Systematic Error-Correcting Codes for Rank Modulation

Hongchao Zhou
Electrical Engineering Dept.
California Institute of Technology
Pasadena, CA 91125
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
Electrical Engineering Dept.
California Institute of Technology
Pasadena, CA 91125
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; however, existing results have bee limited.**

In this work, we explore a new approach, *systematic error-correcting codes for rank modulation*. 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 rates 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. The concrete constructions of error-correcting codes, however, have been very limited.

There has also been some work on error-correcting codes for rank modulation equipped with the L_∞ distance [11], [12]. The distance metric is more appropriate for cells where the noise in cell levels has limited magnitudes.

In this paper, we explore a new approach for code design: *systematic error-correcting codes for rank modulation*. 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. 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 rates 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 $2 \le k < n$, 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. In Section VI, we present the concluding remarks.

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) $\alpha = [a_1, a_2, \cdots, a_k] \in \mathcal{S}_k$ 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 $1 \le i_1 < i_2 < \cdots < i_k \le n$.

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 $[2,5,1,3,6,4]$.* □

For an (n,k) systematic code, it is required that for every permutation $\alpha = [a_1, a_2, \cdots, a_k] \in \mathcal{S}_k$, there is exactly one codeword with α as its information sector, which we will denote by $f(\alpha)$. The code has $k!$ codewords, and we define its *rate* as $\frac{\ln k!}{\ln n!}$. Given an information sector $\alpha \in \mathcal{S}_k$, we can see its corresponding codeword $f(\alpha) = [x_1, x_2, \cdots, x_n] \in \mathcal{S}_n$ as constructed this way: First, we insert the integer $k+1$ (namely, the $(k+1)$th cell) into the permutation $[a_1, a_2, \cdots, a_k]$, where there are $k+1$ possible positions to insert it, which we label by $\{0, 1, \cdots, k\}$ from left to right; next, we insert the integer $k+2$ (namely, the $(k+2)$th cell) into the permutation that is induced by the first $k+1$ cells, where there are $k+2$ possible positions to insert it, which we label by $\{0, 1, \cdots, k+1\}$ from left to right; and so on $\cdots$; and finally, we insert the integer n (namely, the nth cell) into the permutation induced by the first $n-1$ cells, where there are n possible positions to insert it, which we label by $\{0, 1, \cdots, n-1\}$ from left to right. More specifically, for $1 \le i \le n-k$, let $p \in \{1, 2, \cdots, n\}$ be the integer such that $x_p = k+i$, and we define $h_i(\alpha)$ as

$$h_i(\alpha) = |\{j | 1 \le j < p, x_j < k+i\}|.$$

Then the integer $h_i(\alpha) \in \mathbb{Z}_{k+i} = \{0, 1, \cdots, k+i-1\}$ denotes the position of the insertion of the integer $k+i$ mentioned above. We call

$$(h_1(\alpha), h_2(\alpha), \cdots, h_{n-k}(\alpha)) \in \mathbb{Z}_{k+1} \times \mathbb{Z}_{k+2} \times \cdots \times \mathbb{Z}_n$$

the *insertion vector* for α. To design good systematic codes, we need to choose the insertion vectors appropriately to maximize the code's minimum distance.

Example 2. *Let $n=6$ and $k=4$. If $\alpha = [1,3,2,4]$, $h_1(\alpha) = 3$ and $h_2(\alpha) = 0$, then $f(\alpha) = [6,1,3,2,5,4]$. That is because by inserting 5 into $[1,3,2,4]$ at position $h_1(\alpha) = 3$, we get $[1,3,2,5,4]$; then by inserting 6 into $[1,3,2,5,4]$ at position $h_2(\alpha) = 0$, we get $[6,1,3,2,5,4]$.* □

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 $\alpha, \beta \in \mathcal{S}_k$, the Kendall's τ-distance between $f(\alpha)$ and $f(\beta)$ satisfies the inequality*

$$d_\tau(f(\alpha), f(\beta)) \ge d_\tau(\alpha, \beta) + \sum_{i=1}^{n-k} |h_i(\alpha) - h_i(\beta)|.$$

Proof: The proof is by induction. As the base case, the inequality is clearly satisfied if $n=k$. Now consider the inductive step. Suppose that the inequality holds for any integer n with $n < k+r$. (Here r is a nonnegative integer.) We need to show that it also holds for $n = k+r$.

Consider a sequence of $d_\tau(f(\alpha), f(\beta))$ adjacent transpositions that changes the permutation $f(\alpha) \in \mathcal{S}_n$ into the permutation $f(\beta) \in \mathcal{S}_n$. Among them, assume that a adjacent transpositions involve the integer n, and b adjacent transpositions do not involve n. (Clearly, $d_\tau(f(\alpha), f(\beta)) = a + b$.) Since the integer n needs to be moved from position $h_{n-k}(\alpha)$ to position $h_{n-k}(\beta)$, we get $a \ge |h_{n-k}(\alpha) - h_{n-k}(\beta)|$. Note that those adjacent transpositions that involve n do not change the relative order of the integers $\{1, 2, \cdots, n-1\}$ in the permutation. So to transform the integers $\{1, 2, \cdots, n-1\}$ from their relative order in permutation $f(\alpha)$ to their relative order in permutation $f(\beta)$, by the induction assumption, we get $b \ge d_\tau(\alpha, \beta) + \sum_{i=1}^{n-k-1} |h_i(\alpha) - h_i(\beta)|$. That leads to the conclusion. ■

Example 4. *Let* $n = 3$ *and* $k = 2$. *If* $\alpha = [1, 2]$, $\beta = [2, 1]$, $h_1(\alpha) = 1$ *and* $h_1(\beta) = 2$, *then* $f(\alpha) = [1, 3, 2]$ *and* $f(\beta) = [2, 1, 3]$. *In this case, the inequality in Theorem* 3 *becomes equality:*

$$d_\tau(f(a), f(b)) = d_\tau(a, b) + |h_1(a) - h_1(b)| = 2.$$

The equality, however, does not always hold. For instance, if $\alpha = [1, 2]$, $\beta = [2, 1]$ *and* $h_1(\alpha) = h_1(\beta) = 1$, *then* $f(\alpha) = [1, 3, 2]$ *and* $f(\beta) = [2, 3, 1]$. *We have*

$$d_\tau(f(\alpha), f(\beta)) = 3 > d_\tau(\alpha, \beta) + |h_1(\alpha) - h_2(\beta)| = 1.$$

□

We now present an inequality for ball sizes in $\mathcal{S}_n$, which will be useful for the analysis of systematic 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, $\mathfrak{B}_r(\mathbf{x}) = \{\mathbf{y} \in \mathcal{S}_n | d_\tau(\mathbf{x}, \mathbf{y}) \le r\}$, for $0 \le r \le \frac{n(n-1)}{2}$. (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 $\mathbf{x} \in S_n$.

The value of $|\mathfrak{B}_r(n)|$ is provided in [8]. It is shown that $|\mathfrak{B}_r(n)| = \sum_{i=0}^{r} e_i$, where e_i is the coefficient of x^i in the polynomial $\prod_{j=1}^{n-1} \frac{x^{j+1}-1}{x-1}$. When $1 \le r \le n$, e_r can be obtained explicitly [1]. In this paper, we will use the following inequality for ball sizes in the analysis of systematic codes.

Lemma 5. *For any* $0 \le r \le \frac{n(n-1)}{2}$,

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

Proof: 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. (The concept of insertion is the same as the one we have discussed when defining *insertion vectors*.) For $1 \le i \le n$, 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

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

Then we have $(g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_n(\mathbf{x})) \in \mathbb{Z}_1 \times \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_n$.

By Theorem 3, it can be seen that for any two permutations $\mathbf{x}, \mathbf{y} \in \mathcal{S}_n$, we have

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

(Note that the same observation has been made in [1].)

Let us consider a ball $\mathfrak{B}_r(\mathbf{x})$ with the center $\mathbf{x} = [n, n-1, ..., 1]$. Since $g_1(\mathbf{x}) = g_2(\mathbf{x}) = \cdots = g_n(\mathbf{x}) = 0$, for any permutation $\mathbf{y} \in \mathcal{S}_n$, we have

$$d_\tau(\mathbf{x}, \mathbf{y}) \ge \sum_{i=1}^{n} |g_i(\mathbf{y}) - g_i(\mathbf{x})| = \sum_{i=1}^{n} g_i(\mathbf{y}) = \sum_{i=2}^{n} g_i(\mathbf{y})$$

with $g_i(\mathbf{y}) \in \mathbb{Z}_i$. (Note that $g_1(\mathbf{y}) = 0$.)

To compute $|\mathfrak{B}_r(\mathbf{x})|$, we let $d_\tau(\mathbf{x}, \mathbf{y}) \le r$. It yields the relaxed condition

$$\sum_{i=2}^{n} g_i(\mathbf{y}) \le r.$$

If we further relax the constraint that $g_i(\mathbf{y}) \le i - 1$ and only consider the constraint that $g_i(\mathbf{y}) \ge 0$, then there are $\binom{n+r-1}{n-1}$ different solutions to $(g_2(\mathbf{y}), g_3(\mathbf{y}), \cdots, g_n(\mathbf{y}))$ for the inequality $\sum_{i=2}^{n} g_i(\mathbf{y}) \le r$. (It is equivalent to the problem of placing r balls in n boxes.) Since every permutation $\mathbf{y} \in \mathcal{S}_n$ can be distinctly determined by its corresponding vector $(g_2(\mathbf{y}), g_3(\mathbf{y}), \cdots, g_n(\mathbf{y}))$, there are at most $\binom{n+r-1}{n-1}$ permutations in $\mathcal{S}_n$ whose distance to $\mathbf{x}$ is at most r. ■

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 rates among systematic codes, unless perfect systematic one-error-correcting codes, which meet the sphere-packing bound, exist.

A. Properties of One-error-correcting Codes

A r-error-correcting code $C \subseteq \mathcal{S}_n$ for rank modulation needs to satisfy the sphere-packing bound: $|C| \le \frac{n!}{|\mathfrak{B}_r(n)|}$. 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 6. *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. In the following, we prove that there does not exist any perfect systematic one-error-correcting code when $k = 3$.

Theorem 7. *There does not exist any perfect* $(4, 3)$ *systematic one-error-correcting code for rank modulation.*

Proof: The proof is by contradiction. Suppose that there exists a perfect $(4, 3)$ systematic one-error-correcting code, which we denote by C. As before, for any permutation $\alpha \in S_3$, we let $f(\alpha) \in \mathcal{S}_4$ denote the unique codeword in C with α as its information sector, and let $(h_1(\alpha))$ denote its insertion vector (of length one). And for convenience of expression in the following analysis, given any two information sectors $\alpha, \beta \in \mathcal{S}_3$, we denote the distance between their corresponding codewords $f(\alpha), f(\beta)$ by $d_\tau^{(f)}(\alpha, \beta)$ instead of $d_\tau(f(\alpha), f(\beta))$.

We first prove that at least one of the codewords in C does not start or end with 4; namely, there exists a permutation $\alpha \in S_3$ such that $h_1(\alpha) \notin \{0, 3\}$. This statement can be proved by contradiction. Assume that every codeword in C either starts with 4 or ends with 4. Without loss of generality, we can let $h_1([1,2,3]) = 3$. Then the only possible choice for $h_1([2,1,3])$ and $h_1([1,3,2])$ is 0 because otherwise, $d_\tau^{(f)}([1,2,3],[2,1,3])$ and $d_\tau^{(f)}([1,2,3],[1,3,2])$ would equal 1, which would contradict the requirement that C has minimum distance at least 3. Hence we get two codewords $[4,2,1,3]$ and $[4,1,3,2]$. However, in this case, their distance equals 2, which contradicts our assumption.

So there exists at least one permutation $\alpha \in \mathcal{S}_3$ such that $h_1(\alpha) \in \{1, 2\}$. Without loss of generality (by symmetry), we can let $\alpha = [1,2,3]$ and let $h_1(\alpha) = 2$. Its corresponding codeword is $[1,2,4,3]$.

We now consider the codewords whose information sectors are $[2,1,3]$, $[1,3,2]$, $[3,1,2]$, $[3,2,1]$, $[2,3,1]$, respectively.

1) $[2,1,3]$ is at distance one from $[1,2,3]$. Hence the only possible codeword with $[2,1,3]$ as its information sector is $[4,2,1,3]$ because otherwise, we would have $d_\tau^{(f)}([2,1,3]),[1,2,3]) < 3$.
2) $[1,3,2]$ is also at distance one from $[1,2,3]$. To make $d_\tau^{(f)}([1,3,2],[1,2,3]) \geq 3$, we have $h_1([1,3,2]) \in \{0, 2\}$. Since it is required that $d_\tau^{(f)}([1,3,2],[2,1,3]) \geq 3$, the only possible value for $h_1([1,3,2])$ is 2. Therefore, the codeword with $[1,3,2]$ as its information sector is $[1,3,4,2]$.
3) With a similar analysis, we get $h_1([3,1,2]) = 0$. Its corresponding codeword is $[4,3,1,2]$.
4) Since it is required that $d_\tau^{(f)}([3,2,1],[3,1,2]) \geq 3$, we need $h_1([3,2,1]) \in \{2,3\}$. Since it is required that $d_\tau^{(f)}([2,3,1],[2,1,3]) \geq 3$, we need $h_1([2,3,1]) \in \{2,3\}$. However, in this case, by enumerating all the possible values for $h_1[3,2,1]$ and $h_1([2,3,1])$, we can see that
$$d_\tau^{(f)}([3,2,1],[2,3,1]) < 3,$$
which is a contradiction.

Based on the above analysis, it can be concluded that there does not exist any $(4,3)$ systematic code correcting one error 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 code. We show such a $(6,4)$ systematic code in the appendix. In the following subsection, we present a family of $(k+2,k)$ systematic codes, where either k or $k+1$ is a prime number.

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 8. *Let $k \geq 3$ be an integer such that either k or $k+1$ is a prime number. Given any information sector $\alpha = [a_1, a_2, ..., a_k] \in \mathcal{S}_k$, let $(h_1(\alpha), h_2(\alpha))$ be its insertion vector. (Namely, $h_1(\alpha) \in \mathbb{Z}_{k+1}$ is the position of inserting $k+1$, and $h_2(\alpha) \in \mathbb{Z}_{k+2}$ is the position of inserting $k+2$.) We set*

$$\begin{array}{c} h_1(\alpha) = \sum_{i=1}^{k}(2i-1)a_i \mod m \\ h_2(\alpha) = \sum_{i=1}^{k}(2i-1)^2 a_i \mod m \end{array} \tag{1}$$

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 9. *The $(k+2,k)$ systematic code in Construction 8 has minimum distance at least 3. Hence it is a one-error-correcting code.*

Proof: In the $(k+2,k)$ code of Construction 8, either k or $k+1$ is a prime number. Let us first consider the case that k is a prime number. Assume that $\alpha = [a_1, a_2, \cdots, a_k] \in \mathcal{S}_k$ and $\beta = [b_1, b_2, \cdots, b_k] \in \mathcal{S}_k$ are two distinct information sectors, whose corresponding codewords are $\mathbf{x} = f(\alpha) \in \mathcal{S}_n$ and $\mathbf{y} = f(\beta) \in \mathcal{S}_n$, respectively. Our goal is to prove that $d_\tau(\mathbf{x}, \mathbf{y}) \geq 3$. We consider three cases:

1) Case 1: $d_\tau(\alpha, \beta) \geq 3$. In this case, we have $d_\tau(\mathbf{x}, \mathbf{y}) \geq d_\tau(\alpha, \beta) \geq 3$.
2) Case 2: $d_\tau(\alpha, \beta) = 1$. In this case, we can write β as $\beta = [b_1, b_2, \cdots, b_k] = [a_1, a_2, \cdots, a_{i+1}, a_i, \cdots, a_k]$ for some $i \in \{1, 2, \cdots, k-1\}$. If we define $\Delta = a_{i+1} - a_i$, then we get
$$h_1(\alpha) - h_1(\beta) = 2\Delta \pmod{\mathrm{k}}.$$
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
$$|h_1(\alpha) - h_1(\beta)| \geq 1.$$
Similarly, we have
$$\begin{array}{rl} & h_2(\alpha) - h_2(\beta) \\ = & (2i-1)^2 a_i + (2i+1)^2(a_i + \Delta) \\ & -(2i-1)^2(a_i + \Delta) - (2i+1)^2 a_i \\ = & 8i\Delta \pmod{\mathrm{k}} \end{array}$$
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 $|h_2(\alpha) - h_2(\beta)| \geq 1$.
So by Theorem 3, we get $d_\tau(\mathbf{x}, \mathbf{y}) = d_\tau(f(\alpha), f(\beta)) \geq d_\tau(\alpha, \beta) + |h_1(\alpha) - h_1(\beta)| + |h_2(\alpha) - h_2(\beta)| \geq 1 + 1 + 1 = 3$.
3) Case 3: $d_\tau(\alpha, \beta) = 2$. In this case, it takes at least two adjacent transpositions to change the permutation α into β. These two transpositions can be either separated (which means that the two pairs of integers involved in the two transposition do not share any common integer) or adjacent to each other (which means that the two pairs

of integers involved in the two transpositions share one common integer). We consider the two cases.

In the first case that the two adjacent transpositions are separated, we can write β as

$$\beta = [a_1, ..., a_{i+1}, a_i, ..., a_{j+1}, a_j, ..., a_k]$$

for some $1 < i+1 < j < k$. Let us define $\Delta_1 = a_{i+1} - a_i$ and $\Delta_2 = a_{j+1} - a_j$. Then we get

$$h_1(\alpha) - h_1(\beta) = 2(\Delta_1 + \Delta_2) \quad (\mathrm{mod\ k}).$$

If $\Delta_1 + \Delta_2$ is not a multiple of k, then $|h_1(\alpha) - h_1(\beta)| \geq 1$. This leads to $d_\tau(\mathbf{x}, \mathbf{y}) \geq d_\tau(\alpha, \beta) + |h_1(\alpha) - h_1(\beta)| \geq 2 + 1 = 3$. If $\Delta_1 + \Delta_2$ is a multiple of k, we can write Δ_2 as $\Delta_2 = tk - \Delta_1$ for some integer $t \in \{-1, 0, 1\}$. Hence

$$\begin{aligned} & h_2(\alpha) - h_2(\beta) \\ = \;& (2i-1)^2 a_i + (2i+1)^2 (a_i + \Delta_1) \\ & +(2j-1)^2 a_j + (2j+1)^2 (a_j + tk - \Delta_1) \\ & -(2i-1)^2 (a_i + \Delta_1) - (2i+1)^2 a_i \\ & -(2j-1)^2 (a_j + tk - \Delta_1) - (2j+1)^2 a_j \\ = \;& 8(j-i)\Delta_1 \quad (\mathrm{mod\ k}) \end{aligned}$$

where $8(j-i)\Delta_1$ is not a multiple of k. So $|h_2(\alpha) - h_2(\beta)| \geq 1$, which leads to $d_\tau(\mathbf{x}, \mathbf{y}) \geq d_\tau(\alpha, \beta) + |h_2(\alpha) - h_2(\beta)| \geq 2 + 1 = 3$.

In the second case that the two transpositions are adjacent to each other, we have either

$$\beta = [a_1, ..., a_{i+2}, a_i, a_{i+1}, ..., a_k]$$

or

$$\beta = [a_1, ..., a_{i+1}, a_{i+2}, a_i, ..., a_k]$$

for some $1 \leq i \leq k-2$.

By defining $\Delta_1 = a_{i+2} - a_{i+1}$ and $\Delta_2 = a_{i+2} - a_i$ (or $\Delta_1 = a_{i+1} - a_i$ and $\Delta_2 = a_{i+2} - a_i$), with the same argument as above, it can be proved that either $|h_1(\alpha) - h_1(\beta)| \geq 1$ or $|h_2(\alpha) - h_2(\beta)| \geq 1$. Therefore we again have $d_\tau(\mathbf{x}, \mathbf{y}) \geq d_\tau(\alpha, \beta) + |h_1(\alpha) - h_1(\beta)| + |h_2(\alpha) - h_2(\beta)| \geq 2 + 1 = 3$.

Therefore, we can conclude that when k is a prime number, for any two distinct codewords $\mathbf{x}, \mathbf{y}$, their distance is at least 3. When $k+1$ is a prime number, we can apply the same procedure for the proof, – with only replacing "mod k" by "mod $k+1$", – and get the result that $d_\tau(\mathbf{x}, \mathbf{y}) \geq 3$. 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.) For encoding, given an information symbol $\ell \in L$, it can be mapped to its corresponding permutation (i.e., information sector) $\alpha \in \mathcal{S}_k$ in time linear in k [10]. Based on Construction 8, the insertion vector $(h_1(\alpha), h_2(\alpha))$ can be directly computed, which gives us the codeword $f(\alpha)$. That completes the encoding algorithm.

We now describe the decoding algorithm. Let $\mathbf{x} \in \mathcal{S}_{k+2}$ denote the correct codeword, and let $\alpha = [a_1, a_2, \cdots, a_k] \in \mathcal{S}_k$ be its information sector. Let $\mathbf{y} \in \mathcal{S}_{k+2}$ denote the received (possibly noisy) codeword, and let $\beta = [b_1, b_2, \cdots, b_k] \in \mathcal{S}_k$ be its information sector. Suppose that there is at most one error in $\mathbf{y}$. A straightforward decoding algorithm is to check all the $k+2$ permutations within distance one from $\mathbf{y}$ (including $\mathbf{y}$ itself), and verify which one of them is the correct codeword. There is, however, a more efficient decoding algorithm that avoids checking the $k+2$ candidate permutations, which we describe below.

Given the received codeword $\mathbf{y}$, let $g_1 \in \mathbb{Z}_{k+1}$ and $g_2 \in \mathbb{Z}_{k+2}$ denote the positions of the insertion of the integers $k+1$ and $k+2$, respectively. Let $f(\beta)$ be the codeword corresponding to the information sector β, which can be computed based on Construction 8. If $d_\tau(f(\beta), \mathbf{y}) \leq 1$, then $f(\beta) = \mathbf{x}$ is the correct codeword and $\beta = \alpha$ is the correct information sector; otherwise, there is an error in β, which we will find as follows. We can write α as $\alpha = [b_1, ..., b_{i+1}, b_i, ..., b_k]$ for some i with $1 \leq i \leq k-1$. In this case, we have $h_1(\alpha) = g_1$ and $h_2(\alpha) = g_2$ because

$$d_\tau(\alpha, \beta) + |h_1(\alpha) - g_1| + |h_2(\alpha) - g_2| \leq d_\tau(f(\alpha), \mathbf{y}) \leq 1,$$

which implies $|h_1(\alpha) - g_1| = 0$ and $|h_2(\alpha) - g_2| = 0$.

According to the proof of Theorem 9, we know that

$$g_1 - h_1(\beta) = h_1(\alpha) - h_1(\beta) = 2(b_i - b_{i+1}) \quad (\mathrm{mod\ m})$$
$$g_2 - h_2(\beta) = 8i(b_i - b_{i+1}) \quad (\mathrm{mod\ m})$$

where m is the prime number in $\{k, k+1\}$. Based on these two equations, we get

$$g_2 - h_2(\beta) = 4i(g_1 - h_1(\beta)) \quad (\mathrm{mod\ m}) \qquad (2)$$

By solving this equation, we can obtain the value for $i \in \{1, 2, \cdots, k-1\}$ that gives us the correct information sector α and its codeword $\mathbf{x} = f(\alpha)$.

We illustrate the decoding algorithm with the following example.

Example 10. *Let $k = 4$ and the correct information sector be $\alpha = [4, 1, 3, 2]$. Based on Equation (1) in Construction 8, we get its codeword $f(\alpha) = [4, 1, 6, 3, 5, 2]$. Assume that one error happened and we receive the noisy word $\mathbf{y} = [1, 4, 6, 3, 5, 2]$, which we decode in the following way. First, from $\mathbf{y}$, we get $\beta = [1, 4, 3, 2]$ and $g_1 = 3, g_2 = 2$. And we have $h_1(\beta) = 4$, $h_2(\beta) = 1$. Since here*

$$d_\tau(f(\beta), \mathbf{y}) \geq |g_1 - h_1(\beta)| + |g_2 - h_2(\beta)| > 1,$$

there is one error in β. From Equation (2), we get $1 = -4i \mod 5$, which gives us $i = 1 \in \{1, 2, 3\}$. So it is determined that the correct information sector is $[4, 1, 3, 2]$. □

Given k, the $(k+2, k)$ code uses the minimum amount of redundancy among systematic codes, unless there exists a perfect and systematic $(k+1, k)$ one-error-correcting code. And compared to the one-error-correcting code presented in [8], the $(k+2, k)$ codes presented here have more efficient encoding and decoding algorithms.

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 $n-k$.

The one-error-correcting code in Construction 8 can be generalized for correcting multiple errors in the following way. Given any information sector $\alpha = [a_1, a_2, \cdots, a_k] \in \mathcal{S}_k$, we set its insertion vector $(h_1(\alpha), h_2(\alpha), \cdots, h_{n-k}(\alpha))$ as follows: For $j = 1, 2, \cdots, n-k$,

$$h_j(\alpha) = \sum_{i=1}^{k} (2i-1)^j a_i \mod m,$$

where $m = k$ if k is a prime number and $m = k+1$ if $k+1$ is a prime number. This gives us a sequence of codes, including a $(10, 4)$ code of minimum distance 5, a $(14, 4)$ code of minimum distance 7, etc. In this section, we explore the existence of more efficient systematic codes.

We present a generic scheme for constructing an (n, k) systematic code of minimum distance d. The scheme is based on greedy searching. Although it is beyond the scope of this paper to obtain efficient encoding and decoding algorithms for it, 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 11. *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 $s_1, s_2, \cdots, s_{k!}$ denote the $k!$ permutations in $\mathcal{S}_k$, respectively, and let W be a set that is initially empty. For $i = 1, 2, \cdots, k!$, we choose the codeword $f(s_i)$ whose information sector is s_i as follows: Among all the permutations in $\mathcal{S}_n$ that contain s_i as their information sector, choose a permutation $\mathbf{x}$ such that*

$$\forall j \in \{1, 2, \cdots, i-1\}, d_\tau(\mathbf{x}, f(s_j)) \geq d; \qquad (3)$$

then we let $f(s_i) = \mathbf{x}$, and insert $f(s_i)$ into the set W. If all the $k!$ codewords $f(s_1), f(s_2), \cdots, f(s_{k!})$ can be generated successfully this way, we obtain an (n, k) systematic code of minimum distance d. □

Note that given any $\alpha \in \mathcal{S}_k$, there are $(k+1) \times (k+2) \times \cdots \times n = \frac{n!}{k!}$ permutations in $\mathcal{S}_n$ that have α 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 12. *Construction 11 can successfully build an (n, k) systematic code of minimum distance d if*

$$\sum_{i=1}^{d-1} \binom{k+i-2}{i} 2^{\min(d-i-1, n-k)} \binom{d-i-1+n-k}{n-k} < \frac{n!}{k!} \qquad (4)$$

Proof: In Construction 11, for any information sector $s_i = \alpha \in \mathcal{S}_k$ (where $1 \leq i \leq k!$), there are $\frac{n!}{k!}$ possible choices for the insertion vector $[h_1(\alpha), h_2(\alpha), ..., h_{n-k}(\alpha)]$. Our goal is to make sure that at least one of them – which will become the corresponding codeword $f(\alpha)$ – can guarantee to satisfy the requirement

$$\forall \mathbf{y} \in W, d_\tau(f(\alpha), \mathbf{y}) \geq d.$$

Note that here $W = \{f(s_j) | j = 1, 2, \cdots, i-1\}$.

Let us consider the maximum number of choices for the insertion vector $[h_1(\alpha), h_2(\alpha), \cdots, h_{n-k}(\alpha)]$ whose corresponding permutations in $\mathcal{S}_n$ are at distance less than d from at least one permutation in W. Such insertion vectors cannot be chosen for the codeword $f(\alpha)$. For any word $\mathbf{y} \in W$, assume that its information sector is β. If $d_\tau(\alpha, \beta) = j \leq d-1$, to make $d_\tau(f(\alpha), \mathbf{y}) \geq d$, it is enough to let

$$\sum_{t=1}^{n-k} |h_t(\alpha) - h_t(\beta)| \geq d - j.$$

Note that here $[h_1(\beta), h_2(\beta), \cdots, h_{n-k}(\beta)]$ is the insertion vector that has been chosen for the information sector β.

Now we are interested in the number of solutions to $[h_1(\alpha), h_2(\alpha), \cdots, h_{n-k}(\alpha)]$ that satisfy the inequality

$$\sum_{t=1}^{n-k} |h_t(\alpha) - h_t(\beta)| \leq d - j - 1.$$

We call such solutions *unavailable combinations* for $[h_1(\alpha), h_2(\alpha), \cdots, h_{n-k}(\alpha)]$. Note that there are at most $\binom{d-j-1+n-k}{n-k}$ possible choices for

$$[|h_1(\alpha) - h_1(\beta)|, |h_2(\alpha) - h_2(\beta)|, \cdots, |h_{n-k}(\alpha) - h_{n-k}(\beta)|].$$

Among them, at most $\min(d-j-1, n-k)$ elements are not zero. Hence the number of unavailable combinations for $[h_1(\alpha), h_2(\alpha), \cdots, h_{n-k}(\alpha)]$ (due to the constraint imposed by $\mathbf{y}$) is at most

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

Let N_j be the number of permutations in $\mathcal{S}_k$ whose distance to α is j. Based on the union bound, the total number of unavailable combinations for $[h_1(\alpha), h_2(\alpha), \cdots, h_{n-k}(\alpha)]$ is at most

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

According to Lemma 5, there are at most $\binom{k+j-1}{k-1}$ permutations in W for which the distance between their information sectors and α is at most j, namely,

$$1 + \sum_{t=1}^{j} N_t \leq \binom{k+j-1}{k-1}$$

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

In this case, it is not hard to prove that N is maximized when

$$N_j = \binom{k+j-1}{k-1} - \binom{k+j-2}{k-1} = \binom{k+j-2}{k}$$

for $k \geq 2$ and $1 \leq j \leq d-1$ because $2^{\min(d-j-1,n-k)}\binom{d-j-1+n-k}{n-k}$ is a deceasing function of j.

As a result, we get

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

Since the total number of possible combinations for $[h_1(\alpha), h_2(\alpha), \cdots, h_{n-k}(\alpha)]$ is $\frac{n!}{k!}$, if $N < \frac{n!}{k!}$, we can always find an available combination such that Equation (3) is satisfied. And this is true for all information sectors. So the conclusion holds. ■

Given k and d, we can calculate the minimum value of n that satisfies the inequality in Theorem 12.

Example 13. *When $d = 3$ and $n = k+2$, the inequality in Theorem 12 can be simplified as*

$$6\binom{k-1}{1} + \binom{k}{2} < (k+1)(k+2),$$

which holds for any $k \geq 2$. Therefore, there exists a $(k+2,k)$ systematic code that corrects one error for any $k \geq 2$. (Note that this result is consistent with the $(k+2,k)$ systematic one-error-correcting code built in Construction 8.) □

Example 14. *When $d = 4$ and $n = k+3$, the inequality in Theorem 12 can be simplified as*

$$40\binom{k-1}{1} + 8\binom{k}{2} + \binom{k+1}{3} < (k+1)(k+2)(k+3),$$

which holds for all $k \geq 2$. Therefore, there exists a $(k+3,k)$ systematic code of minimum distance 4 for any $k \geq 2$. □

We now prove that for any $2 \leq k < n$, there exists an (n,k) systematic code of minimum distance $n-k$.

Theorem 15. *For any $k \geq 2$ and $d \geq 1$, there exists a $(k+d,k)$ systematic code of minimum distance d.*

Proof: Based on Theorem 12, to show that there exists a $(k+d,k)$ systematic code of minimum distance d, we only need to prove

$$\sum_{i=1}^{d-1} \binom{k+i-2}{i} 2^{d-i-1} \binom{2(d-1)-i}{d-1} < \frac{(k+d)!}{k!}$$

for $k \geq 2, d \geq 2$. (The case of $d = 1$ is trivial.)

Here, we consider a stronger condition,

$$\sum_{i=1}^{d-1} \binom{k+i}{i} 2^{d-i-1} \binom{2(d-1)-i}{d-1} < \frac{(k+d)!}{k!} \qquad (5)$$

We define

$$\psi_d(k) = \frac{\sum_{i=1}^{d-1}\binom{k+i}{i} 2^{d-i-1}\binom{2(d-1)-i}{d-1}}{\frac{(k+d)!}{k!}}.$$

Then we would like to show that the ratio between $\psi_d(k+1)$ and $\psi_d(k)$ is at most 1. That is true because

$$\begin{aligned} \frac{\psi_d(k+1)}{\psi_d(k)} &= \frac{\sum_{i=1}^{d-1}\binom{k+1+i}{i} 2^{d-i-1}\binom{2(d-1)-i}{d-1}}{\sum_{i=1}^{d-1}\binom{k+i}{i} 2^{d-i-1}\binom{2(d-1)-i}{d-1}} \\ &\quad \times \frac{\frac{(k+d)!}{k!}}{\frac{(k+1+d)!}{(k+1)!}} \\ &\leq \max_{i=1}^{d-1} \frac{\binom{k+1+i}{i}}{\binom{k+i}{i}} \frac{\frac{(k+d)!}{k!}}{\frac{(k+1+d)!}{(k+1)!}} \\ &\leq \max_{i=1}^{d-1} \frac{k+1+i}{1+k} \frac{1+k}{1+k+d} \leq 1 \end{aligned}$$

This implies that given any $d \geq 2$, $\psi_d(k)$ is a non-increasing function of k. If $\psi_d(2) < 1$ for all $d \geq 2$, then for any $k, d \geq 2$, we have $\psi_d(k) < 1$, which proves the condition in Equation (5). So our task is to prove $\psi_d(2) < 1$, namely,

$$\sum_{i=1}^{d-1} \binom{2+i}{i} 2^{d-i-1} \binom{2(d-1)-i}{d-1} < \frac{(2+d)!}{2!}$$

for $d \geq 2$.

The left side of the inequality is

$$\begin{aligned} & \sum_{i=1}^{d-1} \binom{2+i}{i} 2^{d-i-1} \binom{2(d-1)-i}{d-1} \\ \leq\ & \sum_{i=1}^{d-1} 3 \times 2^{d-2} \binom{2d-3}{d-1} \\ & \times \left(\frac{(i+2)(i+1)}{6} 2^{1-i} \prod_{j=2}^{i} \frac{d-j}{2d-1-j}\right) \\ \leq\ & \sum_{i=1}^{d-1} 3 \times 2^{d-2} \binom{2d-3}{d-1} \left(\frac{1}{2}\right)^{i-1} \\ \leq\ & 6 \times 2^{d-2} \binom{2d-3}{d-1} \end{aligned}$$

Now, we need to show that

$$6 \times 2^{d-2} \binom{2d-3}{d-1} < \frac{(2+d)!}{2!}$$

for any $d \geq 2$. When $2 \leq d \leq 8$, we can show that the inequality holds by computing the exact values. When $d \geq 8$, we define

$$\phi(d) = \frac{6 \times 2^{d-2}\binom{2d-3}{d-1}}{\frac{(2+d)!}{2!}}.$$

Then

$$\frac{\phi(d+1)}{\phi(d)} = \frac{2(2d-1)(2d-2)}{d(d+1)(d+3)} \leq \frac{8}{d} \leq 1.$$

Since $\phi(8) < 1$, we get $\phi(d) < 1$ when $d \geq 8$.

Based on the above analysis, we see that the condition in Equation (5) always holds when $d, k \geq 2$. That leads to the conclusion. ■

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. So they have the same asymptotic performance in terms of the error correction capability.

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

$$C(d) = \lim_{n\to\infty} \frac{\ln A(n,d)}{\ln n!}.$$

It is shown in [1] that

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

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

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

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

Theorem 16. *The capacity of systematic codes of minimum distance d is*

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

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

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

That is because

$$\binom{k+d}{d} 2^n \binom{d+n-k}{n-k}$$

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

for all $n > k \geq 2$ and $d \geq 2$.

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

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

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

$$\lim_{n\to\infty} \frac{\ln \binom{k+d}{d} 2^n \binom{d+n-k}{n-k}}{\ln \frac{n!}{k!}} = 1 \tag{7}$$

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

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

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

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

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

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

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

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

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

VI. Conclusion

In this paper, we study systematic error-correcting codes for rank modulation. We present $(k+2,k)$ systematic codes for correcting one error, and analyze systematic codes that correct multiple errors. We prove that systematic codes have the same capacity as general codes. There are still many open problems for systematic codes for rank modulation. It is important to design multi-error-correcting codes of high rates with efficient encoding and decoding algorithms. It is also important to study codes equipped with distance metrics other than the Kendall's τ-distance, based on the different types of noise that are common in nonvolatile memories.

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] 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.

[12] 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.

APPENDIX

In this appendix, we present an alternative $(6,4)$ systematic code, and prove that it can correct one error.

The code is constructed as follows. Let us first show the adjacency graph for the permutations of $\mathcal{S}_k = \mathcal{S}_4$ in Fig. 1 (a), where two permutations are connected by an edge if and only if their Kendall’s τ-distance is 1. The permutations in $\mathcal{S}_4$ are the permutations induced by the $k = 4$ information cells. And for any two permutations $\alpha, \beta \in \mathcal{S}_4$, their Kendall’s τ-distance $d_\tau(\alpha, \beta)$ equals the shortest-path distance in the adjacency graph in Fig. 1 (a).

Next, we insert a redundant cell – the 5th cell – into the permutations. For every permutation, we place the 5th cell right in the middle. As a result, we get the permutations in Fig. 1 (b). For any two permutations in Fig. 1 (b), they are connected by an edge if and only if their Kendall’s τ-distance is 1. (An interesting thing to notice is that here every node has degree 2 and is in a cycle of length 4.)

In the final step, we insert another redundant cell – the 6th cell – into the permutations. As a result, we get the code in Fig. 1 (c), where the integer beside every codeword is the position of the 6th cell in that codeword (which equals $h_2(\alpha) + 1$ with α being the information sector). The code is a $(6,4)$ systematic code. The following theorem shows that it has minimum distance 3, and therefore is a one-error-correcting code.

Theorem 17. *The $(6,4)$ systematic code in Fig. 1 (c) has minimum distance 3. So it is a one-error-correcting code.*

Proof: Since inserting redundant cells into permutations will only increase the distance between permutations, we just need to focus on the permutation pairs in Fig. 1 (a) that are at distance at most 2 from each other, and show that after adding the $n - k = 2$ redundant cells, their distance is at least 3.

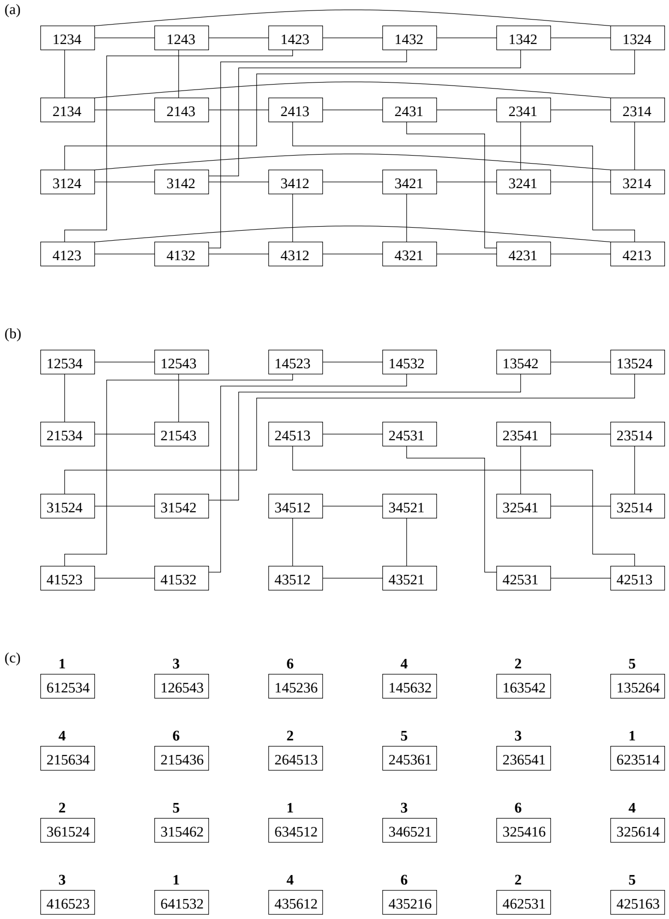

Fig. 1. The construction of an (n,k) systematic one-error-correcting code for $n = 6$ and $k = 4$. (a) The adjacency graph for $\mathcal{S}_4$ (i.e., the permutations induced by the $k = 4$ information cells). (b) Place a redundant cell – the 5th cell – in the middle of every permutation. Here two permutations connected by an edge are still at distance one from each other. (c) Place a redundant cell – the 6th cell – in every permutation. The number beside each permutation indicates the position of the 6th cell in the permutation, which equals $h_2(\alpha)+$ 1 with $\alpha \in \mathcal{S}_4$ being the information sector. Here the distance between every pair of permutations is at least 3. So the code can correct one error.

First, consider the permutation pairs at distance one (i.e., adjacent permutations) in Fig. 1 (a). Every permutation $\alpha \in \mathcal{S}_4$ in Fig. 1 (a) has three neighbors, and they are contained in two cycles: a cycle of length 6 and a cycle of length 4. (For example, the permutation $[1,2,3,4]$ has three neighbors: $[1,2,4,3], [1,3,2,4]$ and $[2,1,3,4]$. The permutations $[1,2,3,4], [1,2,4,3], [1,3,2,4]$ are in a cycle of length 6: $[1,2,3,4] - [1,2,4,3] - [1,4,2,3] - [1,4,3,2] - [1,3,4,2] - [1,3,2,4]$. The permutations $[1,2,3,4], [1,2,4,3], [2,1,3,4]$ are in a cycle of length 4: $[1,2,3,4] - [1,2,4,3] - [2,1,4,3] - [2,1,3,4]$.) We consider the two cases:

- Consider a cycle of length 6. Let $S = (s_1, s_2, s_3, s_4, s_5, s_6)$ denote the positions of the number “6” in the final permutations in Fig. 1 (c). (Those positions are the numbers beside the permutations in Fig. 1 (c).) We can see that either $S = (1,3,6,4,2,5)$ or $S = (6,4,1,3,5,2)$ (or its cyclic shifts or inversions). For example, consider the cycle $[3,4,1,2] - [3,4,2,1] - [3,2,4,1] - [3,2,1,4] - [3,1,2,4] - [3,1,4,2]$ in Fig. 1

(a). The corresponding set of permutations in Fig. 1 (c) is $[6,3,4,5,1,2]-[3,4,6,5,2,1]-[3,2,5,4,1,6]-[3,2,5,6,1,4]-[3,6,1,5,2,4]-[3,1,5,4,6,2]$. For this cycle, we have $S=(1,3,6,4,2,5)$.

As another example, consider the cycle $[2,1,4,3]-[2,1,3,4]-[2,3,1,4]-[2,3,4,1]-[2,4,3,1]-[2,4,1,3]$ in Fig. 1 (a). The corresponding set of permutations in Fig. 1 (c) is $[2,1,5,4,3,6]-[2,1,5,6,3,4]-[6,2,3,5,1,4]-[2,3,6,5,4,1]-[2,4,5,3,6,1]-[2,6,4,5,1,3]$. For this cycle, we have $S=(6,4,1,3,5,2)$.

We see that any two adjacent numbers in the cycle S differ by at least 2. The two corresponding permutations in Fig. 1 (a) have distance 1. (Also note that the adjacency graph has no cycle of length less than 4.) So after inserting the redundant cells, their distance is at least $2+1=3$.

- Similarly, consider a cycle of length 4. Let $S=(s_1,s_2,s_3,s_4)$ denote the positions of the number "6" in the final permutations in Fig. 1 (c). (Those positions are the numbers beside the permutations in Fig. 1 (c).) We can see that either $S=(1,3,6,4)$ or $S=(2,5,2,5)$ (or its cyclic shifts or inversions).

  For example, consider the cycle $[1,2,3,4]-[1,2,4,3]-[2,1,4,3]-[2,1,3,4]$ in Fig. 1 (a). The corresponding set of permutations in Fig. 1 (c) is $[6,1,2,5,3,4]-[1,2,6,5,4,3]-[2,1,5,4,3,6]-[2,1,5,6,3,4]$. For this cycle, we have $S=(1,3,6,4)$.

  As another example, consider the cycle $[2,4,1,3]-[2,4,3,1]-[4,2,3,1]-[4,2,1,3]$ in Fig. 1 (a). The corresponding set of permutations in Fig. 1 (c) is $[2,6,4,5,1,3]-[2,4,5,3,6,1]-[4,6,2,5,3,1]-[4,2,5,1,6,3]$. For this cycle, we have $S=(2,5,2,5)$.

  We see that any two adjacent numbers in the cycle S differ by at least 2. The two corresponding permutations in Fig. 1 (a) have distance 1. So after inserting the redundant cells, their distance is at least $2+1=3$.

So for any two adjacent permutations in Fig. 1 (a), after inserting the redundant cells, their distance is at least 3.

Next, consider the permutation pairs at distance two in Fig. 1 (a). Let $\alpha=[a_1,a_2,a_3,a_4]\in\mathcal{S}_4$ and $\beta=[b_1,b_2,b_3,b_4]\in\mathcal{S}_4$ be two permutations at distance two in Fig. 1 (a). After inserting the 5th cell into them, they become $\alpha'=[a_1,a_2,5,a_3,a_4]\in\mathcal{S}_5$ and $\beta'=[b_1,b_2,5,b_3,b_4]\in\mathcal{S}_5$. (See Fig. 1 (b).) After inserting the 6th cell into them, they become $\alpha''\in\mathcal{S}_6$ and $\beta''\in\mathcal{S}_6$. Let $s_\alpha,s_\beta\in\{1,2,3,4,5,6\}$ denote the positions of the number "6" in α'' and β'', respectively. If $s_\alpha\neq s_\beta$, then clearly $d_\tau(\alpha'',\beta'')\geq 2+1=3$. So we only need to consider the case $s_\alpha=s_\beta$. From Fig. 1, we can see it happens only in a cycle of length 4. For example, consider the cycle $[2,4,1,3]-[2,4,3,1]-[4,2,3,1]-[4,2,1,3]$ in Fig. 1 (a). If $\alpha=[2,4,1,3]$ and $\beta=[4,2,3,1]$, then we have $d_\tau(\alpha,\beta)=2$, $\alpha'=[2,4,5,1,3]$, $\beta'=[4,2,5,3,1]$, $\alpha''=[2,6,4,5,1,3]$, $\beta''=[4,6,2,5,3,1]$, $s_\alpha=2$, $s_\beta=2$. It is easy to see that $d_\tau(\alpha'',\beta'')=d_\tau([2,6,4,5,1,3],[4,6,2,5,3,1])>d_\tau([2,6,4],[4,6,2])=3$. Similarly, if $\alpha=[2,4,3,1]$ and $\beta=[4,2,1,3]$, then we have $d_\tau(\alpha,\beta)=2$, $\alpha'=[2,4,5,3,1]$, $\beta'=[4,2,5,1,3]$, $\alpha''=[2,4,5,3,6,1]$, $\beta''=[4,2,5,1,6,3]$, $s_\alpha=5$, $s_\beta=5$. It is easy to see that $d_\tau(\alpha'',\beta'')=d_\tau([2,4,5,3,6,1],[4,2,5,1,6,3])>d_\tau([3,6,1],[1,6,3])=3$. All the other permutation pairs are in similar cases. (Note that either $s_\alpha=s_\beta=2$, or $s_\alpha=s_\beta=5$.) So for any two permutations at distance two in Fig. 1 (a), after inserting the redundant cells, their distance is at least 3.

So the code has minimum distance at least 3, and can correct one error. To see that the minimum distance of the code is exactly 3, we just need to consider a particular pair of codewords – say $[6,1,2,5,3,4]$ and $[1,2,6,5,4,3]$ – whose distance equals 3. ■