Comp. Appl. Math. (2014) 33:575–589
DOI 10.1007/s40314-013-0082-y

A coding scheme that increases the code rate

R. S. Raja Durai · Meenakshi Devi

Received: 4 May 2013 / Revised: 30 August 2013 / Accepted: 24 September 2013 /
Published online: 10 October 2013

Abstract Codes having higher information rates are desirable, since a higher rate code implies a more efficient use of redundancy than a lower rate code. However, when choosing a code for a particular application, we must also consider the error-correcting capabilities of the code. There is a basic trade-off between code rate and minimum distance. The smaller the code rate, the larger is the minimum distance and vice-versa. This paper proposes a simple coding scheme that can construct a code with higher information rate from an existing code. First, the paper derives a low-rate $\mathcal{C}'(n', k', d')$-code from an existing $\mathcal{C}(n, k, d)$-code, where $\frac{k}{n} \geq \frac{k'}{n'}$. An associated decoding procedure for the newly derived class of low-rate codes is also described. Finally, the proposed coding scheme combines $\mathcal{C}$ and a set of $\mathcal{C}'$s to obtain a $\mathcal{C}''(n'', k'', d'')$-code with $\frac{k''}{n''} \geq \frac{k}{n} \geq \frac{k'}{n'}$. Kronecker product is used as a basic tool in the coding procedure.

Keywords Rank distance codes · Rank metric · Code length · Information rate · Minimum distance · Rate-increasing procedure

Mathematics Subject Classification 94B05 (Linear codes, general)

1 Introduction

A *block* code of length n over an alphabet $\mathcal{A}$ is a subset $\mathcal{C} \subseteq \mathcal{A}^n$. Usually, the *alphabet* $\mathcal{A}$ is a finite field $\mathbb{F}_q$, where q is a power of a prime number. An (n, M, d) code over $\mathbb{F}_q$ is a

Communicated by Eduardo Souza de Cursi.

R. S. Raja Durai
Department of Mathematics, Jaypee University of Information Technology,
Waknaghat 173234, District Solan, Himachal Pradesh, India
e-mail: rsraja.durai@juit.ac.in

M. Devi (✉)
Department of Mathematics, Bahra University, Waknaghat 173234,
District Solan, Himachal Pradesh, India
e-mail: meenakshi_juit@yahoo.co.in

set of M q-ary codewords such that any two codewords are at a Hamming distance $\geq d$. An (n, M, d) code is said to be linear if it is a linear subspace of $\mathbb{F}_q^n$. A linear code consists of q^k codewords of length n with minimum distance d is denoted by (n, k, d). The information rate of the code $\mathcal{C}$ is defined to be the ratio $\mathcal{R} = \frac{\log_q |\mathcal{C}|}{n}$.

Error-correcting codes were first discovered abstractly in 1945 by Shannon (1948) in his seminal paper, where he proved that there exist error-correcting codes that can achieve the channel capacity. An important theme in information theory is that longer block lengths are required to achieve higher rates and some closer to the channel capacity. A substantial research work has been devoted to search for codes that can handle a given noisy channel. Coding theory was born with the work of Hamming, who introduced a family of codes which are the first single error-correcting codes ever invented (Hamming 1950). Since then most established codes have been generalizations of Hamming codes: Golay codes (Golay 1949), Reed–Muller codes (Muller 1954), Reed–Solomon codes (Irving 1960), Bose–Chaudhury–Hocquenhem codes (Bose and Ray-Chaudhuri 1960), Goppa codes, and Rank Distance codes (Gabidulin 1985), to name a few. About 50 years after Shannon's result, turbo codes (Berrou et al. 1993) were discovered. Further study on turbo codes led to the rediscovery of LDPC codes (Gallager 1963), a class of codes introduced by Gallager about three decades earlier. Algebraic structures such as groups, rings and finite fields are most important in the context of coding theory. A detailed mathematical background on finite fields can be found in Lidl and Niederreiter (1986). Algebraic characterizations of abstract structures akin to finite fields and viable code constructions are given in Cazaran and Kelarev (1999) and Kelarev (2002).

The main challenge in the field of algebraic coding theory is to come up with 'good' codes along with the efficient coding algorithms. A 'good' code is a code that has the potential to correct as many errors as possible while using as little redundancy as possible. In fact, these are contradictory goals. A bound that shows the trade-off between minimum distance and information rate is the singleton bound (Richard 1964). Attempts are made for the constructions of *asymptotically good error-correcting codes* (Guruswami and Indyk 2005) with *linear-time* encoding and decoding complexity. Using code concatenation, asymptotically good codes are obtained (Alon et al. 1992). The main objective of this paper is to construct codes with higher information rate from existing codes. The information rates of codes were considered in the monograph (Kelarev 2002, Section 9.1 & 9.2), and also in Kelarev (2004a,b, 2005, 2006, 2007, 2008) and Alfaro and Kelarev (2006). The class of high-rate codes derived in this paper, though non-asymptotic, equipped with relatively simple encoding and decoding techniques.

Considering $\mathbb{F}_q$ as the alphabet set, almost all coding procedures exist to-date conventionally encode q^k k-tuples to construct an (n, k, d)-code. The basic idea behind our approach is as follows. Instead of considering all q^k message vectors, why not consider only those message vectors which are a (Kronecker) multiple of a single vector (termed as *basic message* vector). Generating $m > 1$ such code sets, each corresponding to a *basic message* vector and also making use of these m *basic message* vectors as side information, one can obtain a code (n'', k'', d'') with an increase in the information rate: $\frac{k''}{n''} > \frac{k}{n}$. Based on this observation, we propose a coding scheme that constructs a higher rate code. This paper considers only rank distance codes introduced by Gabidulin (1985). The class of rank distance codes is defined as subsets of an n-dimensional space $\mathbb{F}_{q^N}^n$ of n-vectors over an extension field $\mathbb{F}_{q^N}$, where $n \leq N$. Unlike conventional codes with Hamming metric, rank distance codes are equipped with rank metric. Though, the concept of rank metric was conceptualized by Loo-Keng Hua (1951) as *Arithmetic distance* and by Philippe Delsarte as *q-distance* on the set of bilinear forms (Delsarte 1978), Gabidulin introduced rank distance for vector spaces over extension fields.

The following section presents the basic definitions and notations. To facilitate our objective, the paper first derives, in Sect. 3, a low-rate $\mathcal{C}'(n', k', d')$-code from an existing $\mathcal{C}(n, k, d)$-code with $\frac{k'}{n'} \leq \frac{k}{n} < 1$. The coding scheme proposed in section IV combines $\mathcal{C}(n, k, d)$ and $\mathcal{C}'(n', k', d')$ to obtain a code $\mathcal{C}''(n'', k'', d'')$ having rate higher than that of $\mathcal{C}$ and $\mathcal{C}'$: $\frac{\log_{q^n} |\mathcal{C}''|}{n''} \geq \frac{\log_{q^n} |\mathcal{C}|}{n} \geq \frac{\log_{q^n} |\mathcal{C}'|}{n'}$. An associated decoding algorithm for the newly constructed class of codes is also given. The paper is concluded in Section V.

2 Preliminaries

This section describes some fundamentals of rank distance codes introduced by Gabidulin (1985) and notations used throughout in this paper.

2.1 Rank distance codes

Let V^n be an n-dimensional *vector space* over the field $\mathrm{GF}(q^N)$, where q is a power of a *prime* and $n \leq N$. Assume that $u_1, u_2, \ldots, u_N$ is some fixed basis of the field $\mathrm{GF}(q^N)$, regarded as a *vector space* over $\mathrm{GF}(q)$. Then, $x_i = a_{1i}u_1 + a_{2i}u_2 + \cdots + a_{Ni}u_N$ for any $x_i \in \mathrm{GF}(q^N)$. Let $x = (x_1, x_2, \ldots, x_n) \in V^n$ and A_N^n be the collection of all $N \times n$ matrices over $\mathrm{GF}(q)$. Associated with each $x \in V^n$, the $N \times n$ matrix denoted by $A(x)$ is defined as

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{Nn} \end{pmatrix}$$

Definition 2.1 The rank of a vector $x \in V^n$ over $\mathrm{GF}(q^N)$ is defined as the rank of the matrix $A(x)$ and is denoted by $r(x; q)$. The norm $r(x; q)$ specifies a rank metric on V^n as $d(x, y) = r(x - y; q)$ for all $x, y \in V^n$.

Definition 2.2 A linear (n, k, d) code which is a k-dimensional subspace of V^n is said to be a *rank distance* (RD) code if its metric is induced by the rank norm. An (n, k, d) *RD* code is said to be a *maximum rank distance* (MRD) code if $d = n - k + 1$. Here d is the minimum distance of the code and is defined as the minimum rank any non-zero codeword can have.

Definition 2.3 An (n, k, d) MRD code is generated by the generator matrix $\mathbf{G}$ defined as follows:

$$\mathbf{G} = \begin{pmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^q & g_2^q & \cdots & g_n^q \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}$$

where $g_1, g_2, \ldots, g_n \in \mathrm{GF}(q^N)$ are linearly independent over $\mathrm{GF}(q)$. The paper considers the case when $n = N$.

2.2 Notations and abbreviations

A code of block length n, message length k and minimum distance d consisting of M q^n-ary codewords defined over $\mathrm{GF}(q^n)$ is denoted by $(n, k|M, d)$. Further, for arbitrary vectors

$\mathbf{a} = (a_1, a_2, \ldots, a_{m_1})$ and $\mathbf{b} = (b_1, b_2, \ldots, b_{m_2})$, the concatenation of $\mathbf{a}$ and $\mathbf{b}$ is defined as the vector $(a_1, a_2, \ldots, a_{m_1}, b_1, b_2, \ldots, b_{m_2})$ of length $(m_1 + m_2)$ and abbreviated as $(\mathbf{a}, \mathbf{b})$. Further, whenever to refer two vectors—an m_1-tuple and an m_2-tuple from two distinct sets $\mathcal{A}$ and $\mathcal{B}$ having common terminologies (such as message vector, codeword), we denote the respective vectors by $\mathbf{a}^{(r)}$ and $\mathbf{a}_{(s)}$, using a same alphabet:

$$\begin{aligned} \mathbf{a}^{(r)} &= (a_{r1}, a_{r2}, \ldots, a_{rm_1}) \\ \text{and} \quad \mathbf{a}_{(s)} &= (a_{s1}, a_{s2}, \ldots, a_{sm_2}) \end{aligned}$$

for some distinct positive integers r and s.

Definition (*Kronecker product*) For the above two vectors $\mathbf{a}$ and $\mathbf{b}$ of respective lengths m_1 and m_2, the Kronecker product of $\mathbf{a}$ and $\mathbf{b}$ is defined as the m_1m_2-length vector $(a_1\mathbf{b}, a_2\mathbf{b}, \ldots, a_{m_1}\mathbf{b})$ and denoted by $\mathbf{a} \otimes \mathbf{b}$.

3 Construction of low-rate codes

A *normal* basis of $\mathtt{GF}(q^n)$ over $\mathtt{GF}(q)$ is a basis of the form $\left\{\alpha, \alpha^q, \ldots, \alpha^{q^{n-1}}\right\}$ for some $\alpha \in \mathtt{GF}(q^n)$. Let $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ be a *normal* basis in $\mathtt{GF}(q^n)$ with $\alpha_i = \alpha^{q^i}$ for $0 \le i \le n-1$. Then, $\alpha_i^{q^k} = \alpha_{i+k}$ for an integer k, where indices of α are reduced modulo n. It is known that every $\mathtt{GF}(q^n)$ has a *normal* basis over $\mathtt{GF}(q)$ (Lidl and Niederreiter 1986). Since $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ being a *normal* basis, the set $\{\alpha_1^{[j]}, \alpha_2^{[j]}, \ldots, \alpha_n^{[j]}\}$ also forms a *normal* basis in $\mathtt{GF}(q^n)$ for each $j = 1, 2, \ldots, n$, where here and after $[r] = q^r$ for some positive integer r. Further, let the *normal* basis $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ of $\mathtt{GF}(q^n)$ over $\mathtt{GF}(q)$ be *self-complementary*: a basis $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ of $\mathtt{GF}(q^n)$ over $\mathtt{GF}(q)$ is called *self-complementary* if

$$\mathbf{tr}(\alpha_i \alpha_j) = \begin{cases} 1, & i = j \\ 0, & i \ne j \end{cases}$$

where $\mathbf{tr}$ is the *absolute trace* from $\mathtt{GF}(q^n)$ to $\mathtt{GF}(q)$ defined as $\mathbf{tr}(\alpha) = \sum_{i=0}^{n-1} \alpha^{q^i}$. A criteria for the existence of *self-complementary bases* are obtained in Lempel and Weinberger (1988), where it has been established that $\mathtt{GF}(q^n)$ has a *self-complementary normal* basis if and only if n is odd or $n \equiv 2 \pmod 4$ and q is even.

Consider an $(n, k|q^{nk}, d)$ MRD code $\mathcal{C}$ over $\mathtt{GF}(q^n)$ with its generator matrix $\mathbf{G} = [g_i^{q^j}]_{i,j=0}^{n,k-1}$ such that the basis $\{g_1, g_2, \ldots, g_n\}$ is *self-complementary* in $\mathtt{GF}(q^n)$. Let $\mathbf{G} = [\mathcal{I}_k : \mathcal{P}]$, where $\mathcal{I}_k$ denotes the $k \times k$ identity matrix and $\mathcal{P}$ is some $k \times (n-k)$ matrix. Then, $\mathbf{H} = [-\mathcal{P}^{\mathbf{T}} : \mathcal{I}_{n-k}]$ is the parity-check matrix. A procedure for the systematic encoding of MRD codes can be found in (Vasantha Kandasamy, 2012, Section 4.3). Assume that the code $\mathcal{C}$ is equipped with a rank-error correcting decoding algorithm that can correct up to $\lfloor \frac{d-1}{2} \rfloor$ rank errors. The use of *self-complementary basis* as the row elements in the generator matrix $\mathbf{G}$ plays a crucial role in the decoding of high-rate codes discussed in the next section.

3.1 Encoding the low-rate (*secondary*) codes

For an integer $k^o > 0$, choose a k^o-tuple $\mathbf{m}^{(0)} = (m_{01}, m_{02}, \ldots, m_{0k^o}) \in \left[\mathtt{GF}(q^n)\right]^{k^o}$ such that $\mathbf{m}^{(0)} \ne (\mathbf{0})$. Assume that the k^o-tuple $\mathbf{m}^{(0)}$ is known to the encoder and decoder. Call this fixed vector as the *basic message vector*.

For each $i = 1, 2, \ldots, q^{nk}$, let $\mathbf{m}_{(i)}$ denote the k-tuple $(m_{i1},\ m_{i2}, \ldots,\ m_{ik}) \in \left[\mathrm{GF}(q^n)\right]^k$. Using these message vectors of $\mathcal{C}$ and the *basic message vector* $\mathbf{m}^{(0)}$, generate a new set of message vectors each of length kk^o as follows:

$$\begin{aligned}
\mathbf{m}^{(1)} &= \mathbf{m}^{(0)} \otimes \mathbf{m}_{(1)} \\
&= \left(m_{01}\mathbf{m}_{(1)},\ m_{02}\mathbf{m}_{(1)}, \ldots,\ m_{0k^o}\mathbf{m}_{(1)}\right) \\
\mathbf{m}^{(2)} &= \mathbf{m}^{(0)} \otimes \mathbf{m}_{(2)} \\
&= \left(m_{01}\mathbf{m}_{(2)},\ m_{02}\mathbf{m}_{(2)}, \ldots,\ m_{0k^o}\mathbf{m}_{(2)}\right) \\
&\vdots \\
\mathbf{m}^{(q^{nk})} &= \mathbf{m}^{(0)} \otimes \mathbf{m}_{(q^{nk})} \\
&= \left(m_{01}\mathbf{m}_{(q^{nk})},\ m_{02}\mathbf{m}_{(q^{nk})}, \ldots,\ m_{0k^o}\mathbf{m}_{(q^{nk})}\right).
\end{aligned}$$

Considering these q^{nk} newly generated kk^o-tuples as the actual message vectors to be transmitted, the construction procedure attempts to channel encode these kk^o-tuple message vectors using the $k \times n$ generator matrix $\mathbf{G} = [\mathcal{I}_k : \mathcal{P}]$ as follows. Since $\mathbf{m}^{(i)}$ is a kk^o-tuple, to encode it using the $k \times n$ matrix $\mathbf{G}$, select a k-component $\mathbf{m}'_{(i)}$ from $\mathbf{m}^{(i)}$: without loss of generality, $\mathbf{m}'_{(i)} = m_{01}\mathbf{m}_{(i)}$ for each $i = 1, 2, \ldots, q^{nk}$. Since $\mathbf{m}'_{(i)} \in \left[\mathrm{GF}(q^n)\right]^k$ being a message vector of $\mathcal{C}$, one obtains an n-tuple $\mathbf{m}'_{(i)}\mathbf{G} = \mathbf{c}_{(i)}$ for each i. Since $\mathbf{G}$ is assumed to be in standard form, each n-tuple $\mathbf{c}_{(i)}$ has the well-known representation—the first k components are the message symbols and remaining $n - k$ components are the parity-check symbols: $\mathbf{c}_{(i)} = \left(m_{01}\mathbf{m}_{(i)},\ m_{01}\mathbf{m}_{(i)}\mathcal{P}\right)$ for each $i = 1, 2, \ldots, q^{nk}$. Clearly, the codeword $\mathbf{c}_{(i)} \in \mathcal{C}$ is associated with $\mathbf{m}'_{(i)}$.

The newly generated message vectors are then encoded by appending the parity-check symbols associated with the selected k-component message vectors of $\mathcal{C}$: each of the newly generated kk^o-tuple message vector $\mathbf{m}^{(i)}$ is encoded by appending the $(n-k)$-tuple parity-check-symbol of $\mathbf{c}_{(i)}$ to it to form $\mathbf{c}^{(i)} = \left(\mathbf{m}^{(0)} \otimes \mathbf{m}_{(i)},\ m_{01}\mathbf{m}_{(i)}\mathcal{P}\right)$:

$$\begin{aligned}
\mathbf{c}^{(1)} &= \left(\mathbf{m}^{(0)} \otimes \mathbf{m}_{(1)},\ \mathbf{p}_{(1)}\right) \\
\mathbf{c}^{(2)} &= \left(\mathbf{m}^{(0)} \otimes \mathbf{m}_{(2)},\ \mathbf{p}_{(2)}\right) \\
&\vdots \\
\mathbf{c}^{(q^{nk})} &= \left(\mathbf{m}^{(0)} \otimes \mathbf{m}_{(q^{nk})},\ \mathbf{p}_{(q^{nk})}\right),
\end{aligned}$$

where $\mathbf{p}_{(i)} = m_{01}\mathbf{m}_{(i)}\mathcal{P}$.

These newly defined codewords constitute an $(n + k(k^o - 1),\ kk^o | q^{nk},\ d')$-code over $\mathrm{GF}(q^n)$, say $\mathcal{C}'$. Note that the code $\mathcal{C}'$ thus obtained is not an MRD code and consequently the minimum distance of the code can be found in the following sense. Rank distance between two codewords is at most the Hamming distance between them: if d' denotes the Hamming distance, then for all $c_1, c_2 \in \mathcal{C}$, the rank distance satisfies the inequality $d(c_1, c_2) \leq d'(c_1, c_2)$ (Gabidulin 1985). The minimum distance d' of the derived code $\mathcal{C}'$ can be calculated as follows. By the very construction of $\mathcal{C}'$, for an arbitrary codeword $\mathbf{c}^{(i)} \in \mathcal{C}'$, the kk^o-tuple message vector can be written in terms of the Kronecker product of $\mathbf{m}_{(i)} \in \left[\mathrm{GF}(q^n)\right]^k$ and $\mathbf{m}^{(0)} \in \left[\mathrm{GF}(q^n)\right]^{k^o}$: $\mathbf{c}_{(i)} = \left(\mathbf{m}^{(0)} \otimes \mathbf{m}_{(i)},\ m_{01}\mathbf{m}_{(i)}\mathcal{P}\right)$. Consequently, the *Hamming* weight of a non-zero codeword $\mathbf{c}^{(i)} \in \mathcal{C}'$ can be calculated as follows:

$$\begin{aligned} w(\mathbf{c}^{(i)}) &= w\left(\mathbf{m}^{(i)},\ \mathbf{p}_{(i)}\right) \\ &= w\left(\mathbf{m}^{(0)} \otimes \mathbf{m}_{(i)}\right) + w(\mathbf{p}_{(i)}) \\ &= w\left(\mathbf{m}^{(0)}\right) w\left(\mathbf{m}_{(i)}\right) + w(\mathbf{p}_{(i)}) \\ &= k^o w(\mathbf{m}_{(i)}) + w(\mathbf{p}_{(i)}) \\ &= k^o w(\mathbf{m}_{(i)}) + w(\mathbf{c}_{(\mathbf{i})}) - w(\mathbf{m}_{(\mathbf{i})}) \\ &\geq k^o w(\mathbf{m}_{(i)}) + d - w(\mathbf{m}_{(i)}) \\ &= d + w(\mathbf{m}_{(i)})(k^o - 1) \\ &\geq d + k^o - 1 \end{aligned}$$

where $1 \leq w(\mathbf{m}_{(i)}) \leq k$. It follows that, the minimum distance of $\mathcal{C}'$ is $d + k^o - 1$. It is easy to see that $d' \geq d$. The derived code can be described by its *systematic* encoding map as follows. While a q^n-ary $(n, k|q^{nk}, d)$-code $\mathcal{C}$ is specified by an injective map $\mathcal{E} : \left[\mathtt{GF}(q^n)\right]^k \to \left[\mathtt{GF}(q^n)\right]^n$ from the q^n-ary strings of length k to q^n-ary strings of length n, which is a linear transformation $\mathbf{x}_{(0)} \mapsto (\mathbf{x}_{(0)},\ \mathbf{x}_{(0)}\mathcal{P})$, the derived code $\mathcal{C}'$ has an associated encoding map $\mathcal{E}' : \left[\mathtt{GF}(q^n)\right]^{kk^o} \to \left[\mathtt{GF}(q^n)\right]^{kk^o+n-k}$ given by $\mathbf{m}^{(0)} \otimes \mathbf{x}_{(0)} \mapsto (\mathbf{m}^{(0)} \otimes \mathbf{x}_{(0)},\ \mathbf{x}_{(0)}\mathcal{P})$. Clearly, $\mathcal{C} \subseteq \left[\mathtt{GF}(q^n)\right]^n$ and $\mathcal{C}' \subset \left[\mathtt{GF}(q^n)\right]^{kk^o+n-k}$ with $|\mathcal{C}| = |\mathcal{C}'| = q^{nk}$.

As the code $\mathcal{C}'$ is derived from $\mathcal{C}$, we adopt the following terminology: call $\mathcal{C}$ as the *primary* code and the code $\mathcal{C}'$ that is derived from $\mathcal{C}$ as the *secondary* code. Observe that, the *secondary* code $\mathcal{C}'(n',\ k'|M,\ d')$ satisfies the singleton bound, $n' = n + k(k^o - 1)$, $k' = kk^o$, $M = q^{nk}$ and $d' = d + k^o - 1$:

$$\begin{aligned} q^{n'-d'+1} &= q^{[n+k(k^o-1)]-(d+k^o-1)+1} \\ &= q^{(n-k+1)+(kk^o)-(d+k^o-1)} \\ &= \left(\frac{q^{n-k+1}}{q^d}\right) q^{kk^o-(k^o-1)} \\ &= q^{kk^o-(k^o-1)} \\ &= q^{k^o(k-1)+1} \\ &\geq q^{(k-1)+1} \qquad \text{for all } k^o > 0 \\ &= q^{nk}. \end{aligned}$$

The block length, message length and minimum distance of $\mathcal{C}'$ are given by $n' = n + k(k^o - 1)$, $k' = kk^o$, and $d' = d + k^o - 1$, respectively. Note that, each message vector of the *secondary* code is a Kronecker product of the message vectors of the *primary* code with the *basic message vector* $\mathbf{m}^{(0)}$. As every kk^o-tuple message vector contains the k^o-tuple $\mathbf{m}^{(0)}$, the *basic message vector* known at the receiver-end will act as a side-information for the decoder in recovering a received sequence. The decoding technique for the *secondary* code described in the next sub-section is based on the decoding technique of the *primary* code.

3.2 Decoding the *secondary* codes

Let $\mathbf{m}^{(i)} = \mathbf{m}^{(0)} \otimes \mathbf{m}_{(i)}$ be the (actual) message vector of length kk^o to be conveyed to the receiver. Suppose that $\mathbf{c}^{(i)} = \left(\mathbf{m}^{(i)},\ \mathbf{p}_{(i)}\right) \in \mathcal{C}'(n + k(k^o - 1),\ kk^o|q^{nk},\ d + k^o - 1)$, the codeword associated with the message vector $\mathbf{m}^{(i)}$ is transmitted. Let $\mathbf{r}^{(i)} = \left(\mathbf{r}_m^{(i)},\ \mathbf{r}_{(i)}^{\mathtt{p}}\right)$ be

the received vector, where $\mathbf{r}_m^{(i)} = \mathbf{m}^{(i)} + \mathbf{e}^{(i)}$ and $\mathbf{r}_{(i)}^{\mathrm{p}} = \mathbf{p}_{(i)} + \mathbf{e}_{(i)}$ for some error vectors $\mathbf{e}^{(i)} \in \left[\mathtt{GF}(q^n)\right]^{kk^o}$ and $\mathbf{e}_{(i)} \in \left[\mathtt{GF}(q^n)\right]^{n-k}$.

Since $\mathbf{m}^{(i)} = \mathbf{m}^{(0)} \otimes \mathbf{m}_{(i)}$, the associated received message vector can be written as $\mathbf{r}_m^{(i)} = \mathbf{m}^{(0)} \otimes \mathbf{m}_{(i)} + \mathbf{e}^{(i)} = (m_{01}\mathbf{m}_{(i)},\ m_{02}\mathbf{m}_{(i)}, \ldots,\ m_{0k^o}\mathbf{m}_{(i)}) + \mathbf{e}^{(i)}$, where $\mathbf{e}^{(i)} = \left(\mathbf{e}_1^{(i)},\ \mathbf{e}_2^{(i)}, \ldots,\ \mathbf{e}_{k^o}^{(i)}\right)$ with $\mathbf{e}_1^{(i)}, \mathbf{e}_2^{(i)}, \ldots, \mathbf{e}_{k^o}^{(i)} \in \left[\mathtt{GF}(q^n)\right]^k$. On receiving the $(kk^o + n - k)$-tuple $\mathbf{r}^{(i)}$, the decoder extracts the k-symbol vector $\mathbf{r}_{(i)}^m = m_{01}\mathbf{m}_{(i)} + \mathbf{e}_1^{(i)}$ from $\mathbf{r}_m^{(i)}$ and forms the n-tuple $\mathbf{r}_{(i)}$ as follows: $\mathbf{r}_{(i)} = \left(\mathbf{r}_{(i)}^m,\ \mathbf{r}_{(i)}^{\mathrm{p}}\right)$. If $\mathbf{r}_{(i)}$ has $e \leq \lfloor \frac{d-1}{2} \rfloor$ errors, on employing the error-correcting algorithm associated with the *primary* code $\mathcal{C}$ to $\mathbf{r}_{(i)}$, decoder recovers $m_{01}\mathbf{m}_{(i)}$. Since the *basic message vector* $\mathbf{m}^{(0)}$ is known at the receiver-end, upon dividing the k-tuple $m_{01}\mathbf{m}_{(i)}$ by m_{01}, decoder recovers the k-tuple $\mathbf{m}_{(i)}$. The decoder then readily obtains the original transmitted message vector as $\mathbf{m}^{(i)} = \mathbf{m}^{(0)} \otimes \mathbf{m}_{(i)}$.

Although the message vectors of *secondary* code are of length kk^o, the *secondary* code has the lower code rate than the *primary* code. However, an increase in the information rate can be made by not transmitting the *basic message vector* $\mathbf{m}^{(0)}$, as it is also known at the receiver-end. A coding technique proposed in the next section accomplishes this.

4 Construction of high-rate codes

Although the transmitted kk^o-tuple message vector is recovered successfully, the *secondary* code $\mathcal{C}'(n', k', d')$ itself is not useful as the code rate is decreasing for increasing values of k^o: $\frac{k}{n'} < \frac{k}{n}$ for all $k^o > 1$ and $\frac{k}{n'}$ approaches 0 as k^o increases. This is because of the obvious reason that the Kronecker product merely increases the length of each message vector (consequently, the codewords) but not the number of codewords of $\mathcal{C}'$. In the following, this section proposes a coding technique to increase the number of codewords to obtain a code with higher information rate than *primary* and *secondary* codes.

4.1 Encoding the high-rate (*tertiary*) codes

Consider the *primary* code $\mathcal{C}$ and *secondary* code $\mathcal{C}'$. Let the $k \times n$ generator and $(n-k) \times n$ parity-check matrices of $\mathcal{C}$ be such that $\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2\ \ldots\ \mathbf{g}_k]^{\mathbf{T}}$ and $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2\ \ldots\ \mathbf{h}_{n-k}]^{\mathbf{T}}$ for some $\mathbf{g}_1, \ldots, \mathbf{g}_k, \mathbf{h}_1, \ldots, \mathbf{h}_{n-k} \in \left[\mathtt{GF}(q^n)\right]^n$. Consider the following non-zero vectors:

$$\boldsymbol{\beta}^{(1)} = \underbrace{(\beta_1,\ \beta_1, \ldots,\ \beta_1)}_{n \text{ components}}$$

$$\boldsymbol{\beta}^{(2)} = \underbrace{(\beta_2,\ \beta_2, \ldots,\ \beta_2)}_{n \text{ components}}$$

$$\vdots$$

$$\boldsymbol{\beta}^{(q^n-1)} = \underbrace{\left(\beta_{q^n-1},\ \beta_{q^n-1}, \ldots,\ \beta_{q^n-1}\right)}_{n \text{ components}}$$

where $\beta_1, \beta_2, \ldots, \beta_{q^n-1}$ are the non-zero elements of $\mathtt{GF}(q^n)$.

Recall that the *secondary* code $\mathcal{C}'(n + k(k^o - 1),\ kk^o | q^{nk},\ d + k^o - 1)$ obtained is in fact generated from the *primary* code $\mathcal{C}(n, k | q^{nk}, d)$ by considering the Kronecker product of $\mathbf{m}^{(0)} = (m_{01}, m_{02}, \ldots, m_{0k^o})$ with each of its message vectors. In this way, the *secondary*

code $\mathcal{C}'$ is said to be generated by the *basic message vector* $\mathbf{m}^{(0)}$. Considering $\boldsymbol{\beta}^{(j)}$ as the *basic message vector*, let $\mathcal{C}'_j(n+k(n-1),\ kn|q^{nk},\ d+n-1)$ denote the jth *secondary* code generated by $\boldsymbol{\beta}^{(j)}$, $j = 1, 2, \ldots, q^n - 1$. Then, for each j, the codewords of $\mathcal{C}'_j$ are given by $\mathbf{c}_j^{(i)} = \left(\boldsymbol{\beta}^{(j)} \otimes \mathbf{m}_{(i)},\ \beta_j \mathbf{m}_{(i)} \mathcal{P}\right)$, $i = 1, 2, \ldots, q^{nk}$. Note that for $k^o = n$ and $\mathbf{m}^{(0)} = \boldsymbol{\beta}^{(1)}$, we have $\mathcal{C}' = \mathcal{C}'_1$ generated by $\boldsymbol{\beta}^{(1)}$. Clearly, $|\mathcal{C}'_j| = q^{nk}$ for each $j = 1, 2, \ldots, q^n - 1$. However, $|\mathcal{C}'_1 \cup \mathcal{C}'_2 \cup \cdots \cup \mathcal{C}'_{q^n-1}| = (q^n - 1)q^{nk} - (q^n - 2)$; excluding the common codeword appearing more than once.

By convenient abuse of notation, we use the same symbol $\mathcal{C}'$ to represent $\mathcal{C}'_1 \cup \mathcal{C}'_2 \cup \cdots \cup \mathcal{C}'_{q^n-1}$ and call it as the *secondary code* derived from $\mathcal{C}$: $\mathcal{C}' = \mathcal{C}'_1 \cup \mathcal{C}'_2 \cup \cdots \cup \mathcal{C}'_{q^n-1}$. It is important to note that the kn-length message vectors under consideration for transmission are from the $q^n - 1$ *secondary* codes $\mathcal{C}'_1, \mathcal{C}'_2, \ldots, \mathcal{C}'_{q^n-1}$ and the associated $(n-k)$-length parity-check vectors are from the *primary* code $\mathcal{C}$. Consider the kn-tuple (ith message vector) $\mathbf{m}_j^{(i)} = \boldsymbol{\beta}^{(j)} \otimes \mathbf{m}_{(i)}$ (of jth *secondary* code $\mathcal{C}'_j$) that is to be conveyed to the receiver, where $i = 1, 2, \ldots, q^{nk}$ and $j = 1, 2, \ldots, q^n - 1$. For each j, since $\boldsymbol{\beta}^{(j)}$ is also known at the receiver-end, instead of transmitting the $[n + k(n-1)]$-tuple:

$$\mathbf{c}_j^{(i)} = \left(\boldsymbol{\beta}^{(j)} \otimes \mathbf{m}_{(i)},\ \beta_j \mathbf{m}_{(i)} \mathcal{P}\right) \in \mathcal{C}'_j(n + k(n-1), \quad kn|q^{nk},\ d + n - 1)$$

as done in the last section, the transmitter can send only the n-tuple $\mathbf{c}_{(i)}^{j} = \left(\beta_j \mathbf{m}_{(i)},\ \beta_j \mathbf{m}_{(i)} \mathcal{P}\right)$ $\in \mathcal{C}(n,\ k|q^{nk},\ d)$, which is in turn associated with the k-tuple message vector $\mathbf{m}_{(i)}^{j} = \beta_j \mathbf{m}_{(i)}$. To our surprise, observe that, this n-tuple is from the *primary* code $\mathcal{C}(n, k|q^{nk}, d)$. The problem for the receiver is to retrieve first the corresponding jth *basic message vector* $\boldsymbol{\beta}^{(j)}$, which would help the decoder to identify the jth *secondary* code $\mathcal{C}'_j$ from which the codeword was transmitted. In this way, to convey a kn-length message vector, the transmitter is actually sending only the n-length vector $\mathbf{c}_{(i)}^{j} \in \mathcal{C}$, not the $[n + k(n-1)]$-length vector $\mathbf{c}_j^{(i)} \in \mathcal{C}'_j \subseteq \mathcal{C}'$. Consequently, one obtains the code $\mathcal{C}''$ with parameters $(n, kn|(q^n - 1)q^{nk}, d)$, call it as *tertiary* code, as it is obtained from the *primary* and *secondary* codes. By the very encoding of the *tertiary* code $\mathcal{C}''$, the code can be specified by the mapping $\mathcal{E}'' : \mathcal{A}_{q^n}^{kn} \to \left[\mathtt{GF}(q^n)\right]^n$ given by $\boldsymbol{\beta}^{(j)} \otimes \mathbf{x}_{(0)} \mapsto (\beta_j \mathbf{x}_{(0)},\ \beta_j \mathbf{x}_{(0)} \mathcal{P})$, where $\mathcal{A}_{q^n}^{kn} = \left\{\boldsymbol{\beta}^{(j)} \otimes \mathbf{x}_{(0)} \ \mid\ \mathbf{x}_{(0)} \in \left[\mathtt{GF}(q^n)\right]^k,\ \ j = 1, 2, \ldots, q^n - 1\right\}$.

Example 1 (Construction of *tertiary* codes) Consider the *primary* code $\mathcal{C}(5, 3|2^{15}, 3)$, an MRD code, with the generator and parity-check matrices:

$$\mathbf{G} = \begin{bmatrix} \alpha^{18} & \alpha^5 & \alpha^{10} & \alpha^{20} & \alpha^9 \\ \alpha^5 & \alpha^{10} & \alpha^{20} & \alpha^9 & \alpha^{18} \\ \alpha^{10} & \alpha^{20} & \alpha^9 & \alpha^{18} & \alpha^5 \end{bmatrix} \quad \text{and} \quad \mathbf{H} = \begin{bmatrix} \alpha^{20} & \alpha^9 & \alpha^{18} & \alpha^5 & \alpha^{10} \\ \alpha^9 & \alpha^{18} & \alpha^5 & \alpha^{10} & \alpha^{20} \end{bmatrix},$$

where α is a primitive element of $\mathtt{GF}(2^5)$ such that $\alpha^5 = \alpha^2 + 1$. Consider the following non-zero vectors:

$$\begin{aligned} \boldsymbol{\beta}^{(1)} &= (1,\ 1,\ 1,\ 1,\ 1) \\ \boldsymbol{\beta}^{(2)} &= (\alpha,\ \alpha,\ \alpha,\ \alpha,\ \alpha) \\ &\vdots \\ \boldsymbol{\beta}^{(31)} &= \left(\alpha^{30},\ \alpha^{30},\ \alpha^{30},\ \alpha^{30},\ \alpha^{30}\right), \end{aligned}$$

where $1, \alpha, \ldots, \alpha^{30} \in \mathrm{GF}(2^5)$. For each $j = 1, 2, \ldots, 31$, using *basic message vector* $\boldsymbol{\beta}^{(j)}$, the *secondary* code $\mathcal{C}'_j$ can be generated from the *primary code* $\mathcal{C}(5, 3|2^{15}, 3)$ by taking the Kronecker product of $\boldsymbol{\beta}^{(j)}$ with each of the message vector $\mathbf{m}_{(i)} \in \left[\mathrm{GF}(2^5)\right]^3$ of $\mathcal{C}$. For instance, construction of the *secondary* code $\mathcal{C}'_2$ using $\boldsymbol{\beta}^{(2)} = (\alpha,\ \alpha,\ \alpha,\ \alpha,\ \alpha)$ is done in what follows. Considering Kronecker product of $\boldsymbol{\beta}^{(2)}$ with the ith message vector $\mathbf{m}_{(i)} = (\alpha, \alpha^2, \alpha^3)$ of the *primary* code $\mathcal{C}$, we construct the second *secondary* code $\mathcal{C}'_2$ as done below:

$$\text{let } \mathbf{m}^{(i)} = \boldsymbol{\beta}^{(2)} \otimes \mathbf{m}_{(i)} = (\alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4)$$

be the message vector to be transmitted. To channel encode this 15-tuple message vector $\mathbf{m}^{(i)}$, we select 3-component $\mathbf{m}'_{(i)} = (\alpha^2, \alpha^3, \alpha^4)$ from $\mathbf{m}^{(i)}$ and encode conventionally with the 3×5 generator matrix in systematic form:

$$\mathbf{G} = [\mathcal{I}_3 \ :\ \mathcal{P}] \ = \ \begin{bmatrix} 1 & 0 & 0 & \alpha^{22} & \alpha \\ 0 & 1 & 0 & \alpha^{24} & \alpha^7 \\ 0 & 0 & 1 & \alpha^5 & \alpha^{10} \end{bmatrix}.$$

Since $\mathbf{m}'_{(i)} \in \left[\mathrm{GF}(2^5)\right]^3$ being a message vector of $\mathcal{C}$, one obtains the 5-tuple $\mathbf{m}'_{(i)}\mathbf{G} = \mathbf{c}_{(i)} = (\alpha^2, \alpha^3, \alpha^4, \alpha^{23}, \alpha^{11})$. Each of the newly generated 15-tuple message vector $\mathbf{m}^{(i)}$ is then encoded by simply appending $(n - k = 2)$-tuple parity-check symbol of $\mathbf{c}_{(i)}$ to it to form

$$\begin{aligned} \mathbf{c}_2^{(i)} &= (\boldsymbol{\beta}^{(2)} \otimes \mathbf{m}_{(i)},\ \beta_2 \mathbf{m}_{(i)} \mathcal{P}) \\ &= (\alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^{23}, \alpha^{11}). \end{aligned}$$

In this way, the jth *secondary* code $\mathcal{C}'_j(17, 15|2^{15}, 7)$ is generated by the *basic message vector* $\boldsymbol{\beta}^{(j)}$ for $j = 1, 2, \ldots, 31$. Let $\mathcal{C}' = \mathcal{C}'_1 \cup \mathcal{C}'_2 \cup \cdots \cup \mathcal{C}'_{31}$. As $\boldsymbol{\beta}^{(2)}$ is known to the receiver, instead of transmitting the 15-tuple $\mathbf{c}_2^{(i)} = (\alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^{23}, \alpha^{11}) \in \mathcal{C}'_2 \subseteq \mathcal{C}'$, transmitter can send only 5-tuple $\mathbf{c}_{(i)}^2 = (\alpha^2, \alpha^3, \alpha^4, \alpha^{23}, \alpha^{11}) \in \mathcal{C}(5, 3|2^{15}, 3)$, which is the codeword associated with the 3-tuple message vector $\mathbf{m}_{(i)}^2 = \beta_2 \mathbf{m}_{(i)} = (\alpha^2, \alpha^3, \alpha^4)$. This way of usage of codewords of the *secondary* code $\mathcal{C}'$ results in the *tertiary* code $\mathcal{C}''$ with parameters $(5, 15|31 \cdot 2^{15}, 3)$.

4.2 Decoding the *tertiary* codes

Recovering the jth *basic message vector* $\boldsymbol{\beta}^{(j)}$ from a received n-tuple $\mathbf{r}_{(i)}^j$ (say) is not straightforward. Even if the decoder recovers $\beta_j \mathbf{m}_{(i)}$ from the transmitted $\mathbf{c}_{(i)}^j$, determining either of the unknowns $\beta_j \in \mathrm{GF}(q^n)$ or $\mathbf{m}_{(i)} \in \left[\mathrm{GF}(q^n)\right]^k$ is not possible as their product $\beta_j \mathbf{m}_{(i)}$ is not unique. However, retrieving β_j is mandatory to determine the actual message vector $\mathbf{m}^{(i)} = \boldsymbol{\beta}^{(j)} \otimes \mathbf{m}_{(i)}$. To overcome this situation, instead of transmitting the n-tuple $\mathbf{c}_{(i)}^j \in \mathcal{C}$, the transmitter transmits the n-tuple $\mathbf{c}'_{(i)} = \mathbf{c}_{(i)}^j + \beta_j(\mathbf{1})$, where here and after $(\mathbf{1})$ denotes the all 1s n-tuple $\underbrace{(1, 1, \ldots, 1)}_{\text{n components}}$.

The addition of n-tuple $\beta_j(\mathbf{1})$ (of unit rank) would enable the decoder to identify the associated message vector $\mathbf{m}^{(i)} = \boldsymbol{\beta}^{(j)} \otimes \mathbf{m}_{(i)}$, uniquely. However, upon receiving, the

receiver would treat the added n-tuple $\beta_j(\mathbf{1})$ as an error introduced by the channel in addition to the actual channel error.

4.2.1 Decoding the combined error

Let $\mathbf{r}_{(i)}^j = \mathbf{c}_{(i)}^j + \mathbf{e}_{(i)}^j$ be the received vector, where $\mathbf{e}_{(i)}^j = \beta_j(\mathbf{1}) + \mathbf{e}_{(i)}$ with $\mathbf{e}_{(i)} = (e_1, e_2, \ldots, e_n) \in \left[\mathtt{GF}(q^n)\right]^n$ being the error-vector due to channel noise. By employing the rank-error correcting algorithm associated with the *primary* code $\mathcal{C}$, the k-length vector $\mathbf{m}_{(i)}^j = \beta_j \mathbf{m}_{(i)}$ can be recovered provided the combined error $\mathbf{e}_{(i)}^j$ has $e \leq \lfloor \frac{d-1}{2} \rfloor$ rank errors. Upon recovering the k-tuple $\mathbf{m}_{(i)}^j = \beta_j \mathbf{m}_{(i)}$ and the error-vector $\mathbf{e}_{(i)}^j = \mathbf{e}_{(i)} + \beta_j(\mathbf{1})$, the decoder then attempts to determine the actual message vector $\mathbf{m}_j^{(i)} = \boldsymbol{\beta}^{(j)} \otimes \mathbf{m}_{(i)}$ originally transmitted—it remains now for the decoder to retrieve $\beta_j(\mathbf{1})$ from $\mathbf{e}_{(i)}^j = \mathbf{e}_{(i)} + \beta_j(\mathbf{1})$. To accomplish this task, we device a simple procedure to obtain the channel error $\mathbf{e}_{(i)}$ from $\mathbf{e}_{(i)}^j = \mathbf{e}_{(i)} + \beta_j(\mathbf{1})$ using the $(n-k)$-tuple $\mathbf{r}_{(i)}^j \mathbf{H}^\mathbf{T}$ and k-tuple $\mathbf{r}_{(i)}^j \mathbf{G}^\mathbf{T}$, respectively termed as $\mathbf{H}$-syndrome and $\mathbf{G}$-syndrome, which is outlined in the next sub-section.

4.2.2 Decoding the channel error

Consider the $\mathbf{H}$-syndrome of the received n-tuple $\mathbf{r}_{(i)}^j$:

$$\begin{aligned} \mathtt{S}_\mathbf{H} &= \mathbf{r}_{(i)}^j \mathbf{H}^\mathbf{T} \\ &= \left(\mathbf{c}_{(i)}^j + \beta_j(\mathbf{1}) + \mathbf{e}_{(i)}\right) \mathbf{H}^\mathbf{T} \\ &= (s_1, s_2, \ldots, s_{n-k}), \end{aligned} \tag{1}$$

where

$$\begin{aligned} s_1 &= \beta_j + \mathbf{e}_{(i)} \mathbf{h}_1^\mathbf{T} \\ s_2 &= \beta_j + \mathbf{e}_{(i)} \mathbf{h}_2^\mathbf{T} \\ &\vdots \\ \text{and } s_{n-k} &= \beta_j + \mathbf{e}_{(i)} \mathbf{h}_{n-k}^\mathbf{T} \end{aligned}$$

are the elements from the field $\mathtt{GF}(q^n)$. Further, the $\mathbf{G}$-syndrome of the received vector $\mathbf{r}_{(i)}^j$ is given by,

$$\begin{aligned} \mathtt{S}_\mathbf{G} &= \mathbf{r}_{(i)}^j \mathbf{G}^\mathbf{T} \\ &= \left(\mathbf{c}_{(i)}^j + \beta_j(\mathbf{1}) + \mathbf{e}_{(i)}\right) \mathbf{G}^\mathbf{T} \\ &= \left(\beta_j \mathbf{m}_{(i)} \mathbf{G} + \beta_j(\mathbf{1}) + \mathbf{e}_{(i)}\right) \mathbf{G}^\mathbf{T} \\ &= \beta_j \mathbf{m}_{(i)} + \left(\beta_j(\mathbf{1}) + \mathbf{e}_{(i)}\right) \mathbf{G}^\mathbf{T} \\ &= (s_{n-k+1}, s_{n-k+2}, \ldots, s_n), \end{aligned}$$

where

$$\begin{aligned} s_{n-k+1} &= \beta_j m_1 + \beta_j + \mathbf{e}_{(i)} \mathbf{g}_1^\mathbf{T} \\ s_{n-k+2} &= \beta_j m_2 + \beta_j + \mathbf{e}_{(i)} \mathbf{g}_2^\mathbf{T} \end{aligned}$$

$$\begin{aligned} &\vdots \\ \text{and} \qquad & s_n = \beta_j m_k + \beta_j + \mathbf{e}_{(i)}\mathbf{g}_k^{\mathbf{T}} \end{aligned}$$

are the elements from the field $\mathtt{GF}(q^n)$ and $\mathbf{G}\mathbf{G}^{\mathbf{T}} = \mathbf{I}$—an identity matrix—as $\mathbf{G} = [g_i^{q^j}]_{i,j=0}^{n,k-1}$ is generated by the *self-complementary basis*. As $\mathbf{m}_{(i)}^j = \beta_j \mathbf{m}_{(i)}$ is retrieved already, subtracting it from the $\mathbf{G}$-syndrome above, one obtains the following k components of $\mathbf{r}_{(i)}^j \mathbf{G}^{\mathbf{T}} - \mathbf{m}_{(i)}^j$:

$$\begin{aligned} & \text{let} \quad \mathbb{S}'_{\mathbf{G}} = \mathbf{r}_{(i)}^j \mathbf{G}^{\mathbf{T}} - \mathbf{m}_{(i)}^j \\ \Rightarrow \quad & s'_{n-k+1} = \beta_j + \mathbf{e}_{(i)}\mathbf{g}_1^{\mathbf{T}} \\ & s'_{n-k+2} = \beta_j + \mathbf{e}_{(i)}\mathbf{g}_2^{\mathbf{T}} \\ & \qquad \vdots \\ \text{and} \qquad & s'_n = \beta_j + \mathbf{e}_{(i)}\mathbf{g}_k^{\mathbf{T}}. \end{aligned} \tag{2}$$

Observe the presence of the unknown $\beta_j = \beta_j(\mathbf{1})\mathbf{H}^{\mathbf{T}} = \beta_j(\mathbf{1})\mathbf{G}^{\mathbf{T}}$ in all the n components of the equations (1) and (2). Systematically, β_j can be eliminated from these n components to obtain the following n useful quantities:

$$\begin{aligned} \text{let} \quad r_1 &= s_1 - s_2 \\ r_2 &= s_2 - s_3 \\ &\vdots \\ r_{n-k} &= s_{n-k} - s'_{n-k+1} \\ r_{n-k+1} &= s'_{n-k+1} - s'_{n-k+2} \\ &\vdots \\ r_{n-1} &= s'_{n-1} - s'_n \\ \text{and} \quad r_n &= s'_n - s_1. \end{aligned}$$

With the help of these known quantities, decoder then forms the following system of n equations involving n unknowns $e_1, e_2, \ldots, e_n$:

$$\begin{aligned} r_1 &= \mathbf{e}_{(i)}(\mathbf{h}_1 - \mathbf{h}_2)^{\mathbf{T}} \\ r_2 &= \mathbf{e}_{(i)}(\mathbf{h}_2 - \mathbf{h}_3)^{\mathbf{T}} \\ &\vdots \\ r_{n-k} &= \mathbf{e}_{(i)}(\mathbf{h}_{n-k} - \mathbf{g}_1)^{\mathbf{T}} \\ r_{n-k+1} &= \mathbf{e}_{(i)}(\mathbf{g}_1 - \mathbf{g}_2)^{\mathbf{T}} \\ &\vdots \\ r_{n-1} &= \mathbf{e}_{(i)}(\mathbf{g}_{k-1} - \mathbf{g}_k)^{\mathbf{T}} \\ \text{and} \quad r_n &= \mathbf{e}_{(i)}(\mathbf{g}_k - \mathbf{h}_1)^{\mathbf{T}}. \end{aligned}$$

Since $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k, \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n-k} \in \left[\mathtt{GF}(q^n)\right]^n$ being linearly independent over $\mathtt{GF}(q^n)$, their difference vectors $(\mathbf{h}_1 - \mathbf{h}_2), (\mathbf{h}_2 - \mathbf{h}_3), \ldots, (\mathbf{h}_{n-k} - \mathbf{g}_1), (\mathbf{g}_1 - \mathbf{g}_2), (\mathbf{g}_2 - \mathbf{g}_3), \ldots, (\mathbf{g}_{k-1} - \mathbf{g}_k), (\mathbf{g}_k - \mathbf{h}_1) \in \left[\mathtt{GF}(q^n)\right]^n$ are also linearly independent over $\mathtt{GF}(q^n)$. On

solving the above system for the unknowns $e_1,\ e_2, \ldots,\ e_n$, one can determine $\mathbf{e}_{(i)}$ uniquely. Thus, the only unknown component β_j of the jth *basic message vector* $\boldsymbol{\beta}^{(j)}$ can be readily obtained from $\mathbf{e}^j_{(i)} = \beta_j(\mathbf{1}) + \mathbf{e}_{(i)}$ and consequently, the associated *basic message vector* is known to be $\boldsymbol{\beta}^{(j)}$. Once the jth *basic message vector* is identified, the original message vector can be retrieved as $\mathbf{m}^{(i)}_j = \boldsymbol{\beta}^{(j)} \otimes \mathbf{m}_{(i)}$. The decoding algorithm of *tertiary* codes described above is demonstrated through the following example, which is a continuation of previous example.

Example 2 (Decoding *tertiary* codes) Consider the *primary* code $\mathcal{C}(5, 3|2^{15}, 3)$.

$$\text{Consider } \mathbf{m}^{(i)} = (\alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4).$$
$$\text{Then } \mathbf{c}^{(i)}_2 = (\alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^{23}, \alpha^{11}).$$

$$\text{But, we consider } \mathbf{c}^2_{(i)} = (\alpha^2, \alpha^3, \alpha^4, \alpha^{23}, \alpha^{11}) \text{ and transmit } \mathbf{c}'_{(i)} = \mathbf{c}^2_{(i)} + \beta_2(\mathbf{1}).$$

Decoding the combined error Note that $(\alpha^{13}, \alpha^{28}, \alpha^{14}, \alpha^{12}, \alpha^5)$ is the codeword obtained by encoding the message vector $\mathbf{m}_{(i)} = (\alpha, \alpha^2, \alpha^3)$ with generator matrix $\mathbf{G}$ in non-systematic form which is in fact equivalent to $\mathbf{c}^2_{(i)} = (\alpha^2, \alpha^3, \alpha^4, \alpha^{23}, \alpha^{11})$ that is obtained earlier using the generator matrix $\mathbf{G}$ in systematic form.

Suppose that $\mathbf{r}^2_{(i)} = (\alpha^{24}, \alpha^7, \alpha^{15}, \alpha^{20}, \alpha^{26}) = (\alpha^{13}, \alpha^{28}, \alpha^{14}, \alpha^{12}, \alpha^5)+(\alpha, \alpha, \alpha, \alpha, \alpha^{30})$ $= \mathbf{c}^2_{(i)} + \mathbf{e}^2_{(i)}$ is the received vector, where $\mathbf{e}^2_{(i)} = (\alpha, \alpha, \alpha, \alpha, \alpha^{30}) = \beta_2(\mathbf{1}) + \mathbf{e}_{(i)} = (\alpha, \alpha, \alpha, \alpha, \alpha)+(0, 0, 0, 0, \alpha^4)$ is the combined error with $\mathbf{e}_{(i)} = \left(0, 0, 0, 0, \alpha^4\right) \in \left[\mathrm{GF}(2^5)\right]^5$ being the error-vector of unit-rank due to channel noise. By employing the rank-error correcting algorithm associated with the MRD code $\mathcal{C}(5, 3|2^{15}, 3)$, the 3-length vector $\mathbf{m}^2_{(i)} = \beta_j \mathbf{m}_{(i)} = (\alpha^2, \alpha^3, \alpha^4)$ can be recovered as the combined error $\mathbf{e}^2_{(i)}$ has only $e = \lfloor \frac{3-1}{2} \rfloor = 1$ rank error. Upon recovering the 3-tuple $\mathbf{m}^2_{(i)} = \beta_2 \mathbf{m}_{(i)}$ and the error-vector $\mathbf{e}^2_{(i)} = \mathbf{e}_{(i)} + \beta_2(\mathbf{1})$, the decoder then attempts to determine the actual message vector $\mathbf{m}^{(i)}_2 = \boldsymbol{\beta}^{(2)} \otimes \mathbf{m}_{(i)}$ originally transmitted—it remains now for the decoder to retrieve $\beta_2(\mathbf{1})$ from $\mathbf{e}^2_{(i)} = \mathbf{e}_{(i)} + \beta_2(\mathbf{1})$. We accomplish this task using the 2-tuple $\mathbf{r}^2_{(i)}\mathbf{H}^{\mathbf{T}}$ and 3-tuple $\mathbf{r}^2_{(i)}\mathbf{G}^{\mathbf{T}}$.

Decoding the channel error Consider $\mathbf{H}$-syndrome of received 5-tuple $\mathbf{r}^2_{(i)}$:

$$\begin{aligned}
\mathrm{S}_{\mathbf{H}} &= \mathbf{r}^2_{(i)}\mathbf{H}^{\mathbf{T}} \\
&= \left(\mathbf{c}^2_{(i)} + \beta_2(\mathbf{1}) + \mathbf{e}_{(i)}\right)\mathbf{H}^{\mathbf{T}} \\
&= \begin{pmatrix} \alpha^{24}, & \alpha^7, & \alpha^{15}, & \alpha^{20}, & \alpha^{26} \end{pmatrix} \begin{bmatrix} \alpha^{20} & \alpha^9 \\ \alpha^9 & \alpha^{18} \\ \alpha^{18} & \alpha^5 \\ \alpha^5 & \alpha^{10} \\ \alpha^{10} & \alpha^{20} \end{bmatrix} \\
&= \begin{pmatrix} \alpha^{15}, & \alpha^{13} \end{pmatrix} \\
&= \begin{pmatrix} s_1, & s_2 \end{pmatrix}
\end{aligned}$$

Further, the $\mathbf{G}$-syndrome of the received vector $\mathbf{r}^2_{(i)}$ is given by,

$$\begin{aligned}
\mathrm{S}_{\mathbf{G}} &= \mathbf{r}^2_{(i)}\mathbf{G}^{\mathbf{T}} \\
&= \begin{pmatrix} \alpha^{24}, & \alpha^{7}, & \alpha^{15}, & \alpha^{20}, & \alpha^{26} \end{pmatrix} \begin{bmatrix} \alpha^{18} & \alpha^{5} & \alpha^{10} \\ \alpha^{5} & \alpha^{10} & \alpha^{20} \\ \alpha^{20} & \alpha^{9} & \alpha^{18} \\ \alpha^{9} & \alpha^{18} & \alpha^{5} \end{bmatrix} \\
&= \begin{pmatrix} \alpha^{9}, & \alpha^{15}, & \alpha^{3} \end{pmatrix}.
\end{aligned}$$

$$\begin{aligned}
\text{Let } \mathrm{S}'_{\mathbf{G}} &= \mathbf{r}^j_{(i)}\mathbf{G}^{\mathbf{T}} - \mathbf{m}^j_{(i)} \\
&= \begin{pmatrix} \alpha^{24}, & \alpha^{26}, & \alpha^{21} \end{pmatrix} \\
&= \begin{pmatrix} s'_3, & s'_4, & s'_5 \end{pmatrix}.
\end{aligned}$$

Eliminating the unknown β_2 from $\mathbf{S_H}$ and $\mathbf{S}'_{\mathbf{G}}$:

$$\begin{aligned}
r_1 = s_1 - s_2 &= \alpha^{18} \\
r_2 = s_2 - s'_3 &= \alpha \\
r_3 = s'_3 - s'_4 &= \alpha^{29} \\
r_4 = s'_4 - s'_5 &= \alpha^{23} \\
\text{and } \; r_5 = s'_5 - s_1 &= \alpha^{11}.
\end{aligned}$$

With the help of these known quantities, decoder then forms the following system of 5 equations in 5 unknowns e_1, e_2, e_3, e_4, e_5:

$$\begin{aligned}
\alpha^{18} &= \mathbf{e}_i \begin{pmatrix} \alpha^{28} & \alpha^{25} & \alpha^{19} & \alpha^{7} & \alpha^{14} \end{pmatrix}^{\mathbf{T}} \\
\alpha &= \mathbf{e}_i \begin{pmatrix} \alpha^{25} & \alpha^{19} & \alpha^{7} & \alpha^{14} & \alpha^{28} \end{pmatrix}^{\mathbf{T}} \\
\alpha^{29} &= \mathbf{e}_i \begin{pmatrix} \alpha^{19} & \alpha^{7} & \alpha^{14} & \alpha^{28} & \alpha^{25} \end{pmatrix}^{\mathbf{T}} \\
\alpha^{23} &= \mathbf{e}_i \begin{pmatrix} \alpha^{7} & \alpha^{14} & \alpha^{28} & \alpha^{25} & \alpha^{19} \end{pmatrix}^{\mathbf{T}} \\
\text{and } \; \alpha^{11} &= \mathbf{e}_i \begin{pmatrix} \alpha^{14} & \alpha^{28} & \alpha^{25} & \alpha^{19} & \alpha^{7} \end{pmatrix}^{\mathbf{T}}.
\end{aligned}$$

On solving the above system of equations for the unknowns e_1, e_2, e_3, e_4, e_5 decoder obtains the channel error-vector $\mathbf{e}_{(i)} = (0, 0, 0, 0, \alpha^4)$. Thus, the only unknown component $\beta_2 = \alpha$ of the *basic message vector* $\boldsymbol{\beta}^{(2)}$ can be readily obtained from $\mathbf{e}^2_{(i)} = \beta_2(\mathbf{1}) + \mathbf{e}_{(i)} = (\alpha, \alpha, \alpha, \alpha, \alpha^{30})$ and consequently, the associated *basic message vector* is known to be $\boldsymbol{\beta}^{(2)} = (\alpha, \alpha, \alpha, \alpha, \alpha)$. Once $\boldsymbol{\beta}^{(2)}$ is identified, the original message vector can be retrieved as

$$\begin{aligned}
\mathbf{m}^{(i)}_2 &= \boldsymbol{\beta}^{(2)} \otimes \mathbf{m}_{(i)} \\
&= \boldsymbol{\beta}^{(2)} \otimes (\alpha, \alpha^2, \alpha^3) \\
&= (\alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4, \alpha^2, \alpha^3, \alpha^4).
\end{aligned}$$

Systematically, we are transmitting only n symbols to convey kn-length message symbols—eventually, the information rate $\mathcal{R}''$ of the *tertiary* code $\mathcal{C}'' = \mathcal{C}'_1 \cup \mathcal{C}'_2 \cup \cdots \cup \mathcal{C}'_{q^n-1}$ is

$\frac{\log_{q^n}|\mathcal{C}''|}{n} = \frac{\log_{q^n}[(q^n-1)q^{nk}]}{n} = \frac{k+[\log_{q^n}(q^n-1)]}{n} > \frac{k}{n} = \mathcal{R}$. Observe that, $|\mathcal{C}| = |\mathcal{C}'_j| = q^{nk}$, but $|\mathcal{C}'| = |\mathcal{C}''| = (q^n-1)q^{nk}$.

First choosing a *primary* code $\mathcal{C}$ with the required error correction (for the channel under consideration) followed by a selection of message length needed (i.e, choosing a $k^o > 0$), employing the proposed coding technique on $\mathcal{C}$ and $\mathcal{C}'$, one in fact obtains the code $\mathcal{C}''$ with information rate $\mathcal{R}''$ higher than the information rate $\mathcal{R}$ of $\mathcal{C}$. The increase in the information rate is made possible because of the addition of unit-rank vector prior to transmission. One can see that, the increase in code rate is only $\frac{\log_{q^n}(q^n-1)}{n}$. However, an improvement over the coding scheme described in association with the class of $\mathcal{T}$*-Direct* codes (Vasantha and Raja Durai 2002; Raja Durai and Devi 2011) would lead to a further increase in the code rate.

5 Conclusion

An important goal of coding theory is to construct codes that achieve a prescribed error-correction capability with a minimum amount of redundancy. The problem of constructing error-correcting codes that can meet the optimal trade-off between the information rate and error-correcting capability is considered. A coding scheme to construct codes (from existing codes) with higher code rate is proposed. Choosing a k^o-tuple with non-zero components, the paper first generates kk^o-length message vectors (for transmission) by taking Kronecker product of each message vector of an $\mathcal{C}(n, k|q^{nk}, d)$ MRD (*primary*) code with the chosen k^o-tuple. Then appending to it the corresponding $(n-k)$-tuple parity-check symbols as desired, one obtains a low-rate $\mathcal{C}'(kk^o+n-k, kk^o|q^{nk}, d+k^o-1)$-code. Finally, the coding technique proposed combines q^n-1 such low-rate *secondary* codes to obtain an high-rate *tertiary* code $\mathcal{C}''(n, kn|(q^n-1)q^{nk}, d)$ to harness an increase in the information rate. Associated decoding procedures to the codes constructed are also given. Further research work on the derived class of codes—*secondary* and *tertiary* codes—in association with the class of $\mathcal{T}$*-Direct* codes for a possible increase in the information rate is under consideration by the authors.

References

Alfaro R, Kelarev AV (2006) On cyclic codes in incidence rings. Studia Sci Math Hung 43(1):69–77

Alon N, Bruck J, Naor J, Naor M, Roth RM (1992) Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. IEEE Trans Inf Theory 38(2):509–516

Berrou C, Glavieux A, Thitimajshima P (1993) Near Shannon limit error-correcting coding and decoding: turbo codes. In: Proceedings of international conference on communications, Switzerland, Geneva, pp 1064–1070

Bose RC, Ray-Chaudhuri DK (1960) On a class of error correcting binary group codes. Inf Control 3(1):68–79

Cazaran J, Kelarev AV (1999) On finite principal ideal rings. Acta Math Univ Comeniae 68(1):77–84

Delsarte P (1978) Bilinear forms over a finite field with applications to coding theory. J Combin Theory A 25(3):226–241

Gabidulin EM (1985) Theory of codes with maximum rank distance. Probl Inf Transm 21:1–12

Gallager RG (1963) Low-density parity-check codes. MIT Press, Cambridge

Golay MJE (1949) Notes on digital coding. Proc IRE (corresp) 37(6):657

Guruswami V, Indyk P (2005) Linear-time encodable/decodable codes with near-optimal rate. IEEE Trans Inf Theory 51(10):3393–3400

Hamming RW (1950) Error detecting and error correcting codes. Bell Syst Tech J 29:147–160

Hua L-K (1951) A theorem on matrices over a field and its applications. Chin Math Soc 1(2):109–163

Kelarev AV (2004a) Combinatorial and statistical algorithms for information rates of codes defined by directed graphs. In: Proceedings of 15th Australasian Workshop on Combinatorial Algorithms (AWOCA). New South Wales, Australia, pp 43–51

Kelarev AV (2004b) Minimum distances and information rates for matrix extensions of BCH codes. In: 3rd Workshop on the internet, telecommunications and signal processing (WITSP), Adelaide, pp. 1–6
Kelarev AV (2006) Computing the information rates for a class of polynomial codes. In: 5th workshop on the internet, telecommunications and signal processing (WITSP), Hobart, Australia, pp. 11–13 (2006)
Kelarev AV (2002) Ring constructions and applications. World Scientific, River Edge
Kelarev AV (2005) A statistical algorithm for computing information rates of codes with computer algebra systems. Adv Appl Stat 5(1):87–90
Kelarev AV (2007) Algorithms for computing parameters of graph-based extensions of BCH codes. Discrete Algorithms 5(3):553–563
Kelarev AV (2008) An algorithm for BCH codes extended with finite state automata. Fundamenta Informaticae 84(2):51–60
Lempel A (1975) Matrix factorization over F_2 and trace-orthogonal bases of F_{2^n}. SIAM J Comput 4:175–186
Lempel A, Weinberger M (1988) Self-complementary normal bases in finite fields. SIAM J Discrete Math 1:193–198
Lidl R, Niederreiter H (1986) Introduction to finite fields and their applications. Cambridge University Press, Cambridge
Muller DE (1954) Application of Boolean algebra to switching circuit design and to error detection. IRE Trans Electron Comput 3:6–12
Raja Durai RS, Devi M (2011) Construction of $(\mathcal{N}+\mathcal{M})$-Direct codes in $\mathrm{GF}(2^{\mathcal{N}})$. In: Proceedings of world congress on information and communication technologies (WICT), Mumbai, India, pp 770–775
Reed IS, Solomon G (1960) Polynomial codes over certain finite fields. SIAM J Appl Math 8(2):300–304
Richard C (1964) Singleton, Maximum distance q-ary codes. IEEE Trans Inf Theory 10(2):116–118
Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423, 623–656
Vasantha Kandasamy WB, Smarandache F, Sujatha R, Raja Durai RS (2012) Erasure techniques in MRD codes. ZIP Publishing, Columbus
Vasantha WB, Raja Durai RS (2002) $\mathcal{T}$-Direct codes: an application to $\mathcal{T}$-user BAC. In: Proceedings of IEEE information theory workshop, Bangalore, India