PAPER Special Section on Information Theory and Its Applications

Codes over the Ring of Integers Modulo m*

A.J. Han VINCK†, *Nonmember and* Hiroyoshi MORITA††, *Member*

SUMMARY We discuss the concept of coding over the ring of integers modulo m. This method of coding finds its origin in the early work by Varshamov and Tenengolz [1]. We first give a definition of the codes followed by some general properties. We derive specific code constructions and show computer-search results. We conclude with applications in 8-phase modulation and peak-shift correction in magnetic recording systems [2].

key words: *error control, constrained sequences, number theory, frame synchronization, coded modulation*

1. Introduction

In coding literature, one mainly deals with binary oriented codes. However, in a number of applications, it is natural to represent symbols by integers. We mention the areas of coded modulation and magnetic recording. Motived by the early work of Varshamov and Tenengolz [1] and the results from Levenshtein and Vinck [2], we develop the concept of integer codes, IC, for channels with input alphabet $\{0, 1, \ldots, m-1\}$ and output alphabet $\mathcal{Z}_m$ where $\mathcal{Z}_m$ is the ring of integers modulo m. In coded modulation, as for instance in 8-phase modulation, one can represent the symbols in the signal constellation by the integers $0, 1, \ldots, 7$. In magnetic recording, symbols are phrases of runlength limited sequences. For a $(d, k) = (1, 3)$ sequence, phrases have lengths 2, 3, 4, respectively. It is thus natural to use integers for the symbol representation.

Definition 1: Let $m, M, N \in \mathcal{N}$, $H \in \mathcal{Z}_m^{M \times N}$ and $\boldsymbol{d} \in \mathcal{Z}_m^M$. Then the IC is defined by

$$\{\boldsymbol{a} \in \mathcal{Z}_m^N : \boldsymbol{a}H^T = \boldsymbol{d}\}, \tag{1}$$

where H can be seen as the check matrix for the IC.

Without loss of generality, we may assume that $\boldsymbol{d} = \boldsymbol{0}$. It is easy to show that any code with $\boldsymbol{d} \neq \boldsymbol{0}$ can be transformed into a code with $\boldsymbol{d} = \boldsymbol{0}$, by subtracting one codeword from all others. The code with $\boldsymbol{d} = \boldsymbol{0}$ thus has the maximum number of codewords. Furthermore, we are also able to transform codewords with $\boldsymbol{d} = \boldsymbol{0}$ to codewords with $\boldsymbol{d} \neq \boldsymbol{0}$. Hence, all codes have the same number of codewords for the specific H. The following theorem is important for the enumeration of the number of different codes.

Theorem 1: If the greatest common divisor (gcd) of the $M \times M$ subdeterminants of H is equal to a unit in $\mathcal{Z}_m$, then H defines m^M different codes of equal size m^{N-M}.

Proof: The proof follows from the fact that, under the given condition, for any vector $\boldsymbol{d} \in \mathcal{Z}_m^M$ we can always construct a vector $\boldsymbol{a} \in \mathcal{Z}_m^N$ such that $\boldsymbol{a}H^T = \boldsymbol{d}$ occurs. □

Let $N = 1$. If the gcd of the $M \times M$ subdeterminants of H is equal to b is not a unit in $\mathcal{Z}_m$, then we have m/b different codes each of size $b\,m^{N-1}$. Here, the division m/b is equal to the smallest integer $c < m$ such that $bc = 0 \pmod m$. For nontrivial codes all components of H must be different. Therefore, $N \leq m/b$.

Example: Let $m = 10$, $M = 1$, and gcd $= 6$ ($5 \times 6 = 30 = 0 \pmod{10}$). The units are 3 ($3 \times 7 = 21 = 1 \pmod{10}$), 7 and 9 ($9 \times 9 = 81 = 1 \pmod{10}$). We have 5 codes with $d = 6, 2, 8, 4$ and 0, respectively.

In the rest of the paper we only consider matrices H according to Theorem 1. In our search for good codes, we can eliminate a number of candidate matrices H by using the following definition.

Definition 2: Let C_1 and C_2 be IC codes with check matrices H and G of the same dimensions and over the same ring $\mathcal{Z}_m$. We say that C_1 and C_2 are equivalent if

- the rows of G are given by a permutation of the rows of H and are possibly multiplied with a unit of $\mathcal{Z}_m$,
- the columns of G are given by a permutation of the columns of H and are possibly multiplied with -1.

Example: For $N = 4$, $m = 394$, $H = (1, 3, 9, 16)$ is equivalent to $G = (-1, -3, 12, 17)$ because G can be obtained from H by multiplying H with 17.

Channel errors are additive and have integer values in the set $e \in \{-t, -t+1, \ldots, t-1, t\}$. When we transmit a symbol c, we assume that we receive $r = c + e \pmod m$ and the error is $e = r - c \pmod m$. We furthermore assume that $t < m/2$. For the error vector $\boldsymbol{e} = (e_1, e_2, \ldots, e_n)$, we denote weight $(\boldsymbol{e})$ as the number of non-zero components.

Manuscript received January 16, 1998.
Manuscript revised April 1, 1998.

†The author is with the Institute for Experimental Mathematics, University of Essen, Ellernstr. 29, 45326 Essen, Germany.

††The author is with the Graduate School of Information Systems, University of Electro-Communications, 1–5–1 Chofugaoka, Chofu-shi, 182–8585 Japan.

*Part of the material was presented at SITA '97, Matsuyama, Japan.

Definition 3: Let C be an IC with check matrix H and $\boldsymbol{d} = \boldsymbol{0}$. Then the syndrome $\boldsymbol{S}$ of a received word $\boldsymbol{r} \in \mathcal{Z}_m^N$ is defined by

$$\boldsymbol{S} = \boldsymbol{r}H^T = \boldsymbol{e}H^T. \tag{2}$$

Definition 4: An IC is called s-error correcting with size of error t if all errors $\boldsymbol{e} = (e_1, e_2, \ldots, e_N)$ with weight $(\boldsymbol{e}) \leqq s$ and $e_i \in \{-t, -t+1, \ldots, t-1, t\}$ for all i, can be corrected.

To be able to correct these errors, all syndromes have to be different. Therefore,

$$m^M \geqq \sum_{i=0}^{s} \binom{N}{i} (2t)^i. \tag{3}$$

We call an IC perfect when (3) is an equality. It is easy to check that two equivalent codes have the same s and t.

Example: For $s = 1$, $t = 1$, $M = 2$, and $N = 4$, we can make a perfect code over $\mathcal{Z}_3$ with

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix}.$$

Example: For $s = 1$, $t = 2$, $M = 2$, and $N = 6$, we can make a perfect code over $\mathcal{Z}_5$ with

$$H = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

In a practical situation, we have to limit the set of possible input (code) symbols. The channel output space is defined by the modulo m operation. Suppose that we restrict the code symbols for an IC to the alphabet $\{0, 1, \ldots, m'-1\}$, where $m' < m$. At the receiver we still perform the modulo m operations on the received symbols. From (1) it follows that there is a maximum of m^M possible different codes, one for every $\boldsymbol{d}$. Now, not all codes have the same cardinality, since there is no guaranteed one-to-one mapping between the different codes. Since there are $(m')^N$ different input sequences, there must exist a code with at least $(m')^N/m^M$ codewords. The usual definition for code efficiency or rate R gives

$$R \geq \frac{N-M}{N} + \frac{M}{N} \log_{m'} \left(\frac{m'}{m} \right). \tag{4}$$

Note that when $m' = m$, the code rate $R = (N-M)/M$. From (4) it follows that we have to choose m as close to m' as possible, in order to maximize the expected code rate.

Remark: If the channel input and output are numbers modulo m', we have to choose $m = m'$. This situation occurs for instance in coded 8-phase modulation, where the input and output alphabet is $\{0, 1, \ldots, 7\}$. A channel error causing the detection of a 7 instead of a transmitted symbol 0 is an error $e = -1 \pmod 8$. Taking $m = 9$ would result in an error $e = -2 \pmod 9$.

In the next section we give some coding constructions for $s = 1$ and $s = 2$, followed by results from a computer search. The last section deals with some applications.

2. Code Constructions

In this section we give some code constructions. We first recall some of the results from [2], where the main focus was on the construction of perfect codes for $s = 1$.

Theorem 2: The vector of integers $H = (1, 2, \ldots, N)$ is the check matrix for a perfect IC with $s = 1$ and $t = 1$ over $\mathcal{Z}_m$ for $m = 2N + 1$.

Proof: The proof follows immediately from considering all possible values $\pm h_i$. □

Theorem 3: A perfect IC with $s = 1$, $t = 2$ and $m = 4N + 1$ exists iff

$$\gcd\left(2^{4\ell+2} - 1, 4N + 1\right) = 1, \quad \text{for any } \ell = 0, 1, \ldots, N-1.$$

Proof: See [2]. □

Theorem 4: The vector of integers $H = (h_i = 2i - 1, i = 1, \ldots, N)$ is the check matrix for an IC with $s = 1$ and $t = 2$ over Z_m for $m = 4N + 2$.

Proof: The proof follows immediately from considering all possible values $\pm h_i$ and $\pm(2hi)$. □

We now give a general construction for the specific case where $t = (p-1)/2$, and p is a prime. It is known that in the M-dimensional vector space $\mathcal{F}_p^M$ there exist a one-dimensional subspaces and $(p^M - 1)/(p-1)$ projective vectors the first nonzero coordinates of which are equal to 1 which generate these one-dimensional subspaces. We denote the set of these vectors by H_p^M. We then have:

Theorem 5: Let $t = (p-1)/2$, where p is a prime and $N = (p^M - 1)/(p-1)$, then H_p^M forms the check matrix for a perfect IC with parameters $m = p, M, N$.

Proof: See [2]. □

Note: In [2] it is indicated that for the conditions of Theorem 4, a perfect IC exists for $m = p^M$ as well.

Example: Let $t = 3$, $p = 7$, $M = 2$ and $N = 8$. The check matrix

$$H_7^2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 1 \end{pmatrix}.$$

The corresponding vector $H = (1, 8, 15, 22, 29, 36, 43, 7)$ forms the check matrix for a perfect IC for $m = 49$.

Ulrich Tamm [3], gave a necessary condition for the existence of a perfect code for $s = 1$, $t = 3$ and $t = 4$. He also gives some examples for $t = 3$ codes using $\mathcal{Z}_{6N+1}$.

For $s > 1$, the situation is much more difficult than for $s = 1$. Suppose that $H = (h_1, h_2, \ldots, h_N)$, $s = 2$, and $t = 1$. For the possible $2N^2 + 1$ syndrome values to be different, $m^M \geqq 2N^2 + 1$.

Example: Take $M = 1$, $N = 3$, $s = 2$ and $t = 1$. For $H = (2, 5, 6)$ the following syndrome values $\pm\{2, 5, 6, 1, 4, 3, 7, 8, 11\}$ are different modulo 20. It can be shown that no perfect code for $m = 19$ exists.

Example: Take $M = 1$, $N = 3$, $s = 2$ and $t = 3$. The smallest m for which H matrix exists with these parameters is 200 and $H = (4, 25, 28)$. The lower bound for $m = 127$ follows from (3).

Theorem 6: [4] For the check matrix $H = (h_1, h_2, \ldots, h_N)$ of a perfect IC code with $M = 1$ and $s = 2$, the following must hold:

1) $^{\forall}i\, ^{\exists}j \neq i\, ^{\exists}A \in \{-t, \ldots, t\}\ (Ah_j = (2t+1)h_i)$.

2) $^{\forall}i\, ^{\forall}j \neq i\, ^{\forall}A \in \{-t, \ldots, t\}\ (h_j \neq Ah_i)$.

Proof: If the condition 1) in Theorem 6 is not fulfilled, then the syndromes $\pm(t+1)h_i$ do not occur. Condition 2) is obvious. □

Example: Using the conditions from Theorem 6, one can verify that for $M = 1$, $N = 4$, $s = 2$, and $t = 1$, $m > 33$. For these parameters, the first value of m for which an IC exists is $m = 39$.

Theorem 7: The code defined by $H = (1, 2t+1, \ldots, (2t+1)^{N-1})$ over $\mathcal{Z}_m$, where $m = (2t+1)^N$, is N error correcting with error size t.

Proof: Follows from comparing two error vectors of correctable weight. □

The codes defined in Theorem 7 are perfect.

Example: The code specified by $H = (1, 7)$ is perfect for $s = 2$, $t = 3$, $N = 2$, $M = 1$, and $m = 49$.

Example: The code specified by $H = (1, 3, 9)$ corrects a maximum of $s = 3$ errors for $t = 1$ and $m = 27$. The 26 nonzero different syndromes are $\pm(1, 3, 9, 2, 6, 4, 10, 12, 8, 13, 5, 7, 11)$.

We now describe a property for the components of H that can be used in developing search programs for good codes. We define A_d as the number of h_i with $d|h_i$ and $d|m$.

Theorem 8:

$$\sum_{i=0}^{s} \binom{A_d}{i} (2t)^i \leq \frac{m}{d}. \tag{5}$$

Proof: The number of possible different syndromes divisible by d is m/d. In (5) we count only those syndrome values that are connected with error events. □

Example: Let $m = 35$, $N = 4$, $s = 2$, $t = 1$. Then, for

- $d = 5 \rightarrow 2A_5^2 + 1 \leq 7$, and thus $A_5 = 0$ or $A_5 = 1$.
- $d = 7 \rightarrow 2A_7^2 + 1 \leq 5$, and thus $A_7 = 0$ or $A_7 = 1$.

3. Computer Search Results

In this section we show some computer search results. We give the *optimum* codes, specified by H, i.e. smallest m for a specific N and the indicated parameters.

Example: The optimum code for $s = 2$, $t = 4$, $N = 3$, $M = 1$, and $m = 403$ is specified by

$$H = (1, 38, 196).$$

Table 1 Search Results for $s = 2$, $t = 1$, and $M = 1$.

N	H	m	$2N^2 + 1$
2	(1, 3)	9	9
3	(2, 5, 6)	20	19
4	(1, 3, 9, 14)	39	33
5	(1, 3, 11, 20, 27)	60	51
6	(1, 3, 8, 18, 31, 45)	96	73
7	(1, 3, 8, 21, 31, 46, 58)	130	99
8	(1, 9, 13, 16, 40, 6, 51, 74)	168	129

Table 2 Search Results for $s = 2$, $t = 2$, and $M = 1$.

N	H	m	$8N^2 - 4N + 1$
2	(1, 5)	25	25
3	(3, 13, 15)	78	61
4	(1, 8, 38, 64)	171	113
5	(1, 6, 55, 101, 127)	287	181

Table 3 Search Results for $s = 2$, $t = 1$, and $M = 2$.

N	H	m	$\sqrt{2N^2+1}$
3	$\begin{pmatrix} 4 & 3 & 4 \\ 4 & 5 & 5 \end{pmatrix}$	6	> 4
4	$\begin{pmatrix} 6 & 1 & 2 & 6 \\ 5 & 6 & 6 & 6 \end{pmatrix}$	7	> 5
5	$\begin{pmatrix} 5 & 7 & 8 & 1 & 5 \\ 7 & 7 & 7 & 8 & 8 \end{pmatrix}$	9	> 7

Example: The optimum code for $s = 2$, $t = 2$, $N = 3$, $M = 2$, and $m = 13$ is specified by

$$H = \begin{pmatrix} 5 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

The lower bound for m is 8.

Example: The optimum code for $s = 3$, $t = 1$, $M = 2$, $N = 4$, and $m = 9$ is specified by

$$H = \begin{pmatrix} 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

The lower bound for m is 8.

4. Applications

4.1 Peak-Shift Correcting Codes

The first application can be found in [2], where Levenshtein and Vinck define Peak-Shift correcting codes for magnetic recording systems. In high-density magnetic recording systems, runlength-limited (RLL) sequences are used to increase density and control self clocking [5]. The read-out mechanism detects changes in magnetization and thus from the RLL sequence we can derive a so called (d, k)-sequence, where $d+1$ and $k+1$ correspond to the minimum and maximum length of the RLL substrings, respectively. A (d, k)-sequence is represented by consecutive zero-symbol runs of length $i, d \leq i \leq k$, between pairs of one symbols. Read-out

circuitry imperfection and clock jittering may cause misdetection of magnetization transitions and is supposed to result in peak-shifts left or right in the (d, k)-sequence.

A peak-shift error correcting code is defined as

$$C = \left\{ \boldsymbol{\alpha} : \sum_{i=1}^{N} \alpha_i \omega_i = 0, \text{ where } \alpha_i \in \mathcal{Z}_{m'} \text{ and } \omega_i \in \mathcal{Z}_m \right\}. \quad (6)$$

A peak-shift increases α_i and decreases α_{i+1} with a value $\leqq t$, or vice versa. A peak-shift at position i of size j gives as a syndrome value

$$S = \begin{cases} \pm j(\omega_i - \omega_{i+1}) & \text{if } i = 1, 2, \ldots, N-1, \\ \pm j\omega_N & \text{if } i = N. \end{cases} \quad (7)$$

Now, let

$$\omega_i = \sum_{j=i}^{N} h_j, \qquad i = 1, 2, \ldots, N,$$

where h_i is a component from the check matrix H for an IC with $s = 1$, $M = 1$, maximum error size t. Note that

$$h_i = \omega_i - \omega_{i+1} \quad \text{for} \quad i = 1, \ldots, N-1,$$
$$h_N = \omega_N.$$

We thus constructed a single peak-shift error correcting code, since all syndromes in (7) are different.

4.2 Coded Modulation

In 8-phase modulation, we may number the transmitted symbols from 0 to 7, see Fig. 1. In an IC of length N, we transmit a series of N signals, where each signal corresponds to an integer from the signal constellation. At the receiver, we first map the received analogue signals back to integers and do the decoding operation. Since the transmitted and received symbols are from the set of integers modulo 8, we also have to do the decoding over the set of integers modulo 8. This reduces the set of possible codeword lengths. For instance, for $m = 8$, $t = 3$, $s = 1$, we need $M = 2$ and $N < 11$. No code for $M = 1$ exists with these parameters.

For the standard 8-phase modulation, one uses a mapping from 4 symbols uncoded to 8 symbols coded, or a rate 2/3 encoding. As an alternative, we give an example of an $R = 2/3$ code correcting a single error of maximum size 3 below. The code is specified by the check matrix

$$H = \begin{pmatrix} 0 & 1 & 1 & 3 & 2 & 3 \\ 1 & 0 & 1 & 1 & 3 & 2 \end{pmatrix}.$$

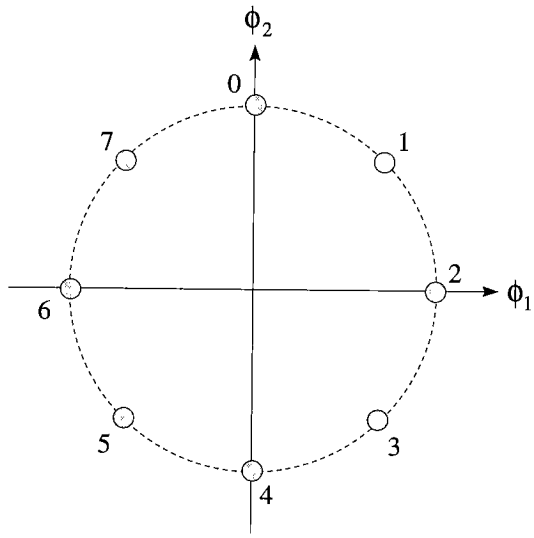


Fig. 1 Constellation for 8-phase modulation.

Table 4 Syndrome Former Outputs.

error position	error value 1	2	3	−1	−2	−3
1	10	20	30	70	60	50
2	01	02	03	07	06	05
3	11	22	33	77	66	55
4	13	26	31	75	62	57
5	32	64	16	56	24	72
6	23	46	61	65	42	27

For a codeword $\boldsymbol{c} = (c_1, c_2, c_3, c_4, c_5, c_6)$, where $c_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$, the syndrome former outputs $\boldsymbol{S} = \boldsymbol{c}H^T = (0, 0) \pmod 8$. Furthermore, note that H defines a systematic encoding procedure. A codeword consists of 4 information symbols and two symbols in the first positions that can be used to satisfy the equations as given by H. The minimum squared Euclidean distance [6] is $6 - 3\sqrt{2}$ which gives an equivalent asymptotic coding gain of 2.44 dB over the additive white Gaussian noise channel.

If we use an error vector $\boldsymbol{e} = (e_1, e_2, e_3, e_4, e_5, e_6)$, where $e_i \in \{0, 1, 2, 3, -1, -2, -3\}$, the syndrome former outputs $\boldsymbol{S} = (\boldsymbol{c} + \boldsymbol{e})H^T = \boldsymbol{e}H^T$ are different for all single error patterns of maximum size 3, and are thus correctable. For completeness, we include all possible syndrome former outputs for the given H, in Table 4.

Another code with $s = 1$, $t = 1$, is defined by $H = (1, 2, 3)$. For this systematic code the rate $R = 2/3$, and the minimum squared Euclidean distance is $4 - \sqrt{2}$, which gives an equivalent asymptotic coding gain of 4.13 dB. Another example is the code of length $N = 3$, specified by

$$H = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \end{pmatrix}.$$

The $R = 1/3$ systematic code has parameters $s = 1$, $t = 3$; $s = 3$, $t = 1$. Hence, it corrects up to 3 errors of

maximum size 1 and 1 error of maximum size 3. The minimum squared Euclidean distance is 6. The equivalent asymptotic coding gain is 4.7 dB. A future research topic is the development of good codes and estimation of the performance in a communication environment.

4.3 Synchronization Error Correcting Codes

Another original application of integer codes is the correction of synchronization (sync) errors caused by inserting a new symbol into a codeword or deleting a symbol from a codeword.

In the literature there are many works on correcting sync errors [7]–[10]. Levenshtein [7] proposed binary codes capable of correcting s or fewer deletions of ones (zeros) in a word. In the same paper he also showed that the Varshamov and Tenengloz code [1] which was originally proposed for correcting asymmetric substitution errors of $1 \rightarrow 0$ (or $0 \rightarrow 1$) is capable of correcting single insertion or deletion in a codeword, where the boundary between two consecutive codewords is assumed to be detected correctly even if those codewords are affected by sync errors. Without this assumption, Calabi and Hartnett [9] gave binary block codes that are capable of correcting, in every $t \geq 3$ consecutive words, either one substitution error in each one of, at the most, $t-2$ words, or one synchronization error (but not both).

In the specific case we assume that frames are of variable length and that beginning and end of frames can be detected without error. This can be achieved by appropriate use of markers. Of course, it is important to consider the case that markers may be affected by sync errors. We discussed such a problem in [10], [11] as an extention of the work in [8].

We give a new example of a fixed-lengh binary code that corrects a single inversion, or an insertion/deletion error in codewords. The code obtained is defined as in (6) with parameters $N = n = 0 \pmod 3$, $m = n+1$, $m' = 2$, and $\omega_i = i$ $(i = 1, \ldots, n)$. Besides, we need an additional condition on the number of ones in codewords:

Theorem 9: The code C with binary codewords of fixed length $n = 0 \pmod 3$, for which

$$\sum_{i=1}^{n} i x_i = 0 \pmod{n+1} \text{ and } \sum_{i=1}^{n} x_i = 0 \pmod 3$$

corrects a single inversion error or a single insertion/deletion error.

Proof: The length condition can be used to detect an insertion or deletion error. If no length change occurs, but condition 1 is violated, we detect an inversion error. The second condition can be used to distinguish between a $0 \rightarrow 1$ or a $1 \rightarrow 0$ inversion. The first condition then gives the final solution for the position. For an insertion or deletion error, we can use the second condition to solve the value of the insertion or deletion. Again, we use the first condition to solve the position. The method is very similar to the codes described by Varshamov and Tenengolz [1]. □

Remark: In general, codes of the above type with cardinality

$$|C| \geq \frac{2^n}{3(n+1)}$$

can be shown to exist.

Example: Let $n = 6$. The following code, with the 4 codewords

$$000000 \quad 111111 \quad 001011 \quad 110100,$$

for which

$$\sum_{i=1}^{6} i x_i = 0 \pmod 7; \quad \sum_{i=1}^{6} x_i = 0 \pmod 3$$

corrects a single inversion error or a single deletion/insertion error. The lower bound for the number of codewords is 3.

The above codes can be made systematic in the following way.

- To satisfy the first condition, we use the values of x_i at positions $1, 2, 4, 8, \ldots, 2^{\lfloor \log n \rfloor}$.
- To satisfy the second condition, we use the positions h, i, j, k such that $h + i = j + k = n + 1$ and $h \neq i \neq j \neq k$. Note that

$$\begin{aligned} &x_h x_i x_j x_k = 0000 \\ &\quad \rightarrow \Sigma = 0 \text{ and } \#1's = 0 \pmod 3 \\ &x_h x_i x_j x_k = 0011 \\ &\quad \rightarrow \Sigma = n+1 \text{ and } \#1's = 2 \pmod 3 \\ &x_h x_i x_j x_k = 1111 \\ &\quad \rightarrow \Sigma = 2n+2 \text{ and } \#1's = 1 \pmod 3. \end{aligned}$$

In [12], Kabatjanskii, Vinck and van Wijngaarden give examples and extensions of the above codes to be used in systems where bit-stuffing errors occur. For synchronization purposes one often uses the marker $011\ldots1$. Bit-stuffing is used to avoid the pattern $011\ldots1$ in the data stream. It is known that in these systems already single inversion errors may cause undetectable error patterns for any CRC code with odd minimum distance. The IC do not have this disadvantage.

Comment: During the first INTAS meeting in Armenia, September 1996, Martirossian [13], discussed a class of codes for the amplitude and phase modulated channel. This class of codes matched the classes described in [2]. He further found an expression for computing the cardinalities of codes defined by congruencies. Similar results for $t = 3$ and 4 were obtained by Akihiro Munemasa [14]. He derived necessary and sufficient conditions for the existence of perfect codes in finite abelian groups.

5. Conclusion

We discuss the application of codes over the ring of integers modulo m. We give bounds on the code size, code properties, code constructions, computer results and applications. It can be concluded that the concept of IC is interesting in the area where messages are represented by integers as in coded modulation and magnetic recording.

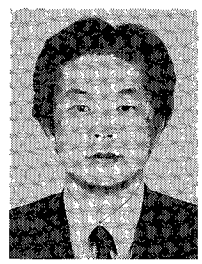
Acknowledgement

The authors wish to thank Rob Stroeks for introducing some general results.

References

[1] R.R. Varshamov and G.M. Tenengolz, "One asymmetrical error-correcting codes," (in Russian) Avtomatika i Telemehanika, vol.26, no.2, pp.288–292, 1965.
[2] V.I. Levenshtein and A.J. Han Vinck, "Perfect (d, k)-codes capable of correcting single peak-shifts," IEEE Trans. Inf. theory, vol.39, no.2, pp.656–662, 1993.
[3] U. Tamm, "On perfect 3-shift N-designs," IEEE ISIT97, Ulm, p.454, June 29–July 4 1997.
[4] R. Stroeks, "Number theoretical codes," Internal Report, University of Essen, June 1997.
[5] K.A.S. Immink, "Coding techniques for digital recording," Prentice Hall, Englewood Cliffs, NJ, 1990.
[6] R.E. Blahut, "Digital transmission of information," Addison-Wesley, New York, 1990.
[7] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Sov. Phys. Doklady, vol.10, pp.707–710, 1966 (translated from Doklady Akaddemii Nauk SSSR, vol.163, pp.845–848, 1965).
[8] J.D. Ullman, "Near-optimal, single-synchronization-error-correcting code," IEEE Trans. Inf. Theory, vol.12, no.4, pp.418–424, 1966.
[9] L. Calabi and W.E. Hartnett, "A family of codes for the correction of substitution and synchronization errors," IEEE Trans. Inf. Theory, vol.5, pp.102–106, 1969.
[10] H. Morita, A.J. van Wijngaarden, and A.J. Han Vinck, "Prefix synchronized codes capable of correcting single insertion/deletion errors," Proc. 1997 Int. Symp. on Inform. Theory, p.409, Ulm, Germany, July 1997.
[11] H. Morita and Y. Daikoku, "On separable decodability of double synchronization error correcting codes," Proc. 20th Symp. on Inform. Theory and its Applications, pp.269–272, Matsuyama, Japan, Dec. 1997.
[12] G. Kabatjanskii, A.J. Han Vinck, and A.J. van Wijngaarden, "On combined synchronization and error control coding," IEEE ISIT94, Trondheim, p.62, June 1994.
[13] S. Martirossian, "Single error-correcting, close packed and perfect codes," Proc. First INTAS International Seminar on Coding Theory and Combinatorics, Thahkadzor, Armenia, ISBN 90-74249-14-0, Oct. 6–11 1996.
[14] A. Munemasa, "On perfect t-shift codes in abelian groups," Department of Mathematics, Kyushu University, Fukuoka, Japan. Manuscript 1997.

A.J. Han Vinck was born in Breda, The Netherland, in 1949. He received the M.sc. degree in 1974 and the Ph.D. degree in 1980, both from the Technical University of Eindhoven, Eindhoven, The Netherlands. From 1985 to 1990, he was an Associate Professor at the University of Eindhoven, and became a Full Professor in 1990 at the University of Essen, Essen, Germany. From 1992 to 1994, he was Director of the Institute for Experimental Mathematics at the University of Essen. His interest is digital communications. He organized the Veldhoven Workshop on Information Theory in 1990 and was Cochairman of the International Symposium on Information Theory (ISIT'97) in Ulm. Germany.



Hiroyoshi Morita received the B.Eng. degree, the M.Eng. degree, and the D.Eng. degree from Osaka University, in 1978, 1980, and 1983, respectively. In 1983, he joined Toyohashi University of Technology, Aichi, Japan as a Research Associate in the School of Production Systems Engineering. In 1990, he joined University of Electro-Communications, Tokyo, Japan, first as an Assistant Professor at the Department of Computer Science and Information Mathematics, where from 1992, he was an Associate Professor. Since 1995 he has been with the Graduate School of Information Systems. He was Visiting Fellow at the Institute of Experimental Mathematics, University of Essen, Essen, Germany, during 1993–1994. His research interests are in combinatorial theory, information theory, and coding theory, with applications to digital communication systems.