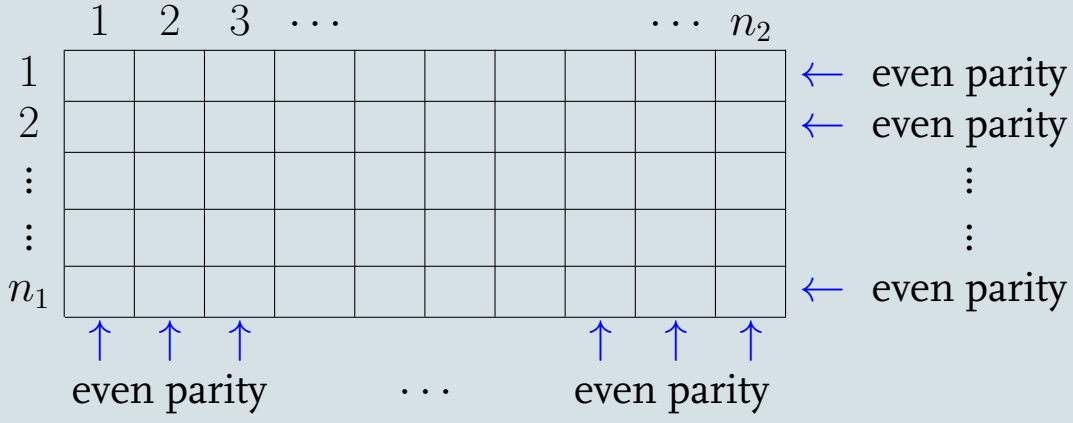
About 25 years later two other classes of burst correcting codes received a lot of attention.

1. Array codes
2. Optimal cyclic burst-correcting codes

They form the main topic of this presentation.

IV.1 Array codes

Definition: An (n_1, n_2)-array code $\mathcal{C}$ consists of all $n_1 \times n_2$ $\{0,1\}$-arrays C whose row and column sums are all congruent to zero modulo 2.



It follows directly from this definition that an (n_1, n_2) array code $\mathcal{C}$ is a linear code with length $n_1 \times n_2$, dimension $(n_1-1)(n_2-1)$, and minimum distance 4.

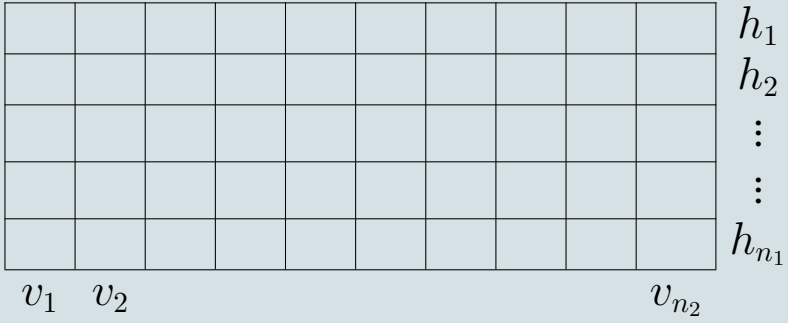
Example: $n_1 = 5,\ n_2 = 8$.

0	1	0	1	1	1	0	0
1	1	1	1	0	1	1	0
1	0	1	0	0	0	1	1
0	0	0	1	0	1	1	1
0	0	0	1	1	1	1	0

is a "codeword".

This code has length $5 \times 8 = 40$ and dimension $4 \times 7 = 28$.

Let R be a received word.



The **horizontal** and **vertical syndrome** of R are defined by:

$$h_i = \sum_{j=1}^{n_2} r_{ij}, 1 \leq i \leq n_1, \quad \text{resp.} \quad v_j = \sum_{i=1}^{n_1} r_{ij}, 1 \leq j \leq n_2.$$

Decoding a single error in this code is extremely simple, because an error at location (i, j) results in the syndrome

$$\underline{h} = (0, \ldots, 0, \overset{i}{1}, 0, \ldots, 0)^T \text{ and } \underline{v} = (0, \ldots, 0, \overset{j}{1}, 0, \ldots, 0).$$

Example continued:

Look at the received word:

1	1	0	0	0	1	0	1	0
0	1	0	0	1	0	1	0	1
1	0	1	0	1	1	0	0	0
0	0	1	1	0	1	1	0	0
0	0	1	1	0	1	0	1	0
0	0	1	0	0	0	0	0	

It is clear where the error occurred.

So, decoding a single error is easy (but not very impressive).

How about decoding bursts?

For burst-correction the particular read-out of the array is important. We follow diagonals, one after another.

Example: $n_1 = 5,\ n_2 = 6$, so $n = 30$.

0	5	10	15	20	25
26	1	6	11	16	21
22	27	2	7	12	17
18	23	28	3	8	13
14	19	24	29	4	9

Without loss of generality we shall assume that $n_2 \geq n_1$.

It is not so difficult to see that $\mathcal{C}$ cannot correct all bursts of length up to n_1.

Indeed, in our example, the two bursts of length 5 indicated below (and many more) have the same syndrome.

0	5	10	15	20	25
26	1	6	11	16	21
22	27	2	7	12	17
18	23	28	3	8	13
14	19	24	29	4	9

and

0	5	10	15	20	25
26	1	6	11	16	21
22	27	2	7	12	17
18	23	28	3	8	13
14	19	24	29	4	9

Both have burst-pattern $(1, 0, 0, 0, 1)$ and the positions of the ones have been indicated in color.

Let us now see when $\mathcal{C}$ can correct all bursts of length $\leq n_1 - 1$.

With a little bit of work one can check that for $n_2 < 2n_1 - 3$ there are always two different weight-2 bursts of length $\leq n_1 - 1$ with the same syndrome.

For instance the two bursts depicted below in red resp. blue have the same syndrome.

0	5	10	15	20	25	1
26	1	6	11	16	21	0
22	27	2	7	12	17	0
18	23	28	3	8	13	1
14	19	24	29	4	9	0
1	0	0	1	0	0	

Theorem: Let $\mathcal{C}$ be the $n_1 \times n_2$ array code, $n_2 \geq n_1$, with $+1$-diagonal read-out as defined above. Then $\mathcal{C}$ can correct all bursts of length $\leq n_1 - 1$ if and only if

$$n_2 \geq 2n_1 - 3.$$

Proof by example: $n_1 = 5,\ n_2 = 7$. (Note that $n_2 = 7 = 2 \times 5 - 3 = 2n_1 - 3$.)

Let the syndrome of a received word be given as below.

0	5	10	15	20	25	30	1
31	1	6	11	16	21	26	0
27	32	2	7	12	17	22	1
23	28	33	3	8	13	18	1
19	24	29	34	4	9	14	0
1	1	0	0	0	1	0	

0	5	10	15	20	25	30	1
31	1	6	11	16	21	26	0
27	32	2	7	12	17	22	1
23	28	33	3	8	13	18	1
19	24	29	34	4	9	14	0
1	1	0	0	0	1	0	

Observation 1: Each burst of length $\leq n_1 - 1\ (= 4)$ affects each row and also each column of the array at most once.

So, cancelation of ones does not occur during the computation of the horizontal and vertical syndromes.

So, the top-row is affected by the burst, the second row is not, etc.

Similarly, the first two columns are affected by the burst, the third not, etc.

0	5	10	15	20	25	30	1
31	1	6	11	16	21	26	0
27	32	2	7	12	17	22	1
23	28	33	3	8	13	18	1
19	24	29	34	4	9	14	0
1	1	0	0	0	1	0	

Observation 2: A burst of length $\leq n_1 - 1\ (= 4)$ affects at most $n_1 - 1$ cyclically consecutive columns. So, at least $n_2 - n_1 + 1\ (= 3)$ (cyclically) consecutive vertical syndromes v_i will be zero.

By Observation 1 we know that the corresponding columns are error-free. So columns 3, 4 and 5 are error free.

There cannot be two error-free gaps of length $\geq (n_2 - n_1 + 1)$, both flanked by ones, because $2(n_2 - n_1 + 1) + 2 \leq n_2$ ($= \sharp$ columns) implies that $n_2 \leq 2n_1 - 4$. A contradiction!

0	5	10	15	20	25	30	1
31	1	6	11	16	21	26	0
27	32	2	7	12	17	22	1
23	28	33	3	8	13	18	1
19	24	29	34	4	9	14	0
1	1	0	0	0	1	0	

We conclude that there is a unique gap of length $\geq n_2 - n_1 + 1\ (= 3)$ in $(v_1, v_2, \ldots, v_{n_2})$ when viewed cyclically.

Let the error-free gap, found above, end in column $v - 1$. Above $v - 1 = 5$, so $v = 6$.

We now have

$$v_i \neq 0 \Rightarrow i \in \{v, v+1, \ldots, v+n_1-2\} \quad \text{modulo } n_2.$$

Above: $v_i \neq 0 \Rightarrow i \in \{6, 7, 1, 2\}$.

0	5	10	15	20	25	30	1 u
31	1	6	11	16	21	26	0
27	32	2	7	12	17	22	1
23	28	33	3	8	13	18	1
19	24	29	34	4	9	14	0
1	1	0	0	0	1	0	

$\leftarrow$ gap $\rightarrow$ v

Observation 3: Let u be the index of the first row from the top with non-zero syndrome (the top row here). Remember that $v = 6$ in this example.

Claim: position (u, v) is in error.

Indeed, row u and column v both must contain one error. If this error in not on position (u, v), there must be an error on a position (i, v) with $i > u$ and an error on a position (u, j) with $j \epsilon \{v + 1, v + 2, \ldots, v + n_1 - 2\}$. However, these positions cannot be in the same burst of length $\leq n_1 - 1$.

0	5	10	15	20	25	30	1
31	1	6	11	16	21	26	0
27	32	2	7	12	17	22	1
23	28	33	3	8	13	18	1
19	24	29	34	4	9	14	0
1	1	0	0	0	1	0	

The other error positions of the burst are now easily found with the same rule: 27 and 28.

0	5	10	15	20	25	30	1
31	1	6	11	16	21	26	0
27	32	2	7	12	17	22	1
23	28	33	3	8	13	18	1
19	24	29	34	4	9	14	0
1	1	0	0	0	1	0	

To judge the efficiency of array codes we take n_2 minimal, so $n_2 = 2n_1 - 3$, and compute the redundancy.

The redundancy r is given by $r = n_2 + n_1 - 1 = 3n_1 - 4$, while the Reiger bound gives $2b = 2n_1 - 2$ as lower bound. So, the prize paid for the simplicity of the coding and decoding consists of at most $(n_1 - 2)$ bits of information.

Other read-outs have also been studied. In particular, the (-1)-read-out: follow a diagonal and then go to the preceding one.

Zhang and Wolf generalize this to the s-read-out, where $\gcd(s, n) = 1$.

0	25	15	5	30	20	10
11	1	26	16	6	31	21
22	12	2	27	17	7	32
33	23	13	3	28	18	8
9	34	24	14	4	29	19

$s = 3$.

IV.2 Optimal, cyclic burst-correcting codes

Definition: A cyclic, b-burst-correcting code of length n is called optimal if its redundancy r satisfies the Abramson inequality with equality:

$$n = 2^{r-b+1} - 1.$$

So, an optimal, cyclic, b-burst correcting code has length $n = 2^m - 1$ and redundancy $r = m + b - 1$. The Reiger bound, $r \geq 2b$, yields

$$m \geq b + 1.$$

We know already such codes for $b = 1$: the 1-error-correcting codes with parity check matrix

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

and generator polynomial $g(x) = m_1(x)$, the minimal polynomial (of degree m) of the primitive element α in $GF(2^m)$.

Note that $r = m$.

These are the well known (binary) Hamming codes.

We shall denote a primitive polynomial of degree m by $p(x)$ from now on.

Consider the cyclic code C with generator polynomial $g(x) = (x+1)p(x)$, or, equivalently,

$$C = \{c(x) \mid c(1) = c(\alpha) = 0\}.$$

Its parity check matrix is given by:

$$H = \begin{pmatrix} 1 & 1 & \cdots & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^i & \cdots & \alpha^{n-1} \end{pmatrix}.$$

Code C has length $n = 2^m - 1$ and minimum distance $d = 4$, as $g(x)$ is divisible by $(x-1)(x-\alpha)(x-\alpha^2)$ and thus has 3 consecutive zeros (BCH bound).

We shall show that it is an optimal, cyclic, 2-burst correcting code!

Note that indeed $r = m + 1$.

$$H = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{i-1} & \alpha^{i} & \alpha^{i+1} & \alpha^{i+2} & \cdots & \alpha^{n-1} \end{pmatrix}.$$

The decoding of a burst of length ≤ 2 is easy.

Let $r(x)$ be the received word and $s_0 = r(1)$ and $s_1 = r(\alpha)$ its syndrome.

- Clearly, if no error (burst) occurred, the syndrome will be $s_0 = s_1 = 0$.
- A burst of length 1 is the same as a single error.
  If this occurred at the i-th coordinate, one has: $s_0 = 1$ and $s_1 = \alpha^i$.
- And a burst of length 2, say at coordinates i and $i+1$, will have syndrome $s_0 = 1 + 1 = 0$ and $s_1 = \alpha^i + \alpha^{i+1} = \alpha^i(1 + \alpha)$.

It is easy to distinguish these cases and to determine i from the syndrome.

Abramson '60 states that $g(x) = (1+x+x^2)p(x)$, where $p(x)$ is a primitive polynomial of even degree $m \geq 4$, generates an optimal, cyclic, 3-burst-correcting code, if a defined by

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

satisfies

$$a \not\equiv 2 \pmod{3}.$$

Note that $r = m + 2$.

Abramson gives such codes for $m = 4, 6, 8$ and 10 and conjectures that they always exist for even m, $m \geq 4$.

For instance, $(1 + x + x^2)(1 + x + x^6)$ generates a optimal, cyclic, 3-burst-correcting code of length $n = 63$.

That conditions like

"the exponent m in $n = 2^m - 1$ must be even"

"the a defined by $1 + x \equiv x^a \pmod{p(x)}$ must satisfy $a \not\equiv 2 \pmod{3}$."

come up is not so surprising.

First of all, the generator polynomial $g(x) = (1 + x + x^2)p(x)$ must divide $x^n - 1$, i.e. $x^{2^m-1} - 1$.

But $1+x+x^2 = (x^3-1)/(x-1)$ divides $x^{2^m-1}-1$ if and only if $3|(2^m-1)$, i.e. if and only if m is even.

Remember from the discussion of Fire codes that each burst can be put in polynomial notation

$$x^i B(x),$$

with $B(0) = 1$ and degree$(B(x)) < b$.

$B(x)$ reflects the pattern of the burst and i the coordinate where the bursts starts.

Example: $n = 7$
Burst

$$0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0$$

starts at coordinate 3, has pattern 101, and is denoted by $x^3(1 + x^2)$.

To understand the second condition, observe that all bursts must have different syndromes, so their difference cannot be equal to a codeword. Consider the two bursts 1 and $x^u(1+x)$. Apparently

$$(1+x+x^2)p(x) \quad \not| \ (1+x^u(1+x)).$$

This can be rewritten as:

$$(1+x+x^2) \mid (1+x^u(1+x)) \quad \Longrightarrow \quad p(x) \ \not| (1+x^u(1+x)).$$

But the condition $(1+x+x^2) \mid (1+x^u(1+x))$ is equivalent with the condition $u \equiv 1 \pmod 3$.

So, we have

$$u \equiv 1 \pmod 3 \quad \Longrightarrow \quad p(x) \ \not| (1+x^u(1+x)).$$

$$u \equiv 1 \pmod 3 \quad \Longrightarrow \quad p(x) \not| (1 + x^u(1 + x)).$$

Since $p(x)$ is a primitive polynomial $1 + x \equiv x^a \pmod{p(x)}$ for some $a,\ 0 \leq a < n$.

In this way one obtains the condition that $p(x)$ does not divide $1 + x^{a+u}$ for any u with $u \equiv 1 \pmod 3$.

By the primitivity of $p(x)$ we know that $p(x) \mid (1+x^{a+u})$ if and only if $a+u \equiv 0 \pmod n$.

So, for each $u \equiv 1 \pmod 3$ we have the condition that $a+u \not\equiv 0 \pmod n$. But $n \equiv 0 \pmod 3$.

Apparently $a \not\equiv 2 \pmod 3$, which is exactly the condition of Abramson.

Elspas and Short '62 state necessary conditions on the generator polynomial $g(x)$ of optimal, cyclic, burst-correcting codes for the general case.

Let $e(x)$ be a polynomial without repeated factors and assume that $e(0) = 1$.

Then the smallest exponent m such that $e(x)$ divides $x^{2^m-1} - 1$ is well-defined and will be denoted by m_e. It is called the degree of the splitting field of $e(x)$.

It follows that $e(x) \mid (x^{2^m-1} - 1)$ if and only if $m_e \mid m$.

For example, $e(x) = 1 + x + x^2$ divides $x^3 - 1 = x^{2^2-1} - 1$, so $m_e = 2$.
And thus $(1 + x + x^2) \mid (x^{2^m-1} - 1)$ if and only if $2|m$.

And $e(x) = 1 + x + x^3$ divides $x^7 - 1 = x^{2^3-1} - 1$, so $m_e = 3$.
And thus $(1 + x + x^3) \mid (x^{2^m-1} - 1)$ if and only if $3|m$.

Theorem Let $g(x)$ be the generator polynomial of an optimal, cyclic, b-burst correcting code of length $n = 2^m - 1$. Then $g(x)$ can be factored into $e(x)p(x)$, where

1. $e(x)$ is a square-free polynomial of degree $b-1$ and $e_0 \neq 0$.
2. $p(x)$ is a primitive polynomial of degree m, $m \geq b+1$, such that $m_e | m$, where m_e is smallest integer with $e(x) \mid (x^{2^{m_e}-1} - 1)$.

Elspas and Short give no proof.

Conditions 1) and 2) partly follow from the fact that $e(x)p(x)$ has to divide $x^n - 1 = x^{2^m-1} - 1$, as this polynomial has no repeated factors and is not divisible by x.

A proof of the other assertions in 1) and 2) is given by Abdel-Ghaffar, McEliece, Odlyzko, and van Tilborg (AMOT) in 1986.

Elspas and Short do find all optimal, cyclic, 3-burst correcting codes generated by $(1+x^2)p(x)$ for $m = 4, 6, 8, 10$ and $m = 12$.

Similarly, they find all optimal, cyclic, 4-burst correcting codes generated by $(1+x^3)p(x)$ for $m = 10$ and 12.

For $m = 6$ and 8 they do not exist. (Note that m has to be divisible by 2 and is at least 5.)

And they find all optimal, cyclic, 4-burst correcting codes generated by $(1+x+x^3)p(x)$ for $m = 9$ and 12.

For $m = 6$ they do not exist. (Note that m has to be divisible by 3 and is at least 5.)

Elspas and Short only give a necessary condition for the existence of an optimal, b-burst-correcting code.

Theorem [AMOT]: A polynomial $g(x)$ generates an optimal, cyclic, b-burst-correcting code of length $n = 2^m - 1$ if and only if $g(x)$ can be factored into $e(x)p(x)$, where:

1. $e(x)$ is a square-free polynomial of degree $b-1$ and $e(0) \neq 0$.
2. $p(x)$ is a primitive polynomial of degree m, $m \geq b+1$, such that $m_e | m$, where m_e is the degree of the splitting field of $e(x)$.
3. $p(x)$ satisfies the AES conditions associated with $e(x)$.

We will not give a description of the AES conditions (named after Abramson, Elspas and Short), but just say that they can be obtained by comparing all possible pairs of different bursts patterns of length up to b.

Of course we like necessary and sufficient conditions for the existence of optimum, cyclic b-burst correcting codes!

Example

In case that $b = 4$ and $e(x) = 1 + x^3$ there are the following four AES conditions:

$a \not\equiv 2 \pmod 3$, where a is defined by $1 + x \equiv x^a \pmod{p(x)}$,

$b \not\equiv 1 \pmod 3$, where b is defined by $1 + x + x^3 \equiv x^b \pmod{p(x)}$,

$c \not\equiv 2 \pmod 3$, where c is defined by $1 + x^2 + x^3 \equiv x^c \pmod{p(x)}$,

and also

$b + 2c \not\equiv 2 \pmod 3$.

These conditions can be found in a similar way as in the $b = 3$ and $e(x) = 1 + x + x^2$ case before: compare all burst patterns of length up to 4 and require that they all have different syndromes.

$$a \not\equiv 2 \pmod 3, \quad \text{with } 1 + x \equiv x^a \pmod{p(x)},$$
$$b \not\equiv 1 \pmod 3, \quad \text{with } 1 + x + x^3 \equiv x^b \pmod{p(x)},$$
$$c \not\equiv 2 \pmod 3, \quad \text{with } 1 + x^2 + x^3 \equiv x^c \pmod{p(x)},$$
$$b + 2c \not\equiv 2 \pmod 3.$$

Possible solutions modulo 3 are

$$\{a, b, c\} = \{0, 0, 0\}, \{0, 2, 1\}, \{1, 0, 0\}, \{1, 2, 1\}$$

The big question is if a primitive polynomial exists that meets these requirements.

One may hope so as the number of primitive polynomials grows exponentially in the degree.

Theorem [AMOT]: Let $e(x)$ be a square free polynomial of degree $b-1$ with $e(0) \neq 0$. Let m_e be the degree of the splitting field of $e(x)$.

Then for all sufficiently large m with $m_e|m$, a primitive polynomial $p(x)$ of degree m exists such that $g(x) = e(x)p(x)$ generates an optimal, cyclic, b-burst-correcting code.

The proof consists of two parts.

First, it is shown that the AES-conditions can never be self-contradictory.

Secondly, by making use of Weil's estimates, it is shown that for sufficiently large m, with $m_e|m$, one can always find primitive polynomials of degree m, satisfying the relations corresponding to the AES-conditions.

This theorem implies that Abramson's conjecture about the existence of optimal, cyclic, 3-burst-correcting codes is true for sufficiently large even values of m. By making careful estimates in the proof of this theorem and by making use of the computer the following results were also established.

Theorem: For every even m, $m \geq 4$, there exists an optimal, cyclic, 3-burst-correcting code of length $2^m - 1$ with generator polynomial $g(x) = (1 + x + x^2)p(x)$, where $p(x)$ is a primitive polynomial of degree m.

Theorem: For every even m, $m \geq 10$, there exists an optimal, cyclic, 4-burst-correcting code of length $2^m - 1$ with generator polynomial $g(x) = (1 + x^3)p(x)$, where $p(x)$ is a primitive polynomial of degree m.

Finally we want to mention that $(1 + x)(1 + x + x^3)p(x)$, with $p(x) = 1 + x + x^2 + x^3 + x^5 + x^9 + x^{10} + x^{13} + x^{15}$, generates an optimal, cyclic, 5-burst correcting code of length $2^{15} - 1$.

V Research directions

- Multiple-burst correcting codes.
- Two-dimensional burst correcting codes.
- Efficient decoding of optimal, cyclic burst correcting codes.
- Improve on the very pessimistic bound in AMOT.
- ...

Proof of 1) and 2) :

From $g(x) \mid (x^n - 1 = (x^{2^m-1} - 1)$, it follows that $g(x)$ has a period dividing $n = 2^m - 1$.

If this period, say p, is less than $2^m - 1$, it follows, because $g(x) \mid (x^p - 1)$, that $x^p - 1$ is a codeword.

But this implies that bursts 1 and x^p will have the same syndrome. A contradiction with the assumption that the code is b-burst-correcting.

We conclude that

$g(x)$ has period $2^m - 1$.

Write $g(x) = f_1(x) \cdots f_l(x)$ (distinct, irreducible polynomials).

The period u of a polynomial $g(x)$ is the smallest positive integer for which $g(x)$ divides $x^u - 1$.

$$g(x) = f_1(x) \cdots f_l(x)$$

Define

$$\begin{aligned} r_i &= \text{degree of } f_i(x) \\ h_i &= \text{period of } f_i(x) \end{aligned}$$

From $f_i(x) \mid g(x) \mid (x^{2^m-1} - 1)$ it follows that $r_i | m$.

From $f_i(x) \mid (x^{2^{r_i}-1} - 1)$ it follows that $h_i | (2^{r_i} - 1.)$

Since $g(x)$ has period $2^m - 1$ we have

$$\begin{aligned} 2^m - 1 &= \text{period of } g(x) = \text{lcm}_{1 \le i \le l} h_i \\ 2^m - 1 &\mid \text{lcm}_{1 \le i \le l} (2^{r_i} - 1) \end{aligned}$$

Since $r_i | m, 1 \le i \le l$, it follows (number theory) that $r_i = m$ for at least one value of i, say for $i = 1$.

$g(x) = f_1(x)f_2(x)\cdots f_l(x)$ (irreducible and distinct)

$f_1(x)$ has degree m (will be named $p(x)$)

No other $f_i(x)$ can have degree m, because of the Reiger bound:

Indeed, $m + b - 1 = r \geq 2m$ implies $b \geq m + 1$,
while $r = m + b - 1 \geq 2b$ implies that $m \geq b + 1$.
A contradiction!

Write $e(x) = f_2(x)\cdots f_l(x)$ (its degree is $b - 1$)

It remains to prove: $f_1(x)$ has period $2^m - 1$.

If h_1 (the period of $f_1(x)$) is less than $2^m - 1$, then $e(x)(x^{h_1} - 1)$ has degree less than $n = 2^m - 1$.

Trivially $g(x) = e(x)f_1(x)$ divides $e(x)(x^{h_1} - 1)$, so $e(x)(x^{h_1} - 1)$ is a codeword!

Apparently the two distinct bursts $e(x)$ and $e(x)x^{h_1}$ have the same syndrome. A contradiction!

Conclusion: $f_1(x)$ is a polynomial of degree m which has period $n = 2^m - 1$. So, $f_1(x)$ is primitive polynomial.

□

Outline of proof of 3).

$a \equiv b \equiv c \equiv \ldots \equiv 0 \pmod{2^{m_e} - 1}$
is always a solution of the AES conditions. In other words, the AES conditions are not self-contradictory.

The number of primitive polynomials of degree m grows exponentially in m.
By making use of Weil's estimates, one can show that for sufficient large m primitive polynomials exist, which do meet $a \equiv b \equiv c \equiv \ldots \equiv 0 \pmod{2^{m_e} - 1}$.