Systematic Error-Correcting Codes for Rank Modulation

Hongchao Zhou
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125
Email: hzhou@caltech.edu

Anxiao (Andrew) Jiang
Computer Science and Eng. Dept.
Texas A&M University
College Station, TX 77843
ajiang@cse.tamu.edu

Jehoshua Bruck
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125
Email: bruck@caltech.edu

***Abstract*—The rank modulation scheme has been proposed recently for efficiently writing and storing data in nonvolatile memories. Error-correcting codes are very important for rank modulation, and they have attracted interest among researchers. In this work, we explore a new approach, *systematic error-correcting codes for rank modulation*. In an (n, k) systematic code, we use the permutation induced by the levels of n cells to store data, and the permutation induced by the first k cells $(k < n)$ has a one-to-one mapping to information bits. Systematic codes have the benefits of enabling efficient information retrieval and potentially supporting more efficient encoding and decoding procedures. We study systematic codes for rank modulation equipped with the Kendall's τ-distance. We present $(k+2, k)$ systematic codes for correcting one error, which have optimal sizes unless *perfect* codes exist. We also study the design of multi-error-correcting codes, and prove that for any $2 \leq k < n$, there always exists an (n, k) systematic code of minimum distance $n-k$. Furthermore, we prove that for rank modulation, systematic codes achieve the same capacity as general error-correcting codes.**

I. Introduction

The rank modulation scheme has been proposed recently for efficiently and robustly writing and storing data in nonvolatile memories (NVMs) [7], [8]. Its applications include flash memories [3], which are currently the most widely used family of NVMs, and several emerging NVM technologies, such as phase-change memories [2]. The rank modulation scheme uses the relative order of cell levels to represent data, where a cell level denotes a floating-gate cell's threshold voltage for flash memories and denotes a cell's electrical resistance for resistive memories (such as phase-change memories). Consider n memory cells, where for $i = 1, 2, \cdots, n$, let $c_i \in \mathbb{R}$ denote the level of the ith cell. It is assumed that no two cells have the same level, which is easy to realize in practice. Let $\mathcal{S}_n$ denote the set of all $n!$ permutations of $\{1, 2, \cdots, n\}$. The n cell levels induce a permutation $[x_1, x_2, \cdots, x_n] \in \mathcal{S}_n$, where $c_{x_1} > c_{x_2} > \cdots > c_{x_n}$. The rank modulation scheme uses such permutations to represent data. It enables memory cells to be programmed efficiently and robustly from lower levels to higher levels, without the risk of over-programming. It also makes it easier to adjust cell levels when noise appears without erasing/resetting cells, and makes the stored data be more robust to asymmetric errors that change cell levels in the same direction [7], [8].

Error-correcting codes for rank modulation are very important for data reliability [3], [9]. Errors are caused by noise in cell levels, and the smallest error that can happen is for two adjacent cell levels to switch their order in the permutation, which is called an *adjacent transposition* [5]. An adjacent transposition changes a permutation $[x_1, x_2, \cdots, x_n] \in \mathcal{S}_n$ to $[x_1, \cdots, x_{i-1}, x_{i+1}, x_i, x_{i+2}, \cdots, x_n]$ for some $i \in \{1, 2, \cdots, n-1\}$. In this paper, as in [1], [8], [9], we measure the distance between two permutations $\mathbf{x} = [x_1, x_2, \cdots, x_n] \in \mathcal{S}_n$ and $\mathbf{y} = [y_1, y_2, \cdots, y_n] \in \mathcal{S}_n$ by the minimum number of adjacent transpositions needed to change $\mathbf{x}$ into $\mathbf{y}$ (and vice versa), and denote it by $d_\tau(\mathbf{x}, \mathbf{y})$. This distance metric is called the Kendall's τ-distance [5]. For example, if $\mathbf{x} = [2, 1, 3, 4]$ and $\mathbf{y} = [3, 1, 4, 2]$, then $d_\tau(\mathbf{x}, \mathbf{y}) = 4$, because to change the permutation from $\mathbf{x}$ to $\mathbf{y}$ (or vice versa), we need at least 4 adjacent transpositions: $[2, 1, 3, 4] \to [1, 2, 3, 4] \to [1, 3, 2, 4] \to [1, 3, 4, 2] \to [3, 1, 4, 2]$. Based on this distance metric, an error-correcting code that can correct t errors is a subset of $\mathcal{S}_n$ whose minimum distance is at least $2t + 1$.

There have been some results on error-correcting codes for rank modulation equipped with the Kendall's τ-distance. In [9], a one-error-correcting code is constructed based on metric embedding, whose size is provably within half of the optimal size. In [1], the capacity of rank modulation codes is derived for the full range of minimum distance between codewords, and the existence of codes whose sizes are within a constant factor of the sphere-packing bound for any fixed number of errors is shown. Some explicit constructions of error-correcting codes have been proposed and analyzed in [11] and [12]. There has also been some work on error-correcting codes for rank modulation equipped with the L_∞ distance [13], [14]. The distance metric is more appropriate for cells where the noise in cell levels has limited magnitudes.

In this paper, we study *systematic* error-correcting codes for rank modulation as a new approach for code design. Let k and n be two integers such that $2 \leq k < n$. In an (n, k) systematic code, we use the permutation induced by the levels of n cells to store data. The first k cells are called *information cells*, whose induced permutation has a one-to-one mapping to information bits. The last $n - k$ cells are called *redundant cells*, which are used to add redundancy to the codewords. Compared to the existing constructions of error-correcting

978-1-4673-2579-0/12/$31.00 ©2012 IEEE

codes for rank modulation, systematic codes have the benefit that they support efficient data retrieval, because when there is no error (or when error correction is not considered), data can be retrieved by only reading the information cells. And since every permutation induced by the information cells represents a unique value of the data, the permutations can be mapped to data (and vice versa) very efficiently via enumerative source coding (e.g., by ordering permutations alphabetically and map them to data) [4], [10]. In addition, the encoding algorithm of the error-correcting code can potentially be made very efficient by defining the positions of the redundant cells in the permutation as a function of the corresponding positions of the information cells.

We study the design of systematic codes, and analyze their performance. We present a family of $(k+2,k)$ systematic codes for correcting one error, where either k or $k+1$ is a prime number. We show that they have optimal sizes among systematic codes, unless *perfect* systematic one-error-correcting codes, which meet the sphere-packing bound, exist. We also study the design of systematic codes that correct multiple errors, and prove that for any $2 \leq k < n$, there exists a systematic code of minimum distance $n-k$. Furthermore, we prove that for rank modulation, systematic codes have the same capacity as general error-correcting codes. This result establishes that asymptotically, systematic codes are as strong in their error correction capability as general codes.

The rest of the paper is organized as follows. In Section II, we define some terms and show properties of systematic codes. In Section III, we study systematic codes that correct one error. In Section IV, we study codes that correct multiple errors. In Section V, we present the capacity of systematic codes, which matches the capacity of general codes. Due to space limitation, we skip some details. Interested readers can refer to [15] for the full paper.

II. Terms and Properties

In this section, we define some terms for systematic codes, and show its basic properties. Let $C \subseteq \mathcal{S}_n$ denote a general (n,k) systematic error-correcting code for rank modulation. Given a codeword $\mathbf{x} = [x_1, x_2, \cdots, x_n] \in C$, we call the permutation induced by the first k cells (i.e., the information cells) $\mathbf{a} = [a_1, a_2, \cdots, a_k] \in \mathcal{S}_k$ the *information sector* of the codeword $\mathbf{x}$. More specifically, if $c_1, c_2, \cdots, c_n$ are the n cells' levels that induce the permutation $[x_1, x_2, \cdots, x_n] \in C$, then we have $c_{a_1} > c_{a_2} > \cdots > c_{a_k}$. Clearly, the information sector $[a_1, a_2, \cdots, a_k]$ is a subsequence of its codeword $[x_1, x_2, \cdots, x_n]$; namely, $[a_1, a_2, \cdots, a_k] = [x_{i_1}, x_{i_2}, \cdots, x_{i_k}]$ for some $1 \leq i_1 < i_2 < \cdots < i_k \leq n$.

Example 1. *Let* $k = 4$ *and* $n = 6$. *Let* $c_1 = 1.0$, $c_2 = 2.1$, $c_3 = 0.8$, $c_4 = 0.2$, $c_5 = 1.5$, $c_6 = 0.6$. *Then the permutation induced by the* $n = 6$ *cells is* $[2,5,1,3,6,4]$. *The permutation induced by the* $k = 4$ *information cells is* $[2,1,3,4]$. *We can see that* $[2,1,3,4]$ *is a subsequence of* $[2,5,1,3,6,4]$. □

Given a permutation $\mathbf{x} = [x_1, x_2, \cdots, x_n] \in \mathcal{S}_n$, we can see it as constructed by sequentially inserting $1, 2, \cdots, n$ into an initially- empty permutation. Hence, we define the *insertion vector* of $\mathbf{x}$ as the positions of inserting $1, 2, \cdots, n$. Specifically, for $1 \leq i \leq n$, let $g_i(\mathbf{x})$ denote the position of the insertion of the integer i. That is, if $p \in \{1, 2, \cdots, n\}$ denotes the integer such that $x_p = i$, then

$$g_i(\mathbf{x}) = |\{j | 1 \leq j < p, x_j < i\}|.$$

Then we have the insertion vector

$$\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_n(\mathbf{x})] \in \mathbb{Z}_1 \times \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_n,$$

where $\mathbb{Z}_i = \{0, 1, 2, ..., i-1\}$. Note that given $\mathbf{g}(\mathbf{x})$, we can reconstruct $\mathbf{x}$ uniquely. It has been shown that for any $\mathbf{x}, \mathbf{y} \in \mathcal{S}_n$ [1],

$$d_\tau(\mathbf{x}, \mathbf{y}) \geq \sum_{i=1}^{n} |g_i(\mathbf{x}) - g_i(\mathbf{y})|.$$

For an (n,k) systematic code, it is required that for every permutation $\mathbf{a} = [a_1, a_2, \cdots, a_k] \in \mathcal{S}_k$, there is exactly one codeword with $\mathbf{a}$ as its information sector, which we will denote by $\mathbf{x_a}$. The code has $k!$ codewords, and we define its *rate* as $\frac{\ln k!}{\ln n!}$. Given an information sector $\mathbf{a} \in \mathcal{S}_k$, we can get the insertion vector of its codeword $\mathbf{x_a}$, namely,

$$\begin{aligned} \mathbf{g}(\mathbf{x_a}) &= [g_1(\mathbf{x_a}), g_2(\mathbf{x_a}), ..., g_n(\mathbf{x_a})] \\ &= [g_1(\mathbf{a}), ..., g_k(\mathbf{a}), g_{k+1}(\mathbf{x_a}), ... g_n(\mathbf{x_a})]. \end{aligned}$$

It means that $\mathbf{x_a}$ can be constructed from $\mathbf{a}$ in the following way: First, we insert $k+1$ (namely, the $(k+1)$th cell) into the permutation $[a_1, a_2, \cdots, a_k]$ at the position $g_{k+1}(\mathbf{x_a}) \in \mathbb{Z}_{k+1}$; next, we insert the integer $k+2$ (namely, the $(k+2)$th cell) at the position $g_{k+2}(\mathbf{x_a}) \in \mathbb{Z}_{k+2}$; and so on. (The last integer to insert is n.) To design good systematic codes, given the information permutation $\mathbf{a}$, we need to find $[g_{k+1}(\mathbf{x_a}), g_{k+2}(\mathbf{x_a}), ..., g_n(\mathbf{x_a})]$ appropriately to maximize the code's minimum distance.

Example 2. *Let* $k = 4$ *and* $n = 6$. *If* $\mathbf{a} = [1,3,2,4]$, $g_5(\mathbf{x_a}) = 3$ *and* $g_6(\mathbf{x_a}) = 0$, *then* $\mathbf{x_a} = [6,1,3,2,5,4]$. □

The following theorem shows how the insertion of redundant cells into the information sector affects the Kendall's τ-distance between codewords.

Theorem 3. *Given two permutations* $\mathbf{a}, \mathbf{b} \in \mathcal{S}_k$, *the Kendall's* τ*-distance between* $\mathbf{x_a}$ *and* $\mathbf{x_b}$ *satisfies the inequality*

$$d_\tau(\mathbf{x_a}, \mathbf{x_b}) \geq d_\tau(\mathbf{a}, \mathbf{b}) + \sum_{i=k+1}^{n} |g_i(\mathbf{x_a}) - g_i(\mathbf{x_b})|.$$

III. One-error-correcting Codes

In this section, we analyze and design systematic codes for correcting one error. Such codes have minimum distance 3. In particular, we present a family of $(k+2,k)$ systematic codes, where either k or $k+1$ is a prime number. It will be shown that the codes have optimal sizes among systematic codes, unless perfect systematic one-error-correcting codes, which meet the sphere-packing bound, exist.

A. Properties of One-error-correcting Codes

Given a permutation $\mathbf{x} \in \mathcal{S}_n$, the ball of radius r centered at $\mathbf{x}$, denoted by $\mathfrak{B}_r(\mathbf{x})$, is the set of permutations in $\mathcal{S}_n$ that are within distance r from $\mathbf{x}$. Namely, $\mathfrak{B}_r(\mathbf{x}) = \{\mathbf{y} \in \mathcal{S}_n | d_\tau(\mathbf{x}, \mathbf{y}) \leq r\}$, for $0 \leq r \leq \frac{n(n-1)}{2}$. (The maximum Kendall's τ-distance for any two permutations in $\mathcal{S}_n$ is $\frac{n(n-1)}{2}$. [8]) A simple relabeling argument suffices to show that the size of a ball does not depend on the choice of its center. So we use $|\mathfrak{B}_r(n)|$ to denote $|\mathfrak{B}_r(\mathbf{x})|$ for any $\mathbf{x} \in S_n$. It can be proved that [15] for any $0 \leq r \leq \frac{n(n-1)}{2}$,

$$|\mathfrak{B}_r(n)| \leq \binom{n+r-1}{n-1}.$$

An r-error-correcting code $C \subseteq \mathcal{S}_n$ for rank modulation needs to satisfy the sphere-packing bound: $|C| \leq \frac{n!}{|\mathfrak{B}_r(n)|}$. If the inequality in the above bound becomes equality, we call the code *perfect*. For one-error-correcting codes, since $|\mathfrak{B}_1(n)| = n$, the following result holds.

Theorem 4. *A systematic (n, k) one-error-correcting code for rank modulation is perfect if and only if $n = k + 1$. More generally, a perfect one-error-correcting code, – systematic or not, – of length n has $(n-1)!$ codewords.*

It is known that perfect codes are often rare. Well-known examples include binary codes, where the only perfects codes are Hamming codes and Golay codes, and Lee metric codes in three-dimensional and higher-dimensional spaces [6]. For rank modulation, there is a simple $(3, 2)$ one-error-correcting code that is perfect: $\{[1, 2, 3], [3, 2, 1]\}$. However, beside this trivial code, no other perfect code has been found yet. If we add the requirement that the code needs to be systematic, it will be even harder for such codes to exist. For instance, it can be proved that there does not exist any perfect systematic one-error-correcting code when $k = 3$.

Theorem 5. *There does not exist any $(4, 3)$ systematic one-error-correcting code for rank modulation.*

For any given $k \geq 3$, if the perfect $(k+1, k)$ code does not exist, then the $(k+2, k)$ code becomes the optimal systematic code.

B. Construction of $(k + 2, k)$ One-error-correcting Codes

We now present the construction that builds a family of $(k + 2, k)$ systematic one-error-correcting codes.

Construction 6. *Let $k \geq 3$ be an integer such that either k or $k + 1$ is a prime number. Given any information sector $\mathbf{a} = [a_1, a_2, ..., a_k] \in \mathcal{S}_k$, let $g_{k+1}(\mathbf{x_a}) \in \mathbb{Z}_{k+1}, g_{k+2}(\mathbf{x_a}) \in \mathbb{Z}_{k+2}$ be the positions of inserting $k + 1$ and $k + 2$. We set*

$$\begin{array}{l} g_{k+1}(\mathbf{x_a}) = \sum_{i=1}^{k}(2i-1)a_i \mod m \\ g_{k+2}(\mathbf{x_a}) = \sum_{i=1}^{k}(2i-1)^2 a_i \mod m \end{array} \quad (1)$$

where $m = k$ if k is a prime number and $m = k + 1$ if $k + 1$ is a prime number. □

The following theorem shows that the above code can correct one error.

Theorem 7. *The $(k + 2, k)$ systematic code in Construction 6 has minimum distance at least 3. Hence it is a one-error-correcting code.*

Proof: In the $(k + 2, k)$ code of Construction 6, either k or $k + 1$ is a prime number. Let us first consider the case that k is a prime number. Assume that $\mathbf{a} = [a_1, a_2, \cdots, a_k] \in \mathcal{S}_k$ and $\mathbf{b} = [b_1, b_2, \cdots, b_k] \in \mathcal{S}_k$ are two distinct information sectors, whose corresponding codewords are $\mathbf{x_a}, \mathbf{x_b} \in \mathcal{S}_n$, respectively. Our goal is to prove that $d_\tau(\mathbf{x_a}, \mathbf{x_b}) \geq 3$. We consider three cases:

1) Case 1: $d_\tau(\mathbf{a}, \mathbf{b}) \geq 3$. In this case, we have $d_\tau(\mathbf{x_a}, \mathbf{x_b}) \geq d_\tau(\mathbf{a}, \mathbf{b}) \geq 3$.
2) Case 2: $d_\tau(\mathbf{a}, \mathbf{b}) = 1$. In this case, we can write $\mathbf{b}$ as $\mathbf{b} = [b_1, b_2, \cdots, b_k] = [a_1, a_2, \cdots, a_{i+1}, a_i, \cdots, a_k]$ for some $i \in \{1, 2, \cdots, k-1\}$. If we define $\Delta = a_{i+1} - a_i$, then we get
$$g_{k+1}(\mathbf{x_a}) - g_{k+1}(\mathbf{x_b}) = 2\Delta \pmod{\text{k}}.$$
Since $1 \leq |\Delta| \leq k - 1$ and $k \geq 3$ is a prime number, we know that 2Δ is not a multiple of k. As a result, we get $|g_{k+1}(\mathbf{x_a}) - g_{k+1}(\mathbf{x_a})| \geq 1$.
Similarly, we have
$$g_{k+2}(\mathbf{x_a}) - g_{k+2}(\mathbf{x_b}) = 8i\Delta \pmod{\text{k}},$$
where $8i\Delta$ is not a multiple of k, either, because $1 \leq i, |\Delta| \leq k-1$ and $k \geq 3$ is a prime number. This implies that $|g_{k+2}(\mathbf{x_a}) - g_{k+2}(\mathbf{x_b})| \geq 1$.
So by Theorem 3, we get $d_\tau(\mathbf{x_a}, \mathbf{x_b}) \geq d_\tau(\mathbf{a}, \mathbf{b}) + |g_{k+1}(\mathbf{x_a}) - g_{k+1}(\mathbf{x_b})| + |g_{k+2}(\mathbf{x_a}) - g_{k+2}(\mathbf{x_b})| \geq 1 + 1 + 1 = 3$.
3) Case 3: $d_\tau(\mathbf{a}, \mathbf{b}) = 2$. In this case, it takes at least two adjacent transpositions to change the permutation $\mathbf{a}$ into $\mathbf{b}$. These two transpositions can be either separated or adjacent to each other. By considering the two cases separately (detailed proof omitted due to space limitation), we can get that $d_\tau(\mathbf{x_a}, \mathbf{x_b}) \geq 3$.

Therefore, we can conclude that when k is a prime number, for any two distinct codewords $\mathbf{x_a}, \mathbf{x_b}$, their distance is at least 3. When $k + 1$ is a prime number, we can apply the same procedure for the proof, – by only replacing "mod k" with "mod $k + 1$", – and get the result that $d_\tau(\mathbf{x_a}, \mathbf{x_b}) \geq 3$. And that concludes the proof. ■

We now present the encoding and decoding algorithms of the $(k + 2, k)$ systematic code. Let $L = \{0, 1, \cdots, k! - 1\}$ denote the set of information symbols to encode. (If the input are information bits, they can be easily mapped to the information symbols in L via enumerative source coding. L can be rounded down to a power of 2.) For encoding, given an information symbol $\ell \in L$, it can be mapped to its corresponding permutation (i.e., information sector) $\mathbf{a} \in \mathcal{S}_k$ in time linear in k [10]. Based on Construction 6, the insertion vector $(g_{k+1}(\mathbf{x_a}), g_{k+2}(\mathbf{x_a}))$ can be directly computed, which gives us the codeword $\mathbf{x_a}$. That completes the encoding algorithm. The decoding algorithm of the construction is also efficient: Given the received codeword $\mathbf{y}$, let $\mathbf{b}$ denote its information sector. If there is an error in $\mathbf{b}$, i.e., $\mathbf{b} \neq \mathbf{a}$, then we can write $\mathbf{a} = [b_1, ..., b_{i+1}, b_i, ..., b_k]$ for some i with $1 \leq i \leq k-1$. Based on the construction, this i can be determined by solving a simple equation, hence, making the decoding process very efficient.

IV. Multi-error-correcting Codes

In this section, we study the design of systematic codes that correct multiple errors, and prove that for any $2 \leq k < n$, there exists an (n, k) systematic code of minimum distance $n-k$.

First, we present a generic scheme for constructing an (n, k) systematic code of minimum distance d. The scheme is based on greedy searching, hence, not explicit. But the analysis of this scheme is very useful for proving the existence of codes with certain parameters, and for deriving the capacity of systematic codes.

Construction 8. *Let $2 \leq k < n$ and $d \geq 1$. In this scheme, we construct an (n, k) systematic code of minimum distance d. It uses a greedy approach for choosing codewords as follows. Let $\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_{k!}}$ denote the $k!$ permutations in $\mathcal{S}_k$, respectively. For $i = 1, 2, \cdots, k!$, we choose the codeword $\mathbf{x_{s_i}}$ whose information sector is $\mathbf{s_i}$ as follows: Among all the permutations in $\mathcal{S}_n$ that contain $\mathbf{s_i}$ as their information sector, choose a permutation $\mathbf{x_{s_i}}$ such that*

$$\forall j \in \{1, 2, \cdots, i-1\}, d_\tau(\mathbf{x_{s_i}}, \mathbf{x_{s_j}}) \geq d. \qquad (2)$$

If all the $k!$ codewords $\mathbf{x_{s_1}}, \mathbf{x_{s_2}}, \cdots, \mathbf{x_{s_{k!}}}$ can be generated successfully this way, we obtain an (n, k) systematic code of minimum distance d. □

Note that given any $\mathbf{a} \in \mathcal{S}_k$, there are $(k+1) \times (k+2) \times \cdots \times n = \frac{n!}{k!}$ permutations in $\mathcal{S}_n$ that have $\mathbf{a}$ as their information sector. For the above code construction to succeed, $n-k$ needs to be sufficiently large. In the following theorem, we derive a bound for the parameters.

Theorem 9. *Construction 8 can successfully build an (n, k) systematic code of minimum distance d if*

$$\sum_{l=1}^{d-1} \binom{k+l-2}{l} 2^{\min(d-l-1, n-k)} \binom{d-l-1+n-k}{n-k} < \frac{n!}{k!} \qquad (3)$$

Proof: In Construction 8, for any information sector $\mathbf{s_i} \in \mathcal{S}_k$ (where $1 \leq i \leq k!$), there are $\frac{n!}{k!}$ possible choices for the vector $[g_{k+1}(\mathbf{x_{s_i}}), g_{k+2}(\mathbf{x_{s_i}}), ..., g_n(\mathbf{x_{s_i}})]$. Our goal is to make sure that at least one of them – which will become the corresponding codeword $\mathbf{x_{s_i}}$ – can guarantee to satisfy the requirement in (2).

Let us consider the maximum number of choices for the vector $[g_{k+1}(\mathbf{x_{s_i}}), g_{k+2}(\mathbf{x_{s_i}}), ..., g_n(\mathbf{x_{s_i}})]$ whose corresponding permutations in $\mathcal{S}_n$ are at distance less than d from at least one permutation in $\{\mathbf{x_{s_1}}, \mathbf{x_{s_2}}, ..., \mathbf{x_{s_{i-1}}}\}$. Such vectors cannot be chosen for the codeword $\mathbf{x_{s_i}}$. Our proof is based on two main observations: First, given $\mathbf{s_i} \in \mathcal{S}_k$, let N_l be the number of permutations in $\mathcal{S}_k$ whose distance to $\mathbf{s_i}$ is l, then

$$1 + \sum_{l=1}^{j} N_l \leq |\mathfrak{B}_j(k)| \leq \binom{k+j-1}{k-1} \qquad (4)$$

for $1 \leq j \leq d-1$.

Second, given $\mathbf{s_j}$ with $j < i$, if $d_\tau(\mathbf{s_i}, \mathbf{s_j}) = l$, there are at most

$$2^{\min(d-j-1, n-k)} \binom{d-j-1+n-k}{n-k} \qquad (5)$$

assignments for $[g_{k+1}(\mathbf{x_{s_i}}), g_{k+2}(\mathbf{x_{s_i}}), ..., g_n(\mathbf{x_{s_i}})]$ such that $d_\tau(\mathbf{x_{s_j}}, \mathbf{x_{s_i}}) \leq d-1$.

From (4) and (5), we can get that the total number of unavailable assignments for $[g_{k+1}(\mathbf{x_{s_i}}), g_{k+2}(\mathbf{x_{s_i}}), ..., g_n(\mathbf{x_{s_i}})]$ is at most

$$\sum_{l=1}^{d-1} \binom{k+l-2}{l} 2^{\min(d-l-1, n-k)} \binom{d-l-1+n-k}{n-k}.$$

This completes the proof. ■

From this theorem, we can further get the following result.

Theorem 10. *For any $k \geq 2$ and $d \geq 1$, there exists a $(k+d, k)$ systematic code of minimum distance d.*

Now, we present an explicit construction of systematic multi-error-correcting codes, by slightly modifying the multi-error-correcting codes derived in [11]. The idea is that given any two integers $g_i(\mathbf{x_a}), g_i(\mathbf{x_b}) < 2^m$, there exists a function $\phi_m : \mathbb{Z}_{2^m} \to \{0, 1\}^m$ (called Gray map) such that

$$|g_i(\mathbf{x_a}) - g_i(\mathbf{x_b})| \geq d_H(\phi_m(g_i(\mathbf{x_a})), \phi_m(g_i(\mathbf{x_b}))),$$

where d_H indicates the Hamming distance between two binary vectors. As a result, we can convert the problem of constructing rank modulation codes to the problem of constructing binary error-correcting codes in Hamming space. To make the code being systematic, we use $\phi_{\lceil \log_2 i \rceil}$ with $1 \leq i \leq k$ for the mapping of information part, instead of using $\phi_{\lfloor \log_2 i \rfloor}$ in the original construction.

Construction 11. *Let $2 \leq k < n$, we construct an (n, k) systematic rank modulation code, denoted by $C_\tau \subset \mathcal{S}_n$. Given any information sector $\mathbf{a} \in \mathcal{S}_k$, to construct its codeword $\mathbf{x_a} \in C_\tau$, we first construct $\mathbf{x_a}$'s image in a binary systematic code C_H, that is*

$$\begin{aligned} f(\mathbf{x_a}) = &[\phi_{\lceil \log_2 1 \rceil}(g_1(\mathbf{a})), ..., \phi_{\lceil \log_2 k \rceil}(g_k(\mathbf{a})), \\ &\phi_{\lfloor \log_2 (k+1) \rfloor}(g_{k+1}(\mathbf{x_a})), ..., \phi_{\lfloor \log_2 n \rfloor}(g_n(\mathbf{x_a}))]. \end{aligned}$$

In $f(\mathbf{x_a})$, the first $k' = \sum_{i=1}^{k} \lceil \log_2 i \rceil$ bits are the information bits and they can be obtained from the information sector $\mathbf{a}$ directly. The rest $r' = \sum_{i=k+1}^{n} \lfloor \log_2 i \rfloor$ bits are the parity-check bits based on the encoding of C_H. Then we can get $\mathbf{x_a} \in \mathcal{S}_n$ from $f(\mathbf{x_a})$ uniquely. If C_H is an $(k'+r', k')$ binary systematic code correcting t errors, then C_τ is an (n, k) systematic rank modulation code correcting t errors.

V. CAPACITY OF SYSTEMATIC CODES

In this section, we prove that for rank modulation, systematic error-correcting codes achieve the same capacity as general error-correcting codes. In [1], Barg and Mazumdar have derived the capacity of general error-correcting codes for rank modulation. Let $A(n,d)$ denote the maximum size of a code of length n and minimum distance d. (So the code is a subset of $\mathcal{S}_n$.) Define the capacity of error-correcting codes of minimum distance d as $C(d) = \lim_{n\rightarrow\infty} \frac{\ln A(n,d)}{\ln n!}$. It is shown in [1] that

$$C(d) = \begin{cases} 1, & \text{if } d = O(n) \\ 1-\epsilon, & \text{if } d = \Theta(n^{1+\epsilon}) \text{ with } 0<\epsilon<1 \\ 0, & \text{if } d = \Theta(n^2). \end{cases} \quad (6)$$

For systematic codes, let $k(n,d)$ denote the maximum number of information cells that can exist in systematic codes of length n and minimum distance d. (Such codes are $(n, k(n,d))$ systematic codes, and have $k(n,d)!$ codewords.) The *capacity* of systematic codes of minimum distance d is

$$C_{sys}(d) = \lim_{n\rightarrow\infty} \frac{\ln k(n,d)!}{\ln n!}.$$

The following theorem shows that systematic codes have the same capacity as general codes.

Theorem 12. *The capacity of systematic codes of minimum distance d is*

$$C_{sys}(d) = \begin{cases} 1, & \text{if } d = O(n) \\ 1-\epsilon, & \text{if } d = \Theta(n^{1+\epsilon}) \text{ with } 0<\epsilon<1 \\ 0, & \text{if } d = \Theta(n^2). \end{cases}$$

Proof: Since systematic codes are a special case of general error-correcting codes, by Equation (6), it is sufficient to prove

$$C_{sys}(d) \geq \begin{cases} 1, & \text{if } d = O(n) \\ 1-\epsilon, & \text{if } d = \Theta(n^{1+\epsilon}) \text{ with } 0<\epsilon<1 \\ 0, & \text{if } d = \Theta(n^2). \end{cases}$$

According to Theorem 9, there exists an (n,k) systematic code of minimum distance d if k is the maximum integer that satisfies

$$\binom{k+d}{d} 2^n \binom{d+n-k}{n-k} < \frac{n!}{k!}.$$

For such k, we have $k(n,d) \geq k$. For convenience, let $\alpha = \lim_{n\rightarrow\infty} \frac{k}{n}$ be a constant. In this case, if $\alpha > 0$,

$$\begin{aligned} C_{sys}(d) &= \lim_{n\rightarrow\infty} \frac{\ln k(n,d)!}{\ln n!} \geq \lim_{n\rightarrow\infty} \frac{\ln k!}{\ln n!} \\ &= \lim_{n\rightarrow\infty} \frac{\alpha n \log(\alpha n)}{n \log n} = \alpha. \end{aligned}$$

To prove the final conclusion, we will show that if $d = O(n)$, then $\alpha = 1$; if $d = \Theta(n^{1+\epsilon})$, then $\alpha \geq 1-\epsilon$. (If $d = \Theta(n^2)$, the result $\alpha \geq 0$ is trivial).

Based on the definition of k, we can get

$$\lim_{n\rightarrow\infty} \frac{\ln \binom{k+d}{d} 2^n \binom{d+n-k}{n-k}}{\ln \frac{n!}{k!}} = 1 \quad (7)$$

We consider two cases:

1) If $d = O(n)$, we have $d \leq \beta n$ for some $\beta > 0$. By Stirling's approximation, the formula above yields

$$\lim_{n\rightarrow\infty} \frac{(\alpha+\beta) n \ln \frac{\alpha+\beta}{\alpha\beta} + n\ln 2 + (\beta+1-\alpha) n \ln \frac{\beta+1-\alpha}{(1-\alpha)\beta}}{n\ln n - \alpha n \ln(\alpha n)} \geq 1$$

which shows that $n \ln n - \alpha n \ln(\alpha n) = O(n)$. Hence α approaches 1 as $n \rightarrow \infty$.

2) If $d = \Theta(n^{1+\epsilon})$ for $0<\epsilon<1$, by applying Stirling's approximation to Equation (7), we get

$$\lim_{n\rightarrow\infty} \frac{n\ln d - k\ln k - (n-k)\ln(n-k) + O(n)}{n\ln n - k\ln k + O(n)} = 1.$$

Since $k = \alpha n$ and $d = \Theta(n^{1+\epsilon})$, we get

$$\lim_{n\rightarrow\infty} \frac{(1+\epsilon) n\ln n - \alpha n \ln n - (1-\alpha) n \ln}{(1-\alpha) n \ln n} = 1.$$

That leads to $\alpha \geq 1-\epsilon$.

Based on the above analysis and the fact that $S_{sys}(d) \geq \alpha$, we get the final conclusion. ■

ACKNOWLEDGMENT

This work was supported in part by an NSF grant ECCS-0801795 and by a BSF grant 2010075.

REFERENCES

[1] A. Barg and A. Mazumdar, "Codes in permutations and error correction for rank modulation," in *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3158–3165, 2010.

[2] G. W. Burr *et al.*, "Phase change memory technology," in *Journal of Vacuum Science and Technology*, vol. 28, no. 2, pp. 223-262, March 2010.

[3] P. Cappelletti, C. Golla, P. Olivo and E. Zanoni (*Ed.*), *Flash memories*, Kluwer Academic Publishers, 1st Edition, 1999.

[4] T. M. Cover, "Enumerative source coding," *IEEE Transactions on Information Theory*, vol. IT-19, no. 1, pp. 73–77, Jan. 1973.

[5] M. Deza and H. Huang, "Metrics on permutations, a survey," *J. Comb. Inf. Sys. Sci.*, vol. 23, pp. 173–185, 1998.

[6] S. W. Golomb and L. R. Welch, "Perfect codes in the Lee metric and the packing of polyominoes," *SIAM J. Appl. Math.*, vol. 18, no. 2, pp. 302–317, 1970.

[7] A. Jiang, R. Mateescu, M. Schwartz and J. Bruck, "Rank modulation for flash memories," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, pp. 1731–1735, July 2008.

[8] A. Jiang, M. Schwartz and J. Bruck, "Error-correcting codes for rank modulation," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, pp. 1736–1740, July 2008.

[9] A. Jiang, M. Schwartz and J. Bruck, "Correcting charge-constrained errors in the rank-modulation scheme," in *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2112–2120, 2010.

[10] M. Mares and M. Straka, "Linear-time ranking of permutations," *Algorithms-ESA*, pp. 187–193, 2007.

[11] A. Mazumdar, A. Barg and G. Zémor, "Construction of rank modulation codes," *IEEE Internatinal Sympos. Inform. Theory*, pp. 834–838, 2011.

[12] A. Mazumdar, A. Barg and G. Zémor, "Parameters of rank modulation codes: examples," *Annual conference on communication, control and computing*, pp. 13–17, 2011.

[13] M. Schwartz and I. Tamo, "Optimal permutation anticodes with the infinity norm via permanents of $(0,1)$-matrices," in *Journal of Combinatorial Theory*, Series A, vol. 118, pp. 1761–1774, 2011.

[14] I. Tamo and M. Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," in *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2551–2560, June 2010.

[15] H. Zhou, A. Jiang and J. Bruck, "Systematic Error-Correcting Codes for Rank Modulation," *http://paradise.caltech.edu/etr.html*, ETR112, California Institute of Technology, 2011.