Fifteenth International Workshop on Algebraic and Combinatorial Coding Theory
June 18-24, 2016, Albena, Bulgaria *pp. 200–205*

On the Mollard code as a partially robust code [1]

D.I. Kovalevskaya daryik@rambler.ru
State University of Aerospace Instrumentation, Saint Petersburg, Russia

Abstract. In this paper a generalization of classic Mollard construction for any code length is given. It is shown that such generalized codes have the property of partial robustness. These codes have less undetectable and miscorrected errors than the traditional linear error-correcting codes, therefore, they are more useful for the detection of multiple and repeating errors. It is also shown that the generalized Mollard construction has some advantages in providing better protection against multiple bit errors over the shortened Vasil'ev code, for some code parameters.

1 Introduction

Classic linear perfect codes, which correct single errors and detect double errors, are usually used to increase the reliability of systems – in sense of protection them against soft errors. Such codes are concentrated on a small subset of the most probable errors (of small multiplicity) – they guarantee the error detection for the errors, which multiplicity is less then the code distance d, but the detection of errors which multiplicity is greater than d, is unpredictable and inefficient. So, in case of multiple or repeating errors occurrence – i.e. hard errors, caused by permanent faults – the reliability of the memory which protection is based on classic linear perfect codes, cannot be guaranteed. As the probability of multiple errors occurrence becomes higher in the presence of new technologies (for example, a flash-memory effected by space radiation), some authors (see, for example, [1]) propose to consider so called robust codes, which guarantee a certain level of error detection.

Recall that a binary *perfect* (or *closely packed*) code C^n of length $n = 2^m - 1$, $m \geq 2$, with the code distance 3 (which correct only single errors) is a code of power $|C^n| = \frac{2^n}{n+1}$. The linear binary perfect code of length n and code distance 3 – i.e. the Hamming code – is usually denoted as $\mathcal{H}^n$.

For any code $D \subset \mathbf{F}^n$ a *detection kernel* (see [1]) is defined as the set of masked errors for all of the codewords: $Ker_d(D) = \{e \in \mathbf{F}^n | e + d \in D, \forall d \in D\}$. If D is a linear code, then $Ker_d(D) = D$.

Let Alg_D be an error correcting algorithm for the code D. If D_{er} is a set of errors which Alg_D tries to correct, then a *correction kernel* (see [1]) is defined

[1] The research is supported by RD N 2.2716.2014/K "Development and investigation of reliable methods of storage in the aerospace systems and complexes"

as $Ker_c(D) = \{e \in \mathbf{F}^n | e \notin D_{er}, d \in D, e' \in D_{er}, Alg_D(e,d) = Alg_D(e',d)\}$. The correction kernel of the Hamming code is $Ker_c(\mathcal{H}^n) = \{e \in \mathbf{F}^n | \exists i, 1 \leq i \leq n : He^T = He_i^T\}$, where e_i is the vector of weight 1 with nonzero i-th coordinate position.

A *robust code* (see [1]) is such a code $D \subset \mathbf{F}^n$ which $Ker_d(D) = 0$. This also means that $max_{(x \in \mathbf{F}^n \setminus \{0\})} Q_D(x) < 1$, where $Q_D(x) = \frac{|d \in D : d+x \in D|}{|D|}$ is the *error masking probability* of x. There are no undetectable errors for the robust codes. What is more, their minimum distance is at most 1, therefore robust codes cannot be used for the error correction. A *partially robust code* (see [1]) is a systematic $(n, 2^k, d)$-code D, which detection kernel is smaller than 2^k: $|Ker_d(D)| < 2^k$. Such codes have undetectable errors, but the number of such errors decrease by several orders in comparison with the linear perfect codes. At the same time, partially robust codes keep some structure of linear codes. For any code D, *the error masking probability* is defined as $Q_{mc}(D) = max_{(e \notin Ker_d(D))} Q_D(e)$. These notions – Ker_c, Ker_d and Q_{mc} – are the characteristics allowing to compare codes, which can be effective for the correcting and detecting multiple errors.

Nonlinearity of some function $f : \mathbf{F}^k \to \mathbf{F}^s$ can be measured by means of its derivative $D_v f(x) = f(x+v) + f(x)$, where $v \in \mathbf{F}^k$. If $Pr(E)$ is the *probability of the event E occurrence*, then the *measure of the function f nonlinearity P_f* can be defined as follows: $P_f = max_{v \in \mathbf{F}^k \setminus \{0\}} max_{b \in \mathbf{F}^s} Pr(D_v f(x) = b)$.

One method of constructing large classes of nonlinear codes with different properties is a switching method. A code $C' = (C \setminus R) \cup R'$ is obtained by a *switching* of some set R with a set R' in a binary code C, if the code C' has the same parameters as C. The first switching code construction was given by Yu.L. Vasil'ev in [3]: if C^s is any perfect binary code of length s, and $f : C^s \to \{0,1\}$ is some boolean function, then the set $V^{2s+1} = \{(x+c, |x| + f(c), x) : x \in \mathbf{F}^s,\ c \in C^s\}$ is a perfect binary code of length $2s+1$ with the code distance 3. If f is a nonlinear function, then the code V^{2s+1} is nonlinear one.

It is proved in [1] that the Vasil'ev code V^{2s+1} is a partially robust code with the power of detection kernel $|Ker_d(V^{2s+1})| = 2^s$ and the error masking probability $Q_{mc}(V^{2s+1}) = P_f$. Therefore, $|Ker_d(V^{2s+1})| = 2^s < 2^{2s - \log_2(s+1)} = |Ker_d(H^{2s+1})|$, and the Vasil'ev code has less undetectable errors than the Hamming code. Also, M. Karpovsky, K.J. Kulikowski and Z. Wang in [1] proved partially robustness for the generalization of extended Vasil'ev codes. Such codes (called shortened Vasil'ev codes $\bar{V}^{n=a+s+2}$) exist for any code length $n \geq 4$. Bounds for the error masking probability, and the number of undetectable and miscorrected errors are found in that paper. The given classic switching construction of Vasil'ev was further generalized by M. Mollard.

In this paper, the Mollard construction [2] is considered. It is proved that the Mollard code is a partially robust code. Also a generalization of such classic construction, error correcting algorithm and comparative analysis of corresponding characteristics of Mollard and Vasil'ev codes are given.

2 The generalized Mollard construction

An arbitrary vector $x \in \mathbf{F}^{tm}$ can be written down as follows: $x = (x_{11}, x_{12}, \ldots, x_{1m}, x_{21}, x_{22}, \ldots, x_{2m}, \ldots, x_{t1}, x_{t2}, \ldots, x_{tm})$. It could be also written down as a matrix X_{tm} with t rows and m columns. The generalized parity check functions are the followng functions: $p_1(x) = (v_1, v_2, \ldots, v_t) \in \mathbf{F}^t, v_i = \sum_{j=1}^{m} x_{ij}$, $p_2(x) = (w_1, w_2, \ldots, w_m) \in \mathbf{F}^m, w_i = \sum_{i=1}^{t} x_{ij}$. Let A^t and B^m be two arbitrary binary codes of length t and m respectively and the codes distance at most 3. Without lost of generality, we assume that both of these codes contain the all-zero vector. Let $f : A^t \to \mathbf{F}^m$ be any function. The following theorem states the classic Mollard code construction.

Theorem 1. *(Mollard M., [2]) A set $M^n = \{(x, a+p_1(x), b+p_2(x)+f(a)) | x \in \mathbf{F}^{tm}, a \in A^t, b \in B^m\}$ is a binary code of length $n = tm + t + m$ which minimal distance equals to* 3.

If A^t and B^m are two perfect binary codes of length $t = 2^{t_1} - 1$ and $m = 2^{m_1} - 1$ respectively, then M^n is also a perfect binary code. Taking $m = 1$ in the Mollard construction, one can obtain some Vasil'ev code. But there exist perfect Mollard codes which are not equivalent to the perfect Vasil'ev codes. The Mollard construction let us to obtain nonlinear codes if f is nonlinear function. The power of the detection kernel of M^n is given further.

Lemma 1. *If A^t and B^m are systematic perfect codes, the Mollard code $M^n = \{(x, a + p_1(x), b + p_2(x) + f(a)) | x \in \mathbf{F}^{tm}, a \in A^t, b \in B^m\}$ is systematic.*

Let A^t and B^m be arbitrary systematic perfect codes with parameters $(t = 2^{t_1} - 1, \frac{2^t}{t+1}, 3)$ and $(m = 2^{m_1} - 1, \frac{2^m}{m+1}, 3)$ respectively. Therefore, the code A^t has $t - t_1$ information bits and t_1 redundant bits, the code B^m has $m - m_1$ information bits and m_1 redundant bits. Without lost of generality, we assume that the first $t - t_1$ bits in any codeword from A^t and the first $m - m_1$ bits in any codeword from B^m are the information ones.

If $P_1 : \mathbf{F}^{tm} \to \mathbf{F}^t$ and $P_2 : \mathbf{F}^{tm} \to \mathbf{F}^m$ are such mappings that the code distance of $(x, P_1(x), P_2(x))$ equals to 2, the following theorem is true.

Theorem 2. *The Mollard code $M^{tm+t+m} = \{(x, a + P_1 x, b + P_2 x + f(a)) | x \in \mathbf{F}^{tm}, a \in A^t, b \in B^m\}$ with parameters $(tm + t + m, \frac{2^{tm+t+m}}{tm+t+m+1}, 3)$ is a partially robust code with $|Ker_d(M^{tm+t+m})| = \frac{2^{tm+m}}{m+1}$ and $P_f = Q_{mc}(M^{tm+t+m})$.*

The classic Mollard construction can be generalized to build partially robust codes with any given code length n. Let $f : A^t \to \mathbf{F}^m$ be an arbitrary nonlinear function such that $f(0) = 0$.

Theorem 3. *The code $\tilde{M}^n = \{(x, a + p_1(x, 0), b + p_2(x, 0) + f(a)) | x \in \mathbf{F}^z, 0 \in \mathbf{F}^{tm-z}, 0 < z \leq tm, a \in A^t, b \in B^m\}$ is a partially robust code with parameters $(n = z + t + m, \frac{2^{z+t+m}}{tm+t+m+1}, 3)$, where $|Ker_d(\tilde{M}^n)| = \frac{2^{z+m}}{m+1}$, and the error masking probability $Q_{mc}(\tilde{M}^n) = P_f$. Adding one linear parity check bit to $\tilde{M}^n$, we get a* *partially robust code $\bar{M}^n$ with the code distance 4, and power of detection kernel and $max_{(e \notin Ker_d(D))} Q_D(e)$ like that of the code $\tilde{M}^n$.*

3 Memory protection architecture of the extended generalized Mollard code

Let H_A and H_B be the check matrixes of A^t and B^m respectively. If $c = (c_1 = x, c_2 = a + p_1(x,0), c_3 = b + p_2(x,0) + f(y_a), c_4 = p(x) + p(a) + p(b) + p(f(y_a)))$ is the codeword from $\bar{M}^n$, $\tilde{c} = (\tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \tilde{c}_4)$ is the gotten distorted vector, then $c = \tilde{c} + e$, where $e = (e_1, e_2, e_3, e_4)$ is the error vector.

Let y_A and y_B be the information bits of A^t and B^m, r_A and r_B be the redundant bits of A^t and B^m, and $\tilde{y_A}$ and $\tilde{y_B}$ be the distorted information bits of A^t and B^m respectively. For the localization and correction of errors, let us define the signature $S = (S_1, S_2, S_3)$ for the vector $\tilde{c}$ as follows: $S_1 = H_A(p_1(\tilde{c_1}, 0) + \tilde{c_2})$, $S_2 = H_B(p_2(\tilde{c_1}, 0) + f(\tilde{y_A}) + \tilde{c_3})$, $S_3 = p(\tilde{c_1}) + p(p_1(\tilde{c_1}, 0)) + p(p_2(\tilde{c_1}, 0)) + p(\tilde{c_2}) + p(\tilde{c_3}) + p(\tilde{c_4})$.

The purpose of the attached algorithm is to correct single errors in the information part of the code, and to declare multiple errors and single errors in the redundant part of the code as well. Recall that the length of the information part of $\bar{M}^n$ equals to $z + t + m - \log_2(t+1) - \log_2(m+1)$.

The error correction algorithm

Let us compute the signature $S = (S_1, S_2, S_3)$ for the word $\tilde{c}$, where $S_1 \in \mathbf{F}^{\log_2(t+1)}$, $S_2 \in \mathbf{F}^{\log_2(m+1)}$, $S_3 \in \mathbf{F}^1$.

1. If $S = 0^{\log_2(t+1)+\log_2(m+1)+1}$, then errors are not detected. Otherwise, there exist one or more errors, which is/are detected.

2. If S_3 equals to 0, and at least one value from S_1 and S_2 does not equal to 0, then errors of even multiplicity are detected.

3. If $S_3 = 1$, $S_2 = 0^{\log_2(m+1)}$, $S_1 = 0^{\log_2(t+1)}$, then $e_4 = 1$. Therefore, a single bit error in the $z + t + m + 1$th bit of the code occurred. The error is detected, and there will be no attempt to correct it.

4. If $S_3 = 1$, $S_1 = 0^{\log_2(t+1)}$, $S_2 = h_{B^k}$, where h_{B^k} is the kth column of H_B, then a single bit error in the third part of the code, or multiple errors of odd multiplicities are detected.

a) if $k \leq m - \log_2(m+1)$, switch the $(z + t + k)$-th bit of c and recalculate S_2. If $S_2 = 0$, then the single error in the $(z + t + k)$-th bit of c is detected and successfully corrected. Else, multiple errors of odd multiplicities are detected.

b) if $k > m - \log_2(m+1)$, then the error occurred in the redundant bits of B^m. The error is detected, and there will be no attempt to correct it.

5. If $S_3 = 1$, $S_1 = h_{A^i}$, $S_2 = 0$, where h_{A^i} is the i-th column of H_A, then a single bit error in the second part of the code or multiple errors are detected.

a) if $i \leq t - \log_2(t+1)$, switch the $z+i$-th bit of c and recalculate S_1 and S_2. If $S_1 = 0^{\log_2(t+1)}$ and $S_2 = 0^{\log_2(m+1)}$, then the single error in the i-th bit of c_2 – i.e. in the $z+i$-th coordinate of c – is detected and successfully corrected. Else, multiple errors occurred in the second part of the code.

b) if $i > t - \log_2(t+1)$, then the error occurred in the redundant bits of A^t. The error is detected, and there will be no attempt to correct it.

6. If $S_3 = 1$, $S_2 = h_{B j'}$, $S_1 = h_{A i'}$, where $h_{A i'}$ and $h_{B j'}$ are the i'-th and the j'-th columns of H_A and H_B respectively, then the single error in one of $\{1, \ldots, z+t\}$-th bits, or multiple errors are detected.

a) if $1 \leq (i'-1)m + j' \leq t - \log_2(t+1) \leq z$ or $1 \leq (i'-1)m + j' \leq z \leq t - \log_2(t+1)$, let us switch the $i = (i'-1)m + j'$-th bit of c and recalculate S_1 and S_2. If $S_1 = 0^{\log_2(t+1)}$ and $S_2 = 0^{\log_2(m+1)}$, then the single error in the i-th coordinate of c is detected and successfully corrected.

Otherwise, let us switch the i'-th bit of c_2 and recalculate S_1 and S_2. If $S_1 = 0^{\log_2(t+1)}$ and $S_2 = 0^{\log_2(m+1)}$, then the single error in the $z+i$-th bit of c is detected and successfully corrected.

In the other cases, multiple errors occurred.

b) if $1 \leq z \leq (i'-1)m + j' \leq t - \log_2(t+1)$, we switch the i'-th coordinate of c_2 and recalculate S_1 and S_2. If $S_1 = 0^{\log_2(t+1)}$ and $S_2 = 0^{\log_2(m+1)}$, then the single error in the $z+i$-th coordinate of c is detected and successfully corrected. Otherwise, multiple errors occurred.

c) if $1 \leq t - \log_2(t+1) \leq (i'-1)m + j' \leq z$, let us switch the $i = (i'-1)m + j'$-th bit of c, and recalculate S_1 and S_2. If $S_1 = 0^{\log_2(t+1)}$ and $S_2 = 0^{\log_2(m+1)}$, then the single error in the i-th bit of c is detected and successfully corrected. Otherwise, the single error in the redundant bits of A^t or multiple errors occurred.

d) if $1 \leq t - \log_2(t+1) \leq z \leq (i'-1)m + j'$ or $1 \leq z \leq t - \log_2(t+1) \leq (i'-1)m + j'$, then the single error in the redundant bits of A^t or multiple errors occurred. They are detected, but there will be no attempt for correction.

In all the other cases, multiple errors of odd multiplicity ≥ 3 are detected.

Let k_A and k_B be the dimensions of the codes A^t and B^m correspondingly: $k_A = t - \log_2(t+1), k_B = m - \log_2(m+1)$. The next theorem is true.

Theorem 4. *Let $\bar{M}^n$ be the extended generalizeed Mollard code with parameters $(z+m+t+1, \frac{2^{z+m+t}}{tm+t+m+1}, 4)$. There are $2^z(\frac{2^t}{t+1} - 1)$ errors which are conditionally detectable and $|Ker_d| = \frac{2^{z+m}}{m+1}$ undetectable errors. If only errors occurred to the information part of the code are corrected, the number of errors which are conditionally miscorrected is $k_A k_B \cdot 2^{z+k_A}(2^{k_B} - 1)$, and the number of miscorrected errors is $k_A(2^{z+k_A+m} - 1) + k_B 2^{z+k_B} - z$. The conditionally detectable error masking probability and the conditionally miscorrected errors miscorrection probability are limited by nonlinearity P_f of function f.*

4 Conclusion

The generalized Mollard code, as well as the shortened Vasil'ev code, is able to correct all the single errors and to detect multiple errors. It is easy to see that the number of undetectable and miscorrected multiple errors for the generalized Mollard code is much smaller than for the Hamming code.

If $\bar{V}^n = \bar{V}^{a+s+2}$ and $\bar{M}^n = \bar{M}^{z+m+t+1}$ are the shortened Vasil'ev code of length n and the generalized Mollard code of length n respectively, $n = 2^{k_1}+n_1$, where $0 \leq n_1 < 2^{k_1}$, $k_2 = [\log_2(n_1)]$, then $s = t = 2^{k_1}-1$, $a = n_1-1$, $n_1 = z+m$, $n_1 = 2^{k_2} + n_2$, $0 \leq n_2 < 2^{k_2}$. Recall that $|Ker_d(\bar{V}^n)| = 2^{n_1-1}$, $|\bar{V}^n| = 2^{n-k_1-2}$, $|Ker_d(\bar{M}^n)| = \frac{2^{z+m}}{m+1}$, $|\bar{M}^n| = 2^{n-k_1-k_2-1}$. Hence, if $t = 2^{k_1} - 1$ and $m = 2^{k_2} - 1$, then the number of masked errors of $\bar{M}^n$ is $2^{k_2-1} = \frac{m+1}{2}$ times smaller than the number of masked errors of $\bar{V}^n$. At the same time, the power of $\hat{M}^n$ is also $2^{k_2-1} = \frac{m+1}{2}$ times smaller than the power of $\bar{V}^n$.

If the correlation $(t + 1)(m + 1) = 2^{k_1+1}$ is true for some code $\bar{M}^n$ (i.e. in a fixed length n the code $\bar{M}^n$ has the highest power), and $t = 2^{[\log_2(n-1)]}$ (i.e. $|Ker_d(\bar{M}^n)| \geq |Ker_d(\bar{V}^n)|$), then $m = 1$ and this code $\bar{M}^n$ is a Vasil'ev code.

Because the decreasing of t (in a fixed code length n) causes the increasing of $z + m$, the power of $Ker_d(\bar{M}^n)$ is greater than the power of $Ker_d(\bar{V}^n)$ while $t < 2^{[\log_2(n-1)]}$.

It is easy to see that the number of miscorrected errors of $\bar{M}^n$ is less than the number of miscorrected errors of $\bar{V}^n$ while $t < 2^{k_1}-1$ and $a > s-\log_2(s+1)$. But the power of $\bar{M}^n$ is less than or equal to the power of $\bar{V}^n$.

To some up, for some parameters, the Mollard codes have less undetectable and miscorrected errors (and power) than the Vasil'ev codes. Therefore, if multiple bit corruption or repeating errors exist, this construction can provide better protection. The class of different Mollard codes is greater than the class of different Vasil'ev codes. Therefore, there are some advantages to using the extended generalized Mollard codes.

References

[1] Wang Z., Karpovsky M., Kulikowski K.J, Replacing linear hamming codes by robust nonlinear codes results in a reliability improvement of memories, *Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks*, 2009, 514 - 523.

[2] Mollard M., A generalized parity function and its use in the construction of perfect codes, *SIAM J. Alg. Disc. Meth.* V.7. N.1, 1986, 113 – 115.

[3] Vasil'ev Yu.L., On nongroup close-packed codes, *Problems of Cybernetics* V.8, 1963, 337 – 339 (in Russian).