IOP Publishing Journal of Physics A: Mathematical and Theoretical

J. Phys. A: Math. Theor. **53** (2020) 055202 (28pp) https://doi.org/10.1088/1751-8121/ab5f89

Decoding algorithm as a moment problem related to the extended Lotka–Volterra system

Yan Pan[1,2], Xiang-Ke Chang[1,2,3] and Xing-Biao Hu[1,2]

[1] LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, PO Box 2719, Beijing 100190, People's Republic of China
[2] School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China

E-mail: panymath@lsec.cc.ac.cn, changxk@lsec.cc.ac.cn and hxb@lsec.cc.ac.cn

Received 17 June 2019, revised 25 November 2019
Accepted for publication 6 December 2019
Published 8 January 2020

Abstract

The moment problem related to the extended Lotka–Volterra system (sometimes also called the hungry Lotka–Volterra system or the Narita–Itoh–Bogoyavlensky lattice) over finite fields is introduced. It turns out the moment problem could be used to design an algorithm for decoding multiple BCH-Goppa codes over the same finite field simultaneously. When multiple codes have the same error locations, the algorithm requires fewer known syndromes and has lower computational complexity than running the decoding algorithm in Nakamura (1996 *Phys. Lett.* A **223** 75–81) multiple times.

Keywords: BCH-Goppa code, extended Lotka–Volterra lattice, moment problem

1. Introduction

The finite nonperiodic Toda equation over finite fields has shown its importance to the theory of error-correcting codes [1–4]. The pioneering work on this issue is that Faybusovish applied quotient difference (qd) algorithm to the decoding procedure of Goppa codes [1], where the qd algorithm [5] is indeed a discrete-time version of the Toda equation. Shirota suggested that the Stieltjes method for the eigenvector of a Jacobi matrix that appears in the Lax representation also works in the decoding approach to the Reed–Solomon codes (RS codes) [6]. Later, Nakamura put forward a method to transform the BCH-Goppa decoding procedure [7] into the moment problem of the finite nonperiodic Toda equation and proposed a new BCH-Goppa decoding algorithm through a Lax representation of finite Toda equation [2]. Further work

[3] Author to whom any correspondence should be addressed.

on this topic is a new BCH-Goppa decoding algorithm derived from the discrete-time finite Toda molecule [3]. The idea of the algorithm is to calculate a continued fraction expansion of rational approximant of the formal Laurent series over a finite field through qd algorithm [8]. In fact, the famous Berlekamp–Massey algorithm in coding theory [9, 10] may naturally follow from solving the problem of rational approximant of the formal Laurent series over a finite field. These works promote the development of other related subjects such as orthogonal polynomials [11, 12], combinatorics [10], continued fractions [13] etc.

Since Toda type lattices and their discretizations have been studied extensively over the last fifty years, we expect them to shed more light to decoding algorithms. Inspired by the corresponding tau functions and the spectral problems, we would investigate a family of modifications of the Toda lattice called the extended Lotka–Volterra lattice.

The Lotka–Volterra system

$$\frac{\mathrm{d}a_k}{\mathrm{d}t} = a_k\left(a_{k+1} - a_{k-1}\right), \tag{1}$$

is an integrable system, which is intimately related to the Toda equation [14, 22]

$$\frac{\mathrm{d}u_k}{\mathrm{d}t} = u_k(b_{k+1} - b_k), \quad \frac{\mathrm{d}b_k}{\mathrm{d}t} = 2(u_k^2 - u_{k-1}^2). \tag{2}$$

The moment problem and linearized deformation equation of the finite Lotka–Volterra system can be regarded as a reduced form of that for the finite nonperiodic Toda equation [14–16]. The extended Lotka–Volterra lattice

$$\frac{\mathrm{d}a_k}{\mathrm{d}t} = a_k\left(\prod_{j=1}^{M} a_{k+j} - \prod_{j=1}^{M} a_{k-j}\right), \quad \text{for fixed } M \in \mathrm{N} \tag{3}$$

was proposed by Narita [17], Itoh [18], and Bogoyavlensky [19–21] as a generalization of the Lotka–Volterra lattice. Sometimes it is called the Narita–Itoh–Bogoyavlensky lattice or the hungry Lotka–Volterra lattice, but for our convenience, we shall call it the extended Lotka–Volterra lattice throughout the paper. It can also be linearized via the Moser's map [14, 22]. To the best of our knowledge, there is not any decoding algorithm corresponding to the extended Lotka–Volterra lattice (3) available in the literature. The main purpose of this paper is to propose a BCH-Goppa decoding algorithm based on the moment problem related to the extended Lotka–Volterra system (3). This algorithm could be used to find errors for multiple codewords over the same field $\mathrm{GF}(q^u)$ simultaneously. Compared with running the BCH-Goppa decoding algorithm in [2] for a single code multiple times, our approach seems more efficient in the case that the multiple codes have the same error locations.

The outline of this paper is as follows: in section 2, some basic facts of moment problem and coding theory are provided and a brief review of the BCH-Goppa decoding algorithm with the context of moment problem is given. In section 3, the moment problem of the extended Lotka–Volterra systems (3) is introduced. In section 4, the corresponding decoding procedure for multiple codewords is designed and some numerical examples are presented. Section 5 is devoted to conclusion.

2. Basic notations of moment problem and BCH-Goppa decoding

In this section, some basic facts with adaptive symbols on the moment problem [23, 24] and the coding theory [25] are given to make this paper self-contained.

2.1. Moment problem

Assume $f(z)$ admits an expression as

$$f(z) = \int_{-\infty}^{\infty} \frac{\mathrm{d}\mu(\lambda)}{z-\lambda}, \quad \mathrm{d}\mu(\lambda) = \sum_{j=1}^{n} r_j^2 \delta(\lambda - \lambda_j)\mathrm{d}\lambda,$$

where $\delta(z)$ is the delta function. If we define the moments $\langle z^k \rangle$ under the discrete Stieltjes measure $\mu(z)$ as

$$\langle z^k \rangle = \int_{-\infty}^{\infty} z^k \mathrm{d}\mu(z) = \sum_{j=1}^{n} \lambda_j^k r_j^2,$$

then it is obvious that

$$f(z) = \sum_{j=1}^{n} \frac{r_j^2}{z-\lambda_j} = \frac{1}{z} \frac{\sum_{j=1}^{n} r_j^2}{1-\frac{\lambda_j}{z}} = \sum_{k=0}^{\infty} \frac{\langle z^k \rangle}{z^{k+1}}. \tag{4}$$

In this paper, the moment problem is to find a parametric Stieltjes measure $\mu(z)$ from given moments $\langle z^k \rangle$.

2.2. Introduction to BCH-Goppa decoding

Let N, K, q, u, l and r be positive integers, where $r + K \leqslant N$, q is a power of a prime, q and N are coprime.

Definition 2.1 (BCH code). Let α be a primitive Nth root of unity over a finite field $\mathrm{GF}(q^u)$, i.e. $\alpha^N = 1$. A cyclic code of length N over $\mathrm{GF}(q)$ is called a BCH code, if $\alpha^l, \alpha^{l+1}, \cdots, \alpha^{l+r-1}$ are roots of its generator polynomial $G(x)$. In other words, all the BCH codes can be equivalently written as

$$C = \{c(x) = \sum_{j=0}^{N-1} c_j x^j \in \mathrm{GF}(q)(x) | c(\alpha^l) = c(\alpha^{l+1}) = \ldots = c(\alpha^{l+r-1}) = 0\}.$$

If $N = q^u - 1$, i.e. α is a primitive element of $\mathrm{GF}(q^u)$, then the BCH code is called primitive.

Definition 2.2 (RS code). A Reed–Solomon code is a primitive BCH code with $u = 1$. The generator of such a code has the form $G(x) = \prod_{j=l}^{l+r-1}(x - \alpha^j)$ where α is a primitive in $\mathrm{GF}(q)$.

Definition 2.3 (Goppa code). Let $M(x)$ be a (monic) polynomial of degree r over $\mathrm{GF}(q^u)$. Let $\mathcal{L} = \{\alpha_0, \ldots, \alpha_{N-1}\} \subset \mathrm{GF}(q^u)$ such that $\alpha_0, \ldots, \alpha_{N-1}$ are mutually distinct elements of the finite field $\mathrm{GF}(q^u)$ and $M(\alpha_j) \neq 0$ for $j = 0, \ldots, N-1$. We define the Goppa code $\Gamma(\mathcal{L}, M)$ with the Goppa polynomial $M(x)$ to be the set of code words $c = (c_0, \ldots, c_{N-1})$ over $\mathrm{GF}(q)$ for which

$$\sum_{j=0}^{N-1} \frac{c_j}{x - \alpha_j} = 0 \pmod{M(x)}.$$

We now introduce the parity check matrix for a kind of code C in the coding theory, which means a matrix such that $cH^{\top} = 0$ for every $c = (c_0, \ldots, c_{N-1}) \in C$.

Theorem 2.1 ([25]). *The BCH code has the parity check matrix of the form*

$$H = \begin{pmatrix} 1 & \alpha^{l} & \ldots & \alpha^{l(N-1)} \\ 1 & \alpha^{(l+1)} & \ldots & \alpha^{(l+1)(N-1)} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha^{(l+r-1)} & \ldots & \alpha^{(l+r-1)(N-1)} \end{pmatrix}.$$

Theorem 2.2 ([25]). *The Goppa code has the parity-check matrix of the form*

$$H = \begin{pmatrix} M(\alpha_0)^{-1} & M(\alpha_1)^{-1} & \ldots & M(\alpha_{N-1})^{-1} \\ M(\alpha_0)^{-1}\alpha_0 & M(\alpha_1)^{-1}\alpha_1 & \ldots & M(\alpha_{N-1})^{-1}\alpha_{N-1} \\ \vdots & \vdots & & \vdots \\ M(\alpha_0)^{-1}\alpha_0^{r-1} & M(\alpha_1)^{-1}\alpha_1^{r-1} & \ldots & M(\alpha_{N-1})^{-1}\alpha_{N-1}^{r-1} \end{pmatrix}.$$

Assume that the sent codeword is $c = (c_0, \ldots, c_{N-1})$ and the received codeword is $b = (b_0, \ldots, b_{N-1}) = c + e$, where $e = (e_0, \ldots, e_{N-1})$ is the unknown error, a decoding problem is to find the error e from the syndrome sequence

$$s = (S_0, S_1, \ldots, S_{r-1}) = bH^{\top} = eH^{\top},$$

where H is the corresponding parity check matrix. Define the syndrome polynomial as

$$S(x) = \sum_{k=0}^{N-1} S_k x^k,$$

where S_j may be readily computed from the remainder. The main idea of the BCH-Goppa decoding [7] is to find polynomials $\omega(z)$ and $\sigma(z)$ such that

$$\frac{\omega\left(\frac{1}{x}\right)}{\sigma\left(\frac{1}{x}\right)} = \sum_{k=0}^{N-1} S_k x^{k+1} = xS(x) \pmod{x^{r+1}}, \tag{5}$$

$$\deg(\sigma(x)) \leqslant \left\lfloor \frac{r}{2} \right\rfloor, \quad \deg(\omega(x)) \leqslant \left\lfloor \frac{r}{2} \right\rfloor - 1, \tag{6}$$

from which one can know the error positions by factoring the error-locator polynomial $\sigma(x)$, and consequently obtain the error values.

In the BCH case, let $G(x)$ be the generator polynomial. If $G(x)$ has degree $N - K$, we encode an information sequence $(a_0, a_1, \ldots, a_{K-1})$ as a polynomial $a(x)G(x) = (a_0 + a_1 x + \ldots + a_{K-1}x^{K-1})G(x)$. The encoder transmits the codeword $c(x) = \sum_{j=0}^{N-1} c_j x^j$ satisfying the condition $c(x) = a(x)G(x)$. (For the sake of simplicity, let $a(x) = 1$ in this paper.) The received word can be expressed by the polynomial $b(x) = \sum_{j=0}^{N-1} b_j x^j$, and then the error word will be given by $e(x) = \sum_{j=0}^{N-1} e_j x^j = \sum_{j=0}^{N-1} b_j x^j - \sum_{j=0}^{N-1} c_j x^j$.

Let $J = \{j | e_j \neq 0\}$ be the positions where an error occurs. Define P as the number of errors. Then, in the BCH case, the syndromes are

$$S_k = b(\alpha^{(l+k)}) = \sum_{j=0}^{N-1} b_j \alpha^{j(l+k)} = \sum_{j\in J} e_j \alpha^{j(l+k)} = e(\alpha^{(l+k)})$$

for $k = 0, \ldots, r-1$. Similarly, in the Goppa case,

$$S_k = \sum_{j=0}^{N-1} M(\alpha_j)^{-1} b_j \alpha_j^k = \sum_{j\in J} M(\alpha_j)^{-1} e_j \alpha_j^k.$$

for $k = 0, \ldots, r-1$.

Observing that the syndromes are finite summations with number P related to the errors, and the expansion of $\frac{\omega(z)}{\sigma(z)}$ is similar to (4), it was shown by Nakamura that the P-error decoding process to find out $e_j, j \in J$ from $S_0, \ldots, S_{2P-1}$, can be seen as a moment problem and can be solved with the help of the moment problem related to the Toda lattice according to the following diagram designed in [2]:

$$\begin{array}{ccccc} \{u_k, b_k\} & \xrightarrow{(8)} & \{r_p^2, \lambda_p\} & \longrightarrow & \{e_j, \alpha^j\} \\ (10)\uparrow & & \uparrow\downarrow (9) & & \uparrow \text{ want} \\ \{\Delta_k, \tilde{\Delta}_k\} & \xleftarrow[(11)]{} & \{h_k\} & \longleftarrow & \{S_k/S_0\}. \end{array} \tag{7}$$

2.3. Decoding as a moment problem related to Toda lattice

On one hand, the finite Toda equation has the Lax representation $\frac{\mathrm{d}L}{\mathrm{d}t} = [B, L]$, where

$$L = \begin{pmatrix} b_1 & u_1 & & \\ u_1 & b_2 & \ddots & \\ & \ddots & \ddots & u_{P-1} \\ & & u_{P-1} & b_P \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & u_1 & & \\ -u_1 & 0 & \ddots & \\ & \ddots & \ddots & u_{P-1} \\ & & -u_{P-1} & 0 \end{pmatrix}$$

are both tridiagonal matrices. Consider the rational function

$$f(z) = E_1^\top (zI - L)^{-1} E_1, \text{ where } E_1 = (1, 0, \ldots, 0)^\top. \tag{8}$$

It has not only the partial fraction expansion

$$f(z) = \sum_{p=1}^{P} \frac{r_p^2}{z - \lambda_p} \text{ with } \sum_{p=1}^{P} r_p^2 = 1,$$

but also the series expansion

$$f(z) = \sum_{k=0}^{\infty} \frac{h_k}{z^{k+1}},$$

where

$$h_0 = 1 \text{ and } h_k = \sum_{p=1}^{P} \lambda_p^k r_p^2. \tag{9}$$

On the other hand, a solution of the Toda equation can be constructed by the direct formula

$$\begin{aligned} u_k &= \frac{\sqrt{\Delta_{k-1}\Delta_{k+1}}}{\Delta_k}, \quad k = 1, 2, \ldots, P-1, \\ b_k &= \frac{\tilde{\Delta}_k}{\Delta_k} - \frac{\tilde{\Delta}_{k-1}}{\Delta_{k-1}}, \quad k = 1, 2, \ldots, P, \end{aligned} \tag{10}$$

where $\Delta_k, \tilde{\Delta}_k$ are determinants of Hankel type constructed from the moments, that is

$$\begin{aligned} \Delta_k &= \begin{vmatrix} h_0 & h_1 & \ldots & h_{k-1} \\ h_1 & h_2 & \ldots & h_k \\ \vdots & \vdots & & \vdots \\ h_{k-1} & h_k & \ldots & h_{2k-2} \end{vmatrix}, k = 1, 2, \ldots \\ \tilde{\Delta}_1 &= h_1, \qquad \tilde{\Delta}_k = \begin{vmatrix} h_0 & h_1 & \ldots & h_{k-2} & h_k \\ h_1 & h_2 & \ldots & h_{k-1} & h_{k+1} \\ \vdots & \vdots & & \vdots & \vdots \\ h_{k-1} & h_k & \ldots & h_{2k-3} & h_{2k-1} \end{vmatrix}, k = 2, 3, \ldots \quad . \end{aligned} \tag{11}$$

Therefore, if one wants to find out the error, one can firstly calculate S_k from the received codeword and then construct the determinant $\Delta_k, \tilde{\Delta}_k$ to calculate u_k, b_k if $\Delta_k \neq 0$ for $k = 1, 2, \ldots, P$. Finally, consider the partial fraction expansion of the rational function $f(z)$ to get e_j, α_j.

Remark 2.1. Here we remark that, since L is symmetric, the numerator and denominator of the rational function $f(z)$ can be obtained by recursion relationship with coefficients u_k^2 and b_k. Hence the square root in the definition of u_k (10) will not cause any problem.

3. Moment problem related to extended Lotka–Volterra lattice

In this section, we focus on the moment problem related to the extended Lotka–Volterra lattice

$$\frac{\mathrm{d}a_s}{\mathrm{d}t} = a_s \left(\prod_{j=1}^{M} a_{s+j} - \prod_{j=1}^{M} a_{s-j} \right) \tag{12}$$

and show how to solve certain coding problem by using the corresponding moment problem in the following section.

3.1. From bi-orthogonal polynomials to extended Lotka–Volterra latttice

The extended Lotka–Volterra lattice (12) can be derived from the bi-orthogonal polynomials defined in [26, 27]. For a fixed positive integer M, let $\{P_n(z)\}_{n=0}^{\infty}$ and $\{Q_s(z)\}_{n=0}^{\infty}$ be two classes of adjacent monic polynomials satisfying the bi-orthogonal condition

$$\langle P_n(z), Q_s(z) \rangle_M = h_n \delta_{n,s}, \quad n, s = 0, 1, \ldots,$$

where the bilinear functional has the form

$$\langle f(z), g(z) \rangle_M = \int_{-\infty}^{\infty} f(z^M) g(z) w(z) \mathrm{d}\mu(z).$$

If the weight function $w(z)$ satisfies

$$\int_{-\infty}^{\infty} z^{(M+1)j} w(z)\mathrm{d}\mu(z) = g_j$$

and

$$\int_{-\infty}^{\infty} z^{(M+1)j+1} w(z)\mathrm{d}\mu(z) = \cdots = \int_{-\infty}^{\infty} z^{(M+1)j+M} w(z)\mathrm{d}\mu(z) = 0,$$

then it is not hard to prove that the polynomials $P_n(z)$ and $Q_n(z)$ can be expressed as

$$P_{(M+1)k+m}(z) = \frac{z^m}{G_k^{(m)}} \begin{vmatrix} g_m & g_{m+1} & \cdots & g_{m+k-1} & 1 \\ g_{m+M} & g_{m+M+1} & \cdots & g_{m+M+k-1} & z^{M+1} \\ \vdots & \vdots & & \vdots & \vdots \\ g_{m+kM} & g_{m+kM+1} & \cdots & g_{m+kM+k-1} & z^{(M+1)k} \end{vmatrix}$$

and

$$Q_{(M+1)k+m}(z) = \frac{z^m}{G_k^{(m)}} \begin{vmatrix} g_m & g_{m+1} & \cdots & g_{m+k} \\ g_{m+M} & g_{m+M+1} & \cdots & g_{m+M+k} \\ \vdots & \vdots & & \vdots \\ g_{m+M(k-1)} & g_{m+M(k-1)+1} & \cdots & g_{m+M(k-1)+k} \\ 1 & z^{M+1} & \cdots & z^{(M+1)k} \end{vmatrix}, \tag{13}$$

where $k = \lfloor \frac{n}{M+1} \rfloor$, $m = 0, 1, \ldots, M$ and

$$G_0^{(m)} = 1, \quad G_1^{(m)} = g_m,$$

$$G_k^{(m)} = \begin{vmatrix} g_m & g_{m+1} & \cdots & g_{m+k-1} \\ g_{m+M} & g_{m+M+1} & \cdots & g_{m+M+k-1} \\ \vdots & \vdots & & \vdots \\ g_{m+M(k-1)} & g_{m+M(k-1)+1} & \cdots & g_{m+M(k-1)+k-1} \end{vmatrix}. \tag{14}$$

The polynomials $Q_s(z)$ have the following recurrence relation

$$z^M Q_s(z) = Q_{s+M}(z) + a_s Q_{s-1}(z) \tag{15}$$

where

$$a_{(M+1)k+m} = \begin{cases} \frac{G_{k+1}^{(0)} G_{k-1}^{(M)}}{G_k^{(0)} G_k^{(M)}} & \text{for } m = 0, \\ \frac{G_{k+1}^{(m)} G_k^{(m-1)}}{G_{k+1}^{(m-1)} G_k^{(m)}} & \text{for } m = 1, \ldots, M. \end{cases} \tag{16}$$

If we take a single parameter deformation of measure $\mu(x)$,

$$\mathrm{d}\mu(z,t) = \exp(z^{M(M+1)}t)\mathrm{d}\mu(z,0).$$

then the moments admit the time evolution

$$\frac{\mathrm{d}g_j}{\mathrm{d}t} = g_{j+M},$$

which results in the the recursion relationship of polynomials

$$\frac{\mathrm{d}Q_s(z)}{\mathrm{d}t} = -\prod_{j=0}^{M} a_{s-M+j} Q_{s-(M+1)}(z). \tag{17}$$

The compatibility condition of (15) and (17) leads to the extended Lotka–Volterra lattice (12). This implies that the Lax pair of the finite extended Lotka–Volterra lattice reads

$$\frac{\mathrm{d}L}{\mathrm{d}t} = [B, L] \tag{18}$$

where B and L are $n \times n$ matrices

$$L = \begin{pmatrix} \overbrace{0 \quad \cdots \quad 0}^{M} & 1 & & & \\ a_1 & 0 & \cdots 0 & 1 & & \\ & a_2 & \ddots & \ddots & \ddots & \\ & & \ddots \ddots & & \ddots & 1 \\ & & \ddots & \ddots & & 0 \\ & & & \ddots & \ddots & \vdots \\ & & & & a_{n-1} & 0 \end{pmatrix}, \tag{19}$$

$$B = \begin{pmatrix} 0 & & & & & & & \\ \vdots & 0 & & & & & & \\ 0 & & \ddots & & & & & \\ -\prod_{j=0}^{M} a_{1+j} & \ddots & & & \ddots & & & \\ & \ddots & \ddots & & & \ddots & & \\ & & & \ddots & \ddots & & \ddots & \\ & & & & -\prod_{j=0}^{M} a_{n-M-1+j} & \underbrace{0 \quad \cdots \quad 0}_{M+1} & & \end{pmatrix}. \tag{20}$$

3.2. Linearization of extended Lotka–Volterra lattice

In this section, we adopt Moser's approach [14, 22] and Bogoyavlensky's method [28] to introduce a set of rational functions to give an integrable linearization of the extended Lotka–Volterra lattice (12).

Assume that all eigenvalues of (19) are simple. Consider the vth element of resolvent operator $R(z) = (zI - L)^{-1}$ for $v = 1, 2, \ldots, M$. Define the function

$$f_v(z) = E_v^{\top}(zI - L)^{-1}E_v = \sum_{j=1}^{n} \frac{\rho_{j,v}}{z - \lambda_j}, \quad E_v = (\overbrace{0, \ldots, 0}^{v-1}, 1, 0, \ldots, 0)^{\top}, \tag{21}$$

where the vth element of E_v is 1 and other elements of E_v are zero. Moreover, $zf_v(z) \to 1$ as $z \to \infty$, thus we have

$$\sum_{j=1}^{n} \rho_{j,v} = 1. \tag{22}$$

In fact, $f_v(z)$ is the element of the vth row and vth column of $R(z)$ so it can be expressed by

$$f_v(z) = \frac{M_{v,v}(z)}{|zI - L|} \tag{23}$$

where $M_{v,v}(z)$ is the (v, v) cofactor of the matrix $(zI - L)$. Since only 3 diagonals of L are nonzero, $M_{v,v}(z)$ and $|zI - L|$ can be computed quickly through recurrence relations.

Remark 3.1. If we set $T_j(z)$ to be the determinant of the matrix obtained from the last j rows and last j columns of $zI - L$, we have recurrence relations

$$T_j(z) = zT_{j-1}(z) - \prod_{l=1}^{M} a_{n-j+l} T_{j-1-M}(z) \quad \text{for } j = M+1, ..., n,$$
$$T_j(z) = z^j \quad \text{for } j = 0, ..., M.$$

Besides, it is not hard to see that

$$M_{v,v}(z) = z^{v-1} T_{n-v}(z).$$

Although we can consider all the rational functions $f_v(z)$, $v = 1, \ldots, n$, the ones for $v = M+1, \ldots, n$ can be expressed by the linear combinations of the ones for $v = 1, \ldots, M$ by the above recurrence relations.

Remark 3.2. Some of $\rho_{j,v}$ could be zero, which implies that the error values of some codes can be zero in the coding theory. See example 4.5 for more details.

We define $\kappa = \exp \frac{2\pi i}{M+1}$ so that $\kappa^{M+1} = 1$. Let Q be a diagonal matrix with the diagonal entries $Q_{jj} = \kappa^j, j = 1, \ldots, n$. It follows that

$$Q^{-1} L Q = \kappa^{-1} L.$$

Since

$$Q^{-1}(zI - L)^{-1} Q = [Q^{-1}(zI - L)Q]^{-1} = (zI - \kappa^{-1} L)^{-1} = \kappa(\kappa z I - L)^{-1},$$

one can obtain

$$Q^{-1} R(z) Q = \kappa R(\kappa z),$$

which leads to

$$f_v(z) = E_v^\top R(z) E_v = E_v^\top Q^{-1} R(z) Q E_v = E_v^\top \kappa R(\kappa z) E_v = \kappa f_v(\kappa z). \tag{24}$$

By substituting the expansion (21) into (24), we get

$$f_v(z) = \sum_{j=1}^{n} \frac{\rho_{j,v}}{z - \lambda_j} = \kappa \sum_{j=1}^{n} \frac{\rho_{j,v}}{\kappa z - \lambda_j} = \sum_{j=1}^{n} \frac{\rho_{j,v}}{z - \kappa^{-1}\lambda_j}.$$

Hence the corresponding residues ρ_j, ρ_k are identical when there exists an integer m such that $\lambda_j = \kappa^m \lambda_k$. In the case that all eigenvalues are simple, there are $\lfloor \frac{n}{M+1} \rfloor$ sets of nonzero eigenvalues $\lambda_{p,v}$ with the same residues ρ_p and zero eigenvalue with the corresponding residue.

For simplicity, we mainly consider the case that $n = (M+1)P$, In this case, the eigenvalues are simple and all the eigenvalues are nonzero. If we define $P = \frac{n}{M+1}$ and group the terms in (21) by the identical residues, then it turns out that $f_v(z)$ has the following form

$$f_v(z) = \sum_{p=1}^{P} \frac{(M+1)z^M \rho_{p,v}}{z^{M+1} - \lambda_p^{M+1}}.$$

Let

$$h_{Mu+v-1} = \sum_{p=1}^{P} (M+1)\rho_{p,v}\lambda_p^{(M+1)u} \text{ and } h_{v-1} = 1. \tag{25}$$

Then the expansion (21) becomes

$$\begin{aligned} f_v(z) &= \textstyle\sum_{p=1}^{P} \frac{(M+1)z^M \rho_{p,v}}{z^{M+1} - \lambda_p^{M+1}} \\ &= \textstyle\sum_{u=0}^{\infty} \frac{\sum_{p=1}^{P} (M+1)\rho_{p,v}\lambda_p^{(M+1)u}}{z^{(M+1)u+1}} \\ &= \textstyle\sum_{u=0}^{\infty} \frac{h_{Mu+v-1}}{z^{(M+1)u+1}}. \end{aligned} \tag{26}$$

Let $\lambda_1, \ldots, \lambda_n$ and $\psi_1, \ldots, \psi_n$ be the left eigenvalues and the left eigenvectors of the operator L. There exists coefficients $\eta_{1,v}, \ldots, \eta_{n,v}$ such that

$$E_v^\top = \eta_{1,v}\psi_1 + \ldots + \eta_{n,v}\psi_n.$$

Then we have

$$\psi_j R(z) = \frac{1}{z - \lambda_j}\psi_j,$$

$$f_v(z) = E_v^\top R(z) E_v = \sum_{j=1}^{n} \frac{\eta_{j,v}\psi_j E_v}{z - \lambda_j} = \sum_{j=1}^{n} \frac{\rho_{j,v}}{z - \lambda_j}$$

so that $\rho_{j,v} = \eta_{j,v}\psi_j E_v$. There is no difficulty to carry out

$$\frac{\mathrm{d}R(z)}{\mathrm{d}t} = [B, R(z)]$$

leading to

$$\begin{aligned} \tfrac{\mathrm{d}f_v(z)}{\mathrm{d}t} &= E_v^\top \tfrac{\mathrm{d}R(z)}{\mathrm{d}t} E_v = -E_v^\top R(z) B E_v \\ &= \textstyle\prod_{m=0}^{M} a_{v+m} E_v^\top R(z) E_{M+v+1} = \prod_{m=0}^{M} a_{v+m} \sum_{j=1}^{n} \frac{\eta_{j,v}\psi_j E_{M+v+1}}{z - \lambda_j}. \end{aligned} \tag{27}$$

Denote $\psi_{j,k} = \psi_j E_k$. For L defined in (19), it has the recursive relationships

$$\begin{aligned} &a_1\psi_{j,2} = \lambda_j\psi_{j,1}, \ \cdots, \ a_M\psi_{j,M+1} = \lambda_j\psi_{j,M}, \\ &\psi_{j,1} + a_{M+1}\psi_{j,M+2} = \lambda_j\psi_{j,M+1}, \ \cdots, \ \psi_{j,v} + a_{M+v}\psi_{j,M+v+1} = \lambda_j\psi_{j,M+v}, \end{aligned}$$

which leads to

$$\psi_{j,M+v+1} = \frac{\lambda_j^{M+1} - \sum_{k=1}^{v} \prod_{l=0}^{M-1} a_{k+l}}{\prod_{l=0}^{M} a_{v+l}} \psi_{j,v}. \tag{28}$$

Then, one can obtain

$$\frac{\mathrm{d}f_v(z)}{\mathrm{d}t} = \sum_{j=1}^{n} \frac{\lambda_j^{M+1} - \sum_{k=1}^{v} \prod_{l=0}^{M-1} a_{k+l}}{z-\lambda_j} \rho_{j,v}$$

by substituting the expression (28) into (27). Assume that λ_j is independent of time, then we have

$$\frac{\mathrm{d}f(z)}{\mathrm{d}t} = \sum_{j=1}^{n} \frac{\frac{\mathrm{d}\rho_{j,v}}{\mathrm{d}t}}{z-\lambda_j} = \sum_{j=1}^{n} \frac{\lambda_j^{M+1} - \sum_{k=1}^{v} \prod_{l=0}^{M-1} a_{k+l}}{z-\lambda_j} \rho_{j,v}$$

and subsequently

$$\frac{\mathrm{d}\rho_{j,v}}{\mathrm{d}t} = (\lambda_j^{M+1} - \sum_{k=1}^{v} \prod_{l=0}^{M-1} a_{k+l})\rho_{j,v}.$$

Adding the above equations for the index j from 1 to n, by using the condition (22), we obtain

$$0 = \sum_{j=1}^{n} \rho_{j,v} \lambda_j^{M+1} - \sum_{k=1}^{v} \prod_{l=0}^{M-1} a_{k+l}.$$

On account of (25), we have

$$\sum_{k=1}^{v} \prod_{l=0}^{M-1} a_{k+l} = \sum_{j=1}^{n} \rho_{j,v} \lambda_j^{M+1} = \sum_{p=1}^{P} (M+1) \rho_{p,v} \lambda_p^{M+1} = h_{M+v-1}.$$

It follows that

$$\frac{\mathrm{d}h_{Mu+v-1}}{\mathrm{d}t} = h_{M(u+1)+v-1} - h_{M+v-1} h_{Mu+v-1}.$$

Introducing g_j with

$$\frac{g_{Mu+v-1}}{g_{v-1}} = h_{Mu+v-1} \text{ and } \frac{\mathrm{d}}{\mathrm{d}t} \log g_{v-1} = h_{M+v-1},$$

for $u = 1, 2, \ldots, r_v - 1$ and $v = 1, 2, \ldots, M$, it follows that

$$\frac{\mathrm{d}g_j}{\mathrm{d}t} = g_{j+M}$$

for $j = 0, \ldots, (M+1)(P-1)$. Here r_v denotes the number in (30).

Remark 3.3. Here the evolution of moments is discussed for completeness. However, no time is involved in the following decoding problem since there is no need for the time evolution.

4. Decoding as a moment problem related to extended Lotka–Volterra lattice

In this section, we show that the moment problem related to the extended Lotka–Volterra in the previous section could be used to design a decoding algorithm for multiple codewords. The decoding process is based on a similar diagram to (7), given by

$$\begin{array}{ccccc} \{a_{(M+1)k+m}\} & \overset{(21)}{\longrightarrow} & \{\rho_{p,v}, \lambda_p^{M+1}\} & \longrightarrow & \{e_{j,v}, \alpha^j\} \\ (16)\uparrow & & \uparrow\downarrow (25) & & \uparrow \text{ goal} \\ \{G_k^{(m)}\} & \underset{(14)}{\longleftarrow} & \{g_{Mu+v-1}\} & \longleftarrow & \{S_{u,v}/S_{0,v}\} \end{array} \tag{29}$$

for $k = 0, \ldots, P-1$, $m = 0, \ldots, M$, $p = 1, \ldots, P$, $u = 0, \ldots, r_v - 1$, $v = 1, \ldots, M$, where $n = (M+1)P$, and r_v denotes the number in (30).

4.1. Relation between coding theory and moment problem

Suppose that one sent M codes with sent codewords $c_v = (c_{0,v}, c_{1,v}, \ldots, c_{N-1,v})$, the received codewords $b_v = (b_{0,v}, b_{1,v}, \ldots, b_{N-1,v})$ and the errors $e_v = (e_{0_v}, e_{1,v}, \ldots, e_{N-1,v})$ for $v = 1, 2, \ldots, M$.

In the BCH code case, define the syndromes as

$$S_{u,v} = \sum_{j=0}^{N-1} b_{j,v}\alpha^{j(u+l_v)} = \sum_{j\in J} e_{j,v}\alpha^{j(u+l_v)}, \quad u = 0, 1, \ldots, r_v - 1.$$

Let $J = \{j | \exists v, \mathrm{e}_{j,v} \neq 0\}$ be the positions where at least one error occurs and P be the number of elements in J. We connect the syndromes with moments by setting

$$\rho_{j,v} = \begin{cases} \frac{e_{j,v}\alpha^{jl_v}}{(M+1)\sum_{j'\in J} e_{j',v}\alpha^{j'l_v}}, & \text{for } j \in J, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$\lambda_p^{M+1} = \begin{cases} \alpha^j, & j \in J, \\ 0, & \text{otherwise}, \end{cases}$$

which leads to

$$S_{u,v} = \sum_{j\in J} e_{j,v}\alpha^{j(u+l_v)} = g_{Mu+v-1}$$

and

$$h_{Mu+v-1} = \sum_{p=1}^{M} (M+1)\rho_{p,v}\lambda_p^{(M+1)u} = \frac{\sum_{j\in J} e_{j,v}\alpha^{j(u+l_v)}}{\sum_{j'\in J} e_{j',v}\alpha^{j'l_v}} = \frac{S_{u,v}}{S_{0,v}}.$$

Define the error-locator polynomial as $\sigma(x) = \prod_{j\in J}\left(x - \alpha^j\right)$ and the error-value polynomial of the vth code as $\omega_v(x) = \sum_{j\in J} e_{j,v}\alpha^{jl_v}\prod_{k\in J\setminus\{j\}}(x - \alpha^k)$.

Then, we have

$$\frac{\omega_v(x)}{\sigma(x)} = \sum_{j\in J}\frac{e_{j,v}\alpha^{jl_v}}{x - \alpha^j} = \sum_{u=0}^{\infty}\frac{S_{u,v}}{x^{u+1}} = \frac{S_{0,v}f_v(z)}{z^M},$$

where the rational function $f_v(z)$ defined in (26) and $x = z^{M+1}$. Hence, we can get errors by calculating the partial fraction expansion of $f_v(z)$ produced by L as (21), where L is constructed by the a_j defined by (16).

In the Goppa code case, the corresponding syndromes can be expressed as

$$S_{u,v} = \sum_{j=0}^{N-1} M(\alpha_j)^{-1} b_{j,v} \alpha_j^u = \sum_{j\in J} M(\alpha_j)^{-1} e_{j,v} \alpha_j^u, \quad u = 0, 1, \ldots, r_v - 1.$$

Define the error-locator polynomial as $\sigma(x) = \prod_{j\in J}(x - \alpha_j)$ and the error-value polynomial of the vth code as $\omega_v(x) = \sum_{j\in J} M(\alpha_j)^{-1} e_{j,v} \prod_{k\in J\setminus\{j\}}(x - \alpha_k)$. Then, by setting

$$\rho_{p,v} = \begin{cases} \frac{M(\alpha_j)^{-1} e_{j,v} \exp(\alpha_j t)}{(M+1)\sum_{j'\in J} M(\alpha_{j'})^{-1} e_{j',v} \exp(\alpha_{j'} t)}, & j \in J, \\ 0, & \text{otherwise}, \end{cases}$$

and

$$\lambda_p^{M+1} = \begin{cases} \alpha_j, & j \in J, \\ 0, & \text{otherwise}, \end{cases}$$

we get $g_{Mu+v-1} = S_{u,v}$, $h_{Mu+v-1} = \frac{S_{u,v}}{S_{0,v}}$ and

$$\frac{\omega_v(x)}{\sigma(x)} = \sum_{j\in J} \frac{M_v(\alpha_j)^{-1} e_{j,v}}{x - \alpha_j} = \sum_{u=0}^{\infty} \frac{S_{u,v}}{x^{u+1}} = \frac{S_{0,v} f_v(z)}{z^M}.$$

The above calculation suggests that the multiple BCH-Goppa decoding problem may be formally solved according to the diagram (29).

It is also necessary to know the number of syndromes needed for decoding. As is known, we need to construct the matrix L of size $(M+1)P \times (M+1)P$ and calculate

$$\frac{\omega_v(x)}{\sigma(x)} = \frac{S_{0,v} f_v(z)}{z^M},$$

where $x = z^{M+1}$. The last a_j involved in L is

$$a_{(M+1)(P-1)+M} = \frac{G_P^{(M)} G_{P-1}^{(M-1)}}{G_P^{(M-1)} G_{P-1}^{(M)}},$$

where the last g_j to be involved is the last element of

$$G_P^{(M)} = \begin{vmatrix} g_M & g_{M+1} & \cdots & g_{M+P-1} \\ g_{2M} & g_{2M+1} & \cdots & g_{2M+P-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{PM} & g_{PM+1} & \cdots & g_{PM+P-1} \end{vmatrix}.$$

Therefore, if there exist P' and Q' such that

$$P' = \left\lfloor \frac{P-1}{M} \right\rfloor, \quad P - 1 = P'M + Q',$$

we get the last element

$$g_{PM+P-1} = g_{(P+P')M+Q'} = S_{P+P',Q'+1},$$

which means the minimum number of the needed syndromes $\{S_{0,v}, S_{1,v}, \ldots, S_{r_v-1,v}\}$ is

$$r_v - 1 = \begin{cases} P + \left\lfloor \frac{P-1}{M} \right\rfloor & \text{for } v = 1, 2, \ldots, P - M\left\lfloor \frac{P-1}{M} \right\rfloor, \\ P + \left\lfloor \frac{P-1}{M} \right\rfloor - 1 & \text{for } v = P - M\left\lfloor \frac{P-1}{M} \right\rfloor + 1, \ldots, M. \end{cases} \tag{30}$$

4.2. Decoding algorithm for multiple BCH codes

For decoding M sets of received BCH codes, we may design the decoding algorithm according to four major steps.

(i) For M sets of known syndromes $\{S_{0,v}, S_{1,v}, \ldots, S_{r_v-1,v}\}$ over the finite field $\mathrm{GF}(q^u)$, where $v = 1, 2, \ldots, M$, let

$$g_{Mu+v-1} = S_{u,v}, \quad u = 0, 1, \ldots, r_v - 1.$$

Calculate the determinants

$$G_0^{(m)} = 1, \quad G_1^{(m)} = g_m,$$

$$G_k^{(m)} = \begin{vmatrix} g_m & g_{m+1} & \cdots & g_{m+k-1} \\ g_{m+M} & g_{m+M+1} & \cdots & g_{m+M+k-1} \\ \vdots & \vdots & & \vdots \\ g_{m+M(k-1)} & g_{m+M(k-1)+1} & \cdots & g_{m+(M+1)(k-1)} \end{vmatrix}$$

for

$$m = 0, 1, \ldots, M, \quad k = 2, 3, \ldots, P.$$

(ii) Introduce the matrix of $(M+1)P \times (M+1)P$

$$L = \begin{pmatrix} \overbrace{0 \quad \cdots \quad 0}^{M} & 1 & & & \\ a_1 & 0 \quad \cdots 0 & 1 & & \\ & a_2 \quad \ddots & \ddots & \ddots & \\ & & \ddots \ddots & \ddots & 1 \\ & & \ddots & & 0 \\ & & & \ddots & \ddots \quad \vdots \\ & & & a_{(M+1)(P-1)+M} & 0 \end{pmatrix},$$

where

$$a_{(M+1)k+m} = \begin{cases} \frac{G_{k+1}^{(0)} G_{k-1}^{(M)}}{G_k^{(0)} G_k^{(M)}} & \text{for } m = 0, \\ \frac{G_{k+1}^{(m)} G_k^{(m-1)}}{G_{k+1}^{(m-1)} G_k^{(m)}} & \text{for } m = 1, \ldots, M \end{cases}$$

for $k = 0, 1, \ldots, P-1$.

(iii) Compute the rational function of the vth row and vth column element of $(zI - L)^{-1}$

$$f_v(z) = E_v^{\top}(zI - L)^{-1}E_v, \quad E_v = (\overbrace{0, \ldots, 0}^{v-1}, 1, 0, \ldots, 0)^{\top}.$$

(iv) Factor the polynomials $\sigma(x)$ and $w(x)$ and calculate the partial fraction expansions

$$\frac{S_{0,v}f_v(z)}{z^M} \triangleq \frac{\omega_v(x)}{\sigma(x)} = \sum_{j\in J}\frac{e_{j,v}\alpha^{jl_v}}{x - \alpha^j}, \tag{31}$$

where $x = z^{M+1}$, to get the errors.

Remark 4.1. All of the calculations are over the finite field $\mathrm{GF}(q^u)$.

Remark 4.2. As for M received Goppa codes, the similar decoding could be implemented by replacing (31) with

$$\frac{S_{0,v}f_v(z)}{z^M} \triangleq \frac{\omega_v(x)}{\sigma(x)} = \sum_{j\in J}\frac{M_v(\alpha_j)^{-1}e_{j,v}}{x - \alpha_j}.$$

Remark 4.3. It is noted that some of the denominators appearing in $a_{(M+1)k+m}$ in step (ii) may be equal to zero. If

$$G_1^{(m)} \neq 0,\ G_2^{(m)} \neq 0,\ \cdots, G_{k-1}^{(m)} \neq 0,$$

for all $0 \leqslant m \leqslant M$, and $G_k^{(l)} = 0$ for some $0 \leqslant l \leqslant M$, the number of error can be decoded by this algorithm becomes $k-1$. However, sometimes, $G_k^{(m)} = 0$ is avoidable by changing the orders of codes. For example, set the syndrome sequence of the second code to be $\{S_{0,1}, S_{1,1}, \ldots, S_{r-1,1}\}$ and set the syndrome sequence of the first code to be $\{S_{0,2}, S_{1,2}, \ldots, S_{r-1,2}\}$.

Remark 4.4. Our decoding algorithm for multiple BCH-Goppa codes might be more efficient than running the decoding algorithm in [2] for a single code multiple times. When the errors happen at the same locations, comparing with the P-error decoding for the BCH-Goppa codes by M times, the number of syndromes we need to know becomes $PM + P$ rather than $2MP$, and the number of P-order determinants needed for calculation in step (i) becomes $M + 1$ rather than $2M$. This means that our decoding algorithm for multiple BCH-Goppa codes is of less condition and lower complexity than running the decoding algorithm in [2] for a single code multiple times. However, if the intersection of the error location sets of any two codes is empty, their algorithm has a better performance. In this case, the number of the required syndromes in our algorithm is $MP + P$, while $2P$ in their algorithm, where $P = \sum_{j=1}^{M} P_j$ and P_j denotes the number of errors in the jth codewords. And, in step (i) we need to compute P-order determinants, where $P = \sum_{j=1}^{M} P_j$, while in [2] the highest order of determinants is only $\max\{P_j, j = 1, 2, \ldots, M\}$.

4.3. Decoding examples

One special case is $M = 1$, that is one P-error BCH-Goppa decoding. In this case we have $r - 1 = P + \left\lfloor \frac{P-1}{M} \right\rfloor = 2P - 1$ and

$$g_k = S_k, \quad k = 0, 1, \ldots, 2P - 1.$$

The determinants turn into

$$G_0^{(m)} = 1, \quad G_1^{(m)} = S_m,$$

$$G_k^{(m)} = \begin{vmatrix} S_m & S_{m+1} & \cdots & S_{m+k-1} \\ S_{m+M} & S_{m+2} & \cdots & S_{m+k} \\ \vdots & \vdots & & \vdots \\ S_{m+k-1} & S_{m+k} & \cdots & S_{m+2(k-1)} \end{vmatrix},$$

$$m = 0, 1, \quad k = 2, 3, \ldots, P.$$

The elements in Jacobi matrix L can be computed by

$$\begin{cases} a_{2k} &= \frac{G_{k+1}^{(0)} G_{k-1}^{(1)}}{G_k^{(0)} G_k^{(1)}}, \\ a_{2k+1} &= \frac{G_{k+1}^{(1)} G_k^{(0)}}{G_{k+1}^{(0)} G_k^{(1)}}, \end{cases}$$

for $k = 0, 1, \ldots, P - 1$. Then, the error can be found by factoring $\sigma(x)$ and $w(x)$ obtained by the rational function $f_v(z)$

$$\frac{S_{0,v} f_v(z)}{z^M} \triangleq \frac{\omega_v(x)}{\sigma(x)} = \sum_{j \in J} \frac{e_{j,v} \alpha^{jl}}{x - \alpha^j}, \quad x = z^2.$$

An example is shown as below to demonstrate this decoding process.

Example 4.1 (BCH code, 3-error decoding, $M = 1, q = 2, u = 4, N = 15, l = 1, r = 6$). Suppose $\alpha \in \mathrm{GF}(2^4)$ is the root of the irreducible polynomial $x^4 + x + 1$ over $\mathrm{GF}(2)$, which results in $\alpha^{15} = 1$. Consider the extension field

$$\mathrm{GF}(2^4) = \{\gamma_0 + \gamma_1 \alpha + \gamma_2 \alpha^2 + \gamma_3 \alpha^3, \gamma_j \in \mathrm{GF}(2), j = 0, 1, 2, 3\} = (\gamma_0, \gamma_1, \gamma_2, \gamma_3).$$

It is not hard to see that there exists a one-to-one map from the cyclic group generated by α to $\mathrm{GF}(2^4)$:

$$\begin{array}{llll} \alpha^0 = (1,0,0,0), & \alpha = (0,1,0,0), & \alpha^2 = (0,0,1,0), & \alpha^3 = (0,0,0,1), \\ \alpha^4 = (1,1,0,0), & \alpha^5 = (0,1,1,0), & \alpha^6 = (0,0,1,1), & \alpha^7 = (1,1,0,1), \\ \alpha^8 = (1,0,1,0), & \alpha^9 = (0,1,0,1), & \alpha^{10} = (1,1,1,0), & \alpha^{11} = (0,1,1,1), \\ \alpha^{12} = (1,1,1,1), & \alpha^{13} = (1,0,1,1), & \alpha^{14} = (1,0,0,1). & \end{array}$$

Note that this map will be helpful for simplifying the calculations.

By considering the minimal polynomials $m_i(x)$ of α^i for $i = 1, 2, \ldots, 6$, then we obtain a polynomial

$$G(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

as the least common multiple of $m_1(x), m_2(x), \ldots, m_6(x)$. Thus we construct a BCH code with the generator polynomial $G(x)$. This example is based on a sent code

$$c = (1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0).$$

When we receive a codeword

$$b = (1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0),$$

we can calculate syndromes

$$\{S_0, S_1, S_2, S_3, S_4, S_5\} = \{1, 1, \alpha^{10}, 1, \alpha^{10}, \alpha^5\}.$$

Then the determinants are obtained as follows:

$$\begin{array}{llll} G_0^{(0)} = 1, & G_1^{(0)} = 1, & G_2^{(0)} = \alpha^5, & G_3^{(0)} = 1, \\ G_0^{(1)} = 1, & G_1^{(1)} = 1, & G_2^{(1)} = \alpha^{10}, & G_3^{(1)} = \alpha^5, \end{array}$$

which results in the tridiagonal matrix L being

$$L = \begin{pmatrix} 0 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & \alpha^5 & 0 & 1 & & \\ & & \alpha^5 & 0 & 1 & \\ & & & 1 & 0 & 1 \\ & & & & 1 & 0 \end{pmatrix}.$$

By calculating the $(1, 1)$th element of $(\lambda I - L)^{-1}$, we obtain

$$f_1(z) = \frac{z^5 + \alpha^5 z}{z^6 + z^4 + \alpha^5}$$

so that by setting $x = z^2$ we have

$$\frac{\omega_1(x)}{\sigma(x)} \triangleq S_0 \frac{f_1(z)}{z} = \frac{x^2 + \alpha^5}{x^3 + x^2 + \alpha^5} = \frac{\alpha^3}{x - \alpha^3} + \frac{\alpha^5}{x - \alpha^5} + \frac{\alpha^{12}}{x - \alpha^{12}}.$$

Eventually, the error is found, i.e. $e = (0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0)$.

Next, we give an example of the algorithm with $M = 2$ for multiple BCH codes. The relationship becomes

$$\begin{array}{ccccc} \{a_{3k+m}\} & \longrightarrow & \{\rho_{p,v}, \lambda_p^3\} & \longrightarrow & \{e_{j,v}, \alpha^j\} \\ \uparrow & & \updownarrow & & \uparrow \textit{goal} \\ \{G_k^{(m)}\} & \longleftarrow & \{h_{2u+v-1}\} & \longleftarrow & \{S_k^v / S_0^v\} \end{array}$$

for $v = 1, 2$ and $m = 0, 1, 2$. The elements in Jacobi matrix can be computed by

$$\begin{cases} a_{3k} & = & \frac{G_{k+1}^{(0)} G_{k-1}^{(2)}}{G_k^{(0)} G_k^{(2)}}, \\ a_{3k+1} & = & \frac{G_{k+1}^{(1)} G_k^{(0)}}{G_{k+1}^{(0)} G_k^{(1)}}, \\ a_{3k+2} & = & \frac{G_{k+1}^{(2)} G_k^{(1)}}{G_{k+1}^{(1)} G_k^{(2)}} \end{cases}$$

for $k = 0, 1, \ldots, P-1$.

Example 4.2 (BCH code, 3-error decoding, $M = 2, q = 3, u = 3, N = 13, l_1 = 1, l_2 = 7, r_1 = 5, r_2 = 4$). Suppose $\alpha \in \mathrm{GF}(3^3)$ is a root of the irreducible polynomial $x^3 + x^2 + x + 2$ over GF(3), then we obtain $\alpha^{13} = 1$ and the extension field

$$\mathrm{GF}(3^3) = \{\gamma_0 + \gamma_1\alpha + \gamma_2\alpha^2, \gamma_j \in \mathrm{GF}(3), j = 0, 1, 2\}.$$

Every element of the cyclic group generated by α can be one-to-one mapped into an element $(\gamma_0, \gamma_1, \gamma_2)$ in $\mathrm{GF}(3^3)$:

$$\begin{array}{llll} \alpha^0 = (1,0,0), & \alpha^1 = (0,1,0), & \alpha^2 = (0,0,1), & \alpha^3 = (1,2,2), \\ \alpha^4 = (2,2,0), & \alpha^5 = (0,2,2), & \alpha^6 = (2,1,0), & \alpha^7 = (0,2,1), \\ \alpha^8 = (1,2,1), & \alpha^9 = (1,0,1), & \alpha^{10} = (1,0,2), & \alpha^{11} = (2,2,1), \\ \alpha^{12} = (1,1,1). & & & \end{array}$$

Let us consider two BCH codes with the generator polynomials

$$G_1(x) = x^9 + x^8 + 2x^7 + x^5 + 2x^3 + 2x^2 + 2$$

satisfying $G_1(\alpha) = G_1(\alpha^2) = G_1(\alpha^3) = G_1(\alpha^4) = G_1(\alpha^5)$ and

$$G_2(x) = x^9 + x^7 + x^6 + 2x^4 + x^2 + 2x + 2$$

satisfying $G_2(\alpha^7) = G_2(\alpha^8) = G_2(\alpha^9) = G_2(\alpha^{10}) = 0$. And the two sent codes are

$$c_1 = (2, 0, 2, 2, 0, 1, 0, 2, 1, 1, 0, 0, 0)$$

and

$$c_2 = (2, 2, 1, 0, 2, 0, 1, 1, 0, 1, 0, 0, 0).$$

Assume that the two codewords we received are respectively

$$b_1 = (2, 1, 2, 2, 2, 1, 0, 2, 1, 0, 0, 0, 0)$$

and

$$b_2 = (2, 1, 1, 0, 0, 0, 1, 1, 0, 2, 0, 0, 0).$$

By calculation, we obtain the syndromes

$$\begin{array}{llll} S_{0,1} = \alpha^5, & S_{1,1} = \alpha^{11}, & S_{2,1} = \alpha^2, & S_{3,1} = 2\alpha^2, \quad S_{4,1} = \alpha^8, \\ S_{0,2} = 2\alpha^{10}, & S_{1,2} = 2\alpha^4, & S_{2,2} = 2\alpha^6, & S_{3,2} = \alpha^5, \end{array}$$

resulting in

$$\begin{array}{llll} g_0 = \alpha^5, & g_2 = \alpha^{11}, & g_4 = \alpha^2, & g_6 = 2\alpha^2, \quad g_8 = \alpha^8, \\ g_1 = 2\alpha^{10}, & g_3 = 2\alpha^4, & g_5 = 2\alpha^6, & g_7 = \alpha^5. \end{array}$$

Then we have the determinants

$$\begin{array}{llll} G_0^{(0)} = 1, & G_1^{(0)} = \alpha^5, & G_2^{(0)} = 2\alpha, & G_3^{(0)} = \alpha^9, \\ G_0^{(1)} = 1, & G_1^{(1)} = 2\alpha^{10}, & G_2^{(1)} = \alpha^4, & G_3^{(1)} = 2\alpha^7, \\ G_0^{(2)} = 1, & G_1^{(2)} = \alpha^{11}, & G_2^{(2)} = 2\alpha, & G_3^{(2)} = \alpha^{10}. \end{array}$$

and the matrix

$$L = \begin{pmatrix} 0 & 0 & 1 & & & & & & \\ 2\alpha^5 & 0 & 0 & 1 & & & & & \\ & 2\alpha & 0 & 0 & 1 & & & & \\ & & 2\alpha^{11} & 0 & 0 & 1 & & & \\ & & & \alpha^{11} & 0 & 0 & 1 & & \\ & & & & \alpha^9 & 0 & 0 & 1 & \\ & & & & & \alpha^5 & 0 & 0 & 1 \\ & & & & & & \alpha^8 & 0 & 0 \\ & & & & & & & \alpha^6 & 0 \end{pmatrix}.$$

The corresponding rational functions $f_v(z)$ are

$$f_1(z) = \frac{z^8 + 2\alpha^8 z^5 + 2\alpha^9 z^2}{z^9 + 2\alpha^2 z^6 + \alpha^{11} z^3 + 2\alpha}, \quad f_2(z) = \frac{z^8 + 2\alpha z^5 + \alpha^{11} z^2}{z^9 + 2\alpha^2 z^6 + \alpha^{11} z^3 + 2\alpha}$$

and by setting $x = z^3$ we get

$$\frac{\omega_1(x)}{\sigma(x)} \triangleq S_{0,1}\frac{f_1(z)}{z^2} = \frac{\alpha^5(x^2 + 2\alpha^8 x + 2\alpha^9)}{x^3 + 2\alpha^2 x^2 + \alpha^{11} x + 2\alpha} = \frac{\alpha}{x-\alpha} + \frac{2\alpha^4}{x-\alpha^4} + \frac{2\alpha^9}{x-\alpha^9},$$

$$\frac{\omega_2(x)}{\sigma(x)} \triangleq S_{0,2}\frac{f_2(z)}{z^2} = \frac{2\alpha^{10}(x^2 + 2\alpha x + \alpha^{11})}{x^3 + 2\alpha^2 x^2 + \alpha^{11} x + 2\alpha} = \frac{2\alpha^7}{x-\alpha} + \frac{\alpha^2}{x-\alpha^4} + \frac{\alpha^{11}}{x-\alpha^9}.$$

Thus, the error of the first code is

$$e_1 = (0, 1, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0)$$

and the error of the second code is

$$e_2 = (0, 2, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0).$$

It follows from $c = b - e$ that original codes are

$$c_1 = (2, 0, 2, 2, 0, 1, 0, 2, 1, 1, 0, 0, 0)$$

and

$$c_2 = (2, 2, 1, 0, 2, 0, 1, 1, 0, 1, 0, 0, 0)$$

respectively.

The method can also be used to decode a Goppa code and a RS code. Some examples are given for $M = 3$. The relationship becomes

$$\begin{array}{ccccc} \{a_{4k+m}\} & \longrightarrow & \{\rho_{p,v}, \lambda_p^4\} & \longrightarrow & \{e_j^v, \alpha^j\} \\ \uparrow & & \updownarrow & & \uparrow goal \\ \{G_k^{(m)}\} & \longleftarrow & \{h_{3u+v-1}\} & \longleftarrow & \{S_k^v/S_0^v\} \end{array}$$

for $v = 1, 2, 3$ and $m = 0, 1, 2, 3$. The elements in the matrix L can be computed according to

$$\begin{cases} a_{4k} = \frac{G_{k+1}^{(0)} G_{k-1}^{(3)}}{G_k^{(0)} G_k^{(3)}}, & a_{4k+1} = \frac{G_{k+1}^{(1)} G_k^{(0)}}{G_{k+1}^{(0)} G_k^{(1)}}, \\ a_{4k+2} = \frac{G_{k+1}^{(2)} G_k^{(1)}}{G_{k+1}^{(1)} G_k^{(2)}}, & a_{4k+3} = \frac{G_{k+1}^{(3)} G_k^{(2)}}{G_{k+1}^{(2)} G_k^{(3)}}, \end{cases}$$

for $k = 0, 1, \ldots, P-1$.

Example 4.3 (Goppa code, 2-error decoding, $M = 3, q = 3, u = 3, N = 13$, $r_1 = 3, r_2 = 3, r_3 = 2$). Suppose $\alpha \in \mathrm{GF}(3^3)$ is the root of the irreducible polynomial $x^3 + x^2 + x + 2$ over $\mathrm{GF}(3)$ and consider the extension field

$$\mathrm{GF}(3^3) = \{\gamma_0 + \gamma_1\alpha + \gamma_2\alpha^2, \gamma_j \in \mathrm{GF}(3), j = 0, 1, 2\}.$$

As indicated in the above example, we have $\alpha^{13} = 1$ and $\alpha_i, i = 1, 2 \ldots, 12$ are mutually distinct.

Consider three Goppa codes $\Gamma(\mathcal{L}, M_1)$, $\Gamma(\mathcal{L}, M_2)$ and $\Gamma(\mathcal{L}, M_3)$ over the same finite field $\mathrm{GF}(3^3)$, where $\mathcal{L} = \{1, \alpha, \ldots, \alpha^{12}\}$ with the Goppa polynomials

$$M_1(z) = z^4 + z + 2, \quad M_2(z) = z^4 + 1, \quad M_3(z) = z^4 + 2z + 1.$$

If three codewords we sent are

$$c_1 = (2, 0, 2, 0, 2, 2, 2, 2, 2, 0, 2, 2, 2),$$

$$c_2 = (1, 2, 0, 2, 1, 0, 0, 2, 2, 2, 1, 2, 1),$$

$$c_3 = (1, 1, 1, 1, 1, 0, 1, 0, 0, 2, 2, 1, 2)$$

and the received codewords are respectively

$$b_1 = (2, 0, 2, 0, 2, 2, 2, 2, 2, 0, 1, 2, 1),$$

$$b_2 = (1, 2, 0, 2, 1, 0, 0, 2, 2, 2, 2, 2, 2),$$

$$b_3 = (1, 1, 1, 1, 1, 0, 1, 0, 0, 2, 1, 1, 0).$$

We compute the syndromes by using the received codewords

$$\begin{array}{lll} S_{0,1} = 2\alpha^3, & S_{1,1} = \alpha^{10}, & S_{2,1} = 2, \\ S_{0,2} = 2\alpha^4, & S_{1,2} = \alpha^3, & S_{2,2} = 2\alpha^{10}, \\ S_{0,3} = 1, & S_{1,3} = 2\alpha^{11}, & \end{array}$$

and then we have

$$\begin{array}{lll} g_0 = 2\alpha^3, & g_3 = \alpha^{10}, & g_6 = 2, \\ g_1 = 2\alpha^4, & g_4 = \alpha^3, & g_7 = 2\alpha^{10}, \\ g_2 = 1, & g_5 = 2\alpha^{11}. & \end{array}$$

The determinants are

$$\begin{array}{lll} G_0^{(0)} = 1, & G_1^{(0)} = 2\alpha^3, & G_2^{(0)} = 1, \\ G_0^{(1)} = 1, & G_1^{(1)} = 2\alpha^4, & G_2^{(1)} = 2\alpha^8, \\ G_0^{(2)} = 1, & G_1^{(2)} = 1, & G_2^{(2)} = \alpha^7, \\ G_0^{(3)} = 1, & G_1^{(3)} = \alpha^{10}, & G_2^{(3)} = \alpha^9, \end{array}$$

so that the matrix L is obtained

$$L = \begin{pmatrix} 0 & 0 & 0 & 1 & & & & \\ \alpha & 0 & 0 & 0 & 1 & & & \\ & 2\alpha^9 & 0 & 0 & 0 & 1 & & \\ & & \alpha^{10} & 0 & 0 & 0 & 1 & \\ & & & 2 & 0 & 0 & 0 & 1 \\ & & & & 2\alpha^7 & 0 & 0 & 0 \\ & & & & & \alpha^3 & 0 & 0 \\ & & & & & & \alpha^5 & 0 \end{pmatrix}.$$

Next, we compute the rational functions

$$f_1(z) = \frac{z^7 + \alpha^{10} z^3}{z^8 + 2\alpha^6 z^4 + \alpha^9},$$

$$f_2(z) = \frac{z^7 + 2\alpha^7 z^3}{z^8 + 2\alpha^6 z^4 + \alpha^9},$$

$$f_3(z) = \frac{z^7 + 2\alpha^9 z^3}{z^8 + 2\alpha^6 z^4 + \alpha^9}.$$

Let $x = z^4$ and calculate the partial fraction expansions

$$\frac{\omega_1(x)}{\sigma(x)} \triangleq S_{0,1} \frac{f_1(z)}{z^3} = \frac{2\alpha^3(x + \alpha^{10})}{x^2 + 2\alpha^6 x + \alpha^9} = \frac{\alpha^6}{x - \alpha^{10}} + \frac{\alpha^2}{x - \alpha^{12}},$$

$$\frac{\omega_2(x)}{\sigma(x)} \triangleq S_{0,2} \frac{f_2(z)}{z^3} = \frac{2\alpha^4(x + 2\alpha^7)}{x^2 + 2\alpha^6 x + \alpha^9} = \frac{2\alpha^9}{x - \alpha^{10}} + \frac{2\alpha^3}{x - \alpha^{12}},$$

$$\frac{\omega_3(x)}{\sigma(x)} \triangleq S_{0,3} \frac{f_3(z)}{z^3} = \frac{(x + 2\alpha^9)}{x^2 + 2\alpha^6 x + \alpha^9} = \frac{\alpha^8}{x - \alpha^{10}} + \frac{2\alpha^7}{x - \alpha^{12}}.$$

Since

$$M_1(\alpha^{10}) = 2\alpha^7,\ M_1(\alpha^{12}) = 2\alpha^{11},\ M_2(\alpha^{10}) = 2\alpha^4,$$

$$M_2(\alpha^{12}) = 2\alpha^{10},\ M_3(\alpha^{10}) = 2\alpha^5,\ M_3(\alpha^{12}) = 2\alpha^6,$$

we eventually get error values

$$e_{10,1} = 2,\ e_{12,1} = 2,\ e_{10,2} = 1,\ e_{12,2} = 1,\ e_{10,3} = 2,\ e_{12,3} = 1,$$

leading to

$$e_1 = \{0,0,0,0,0,0,0,0,0,0,2,0,2\},$$

$$e_2 = \{0,0,0,0,0,0,0,0,0,0,1,0,1\},$$

$$e_3 = \{0,0,0,0,0,0,0,0,0,0,2,0,1\}.$$

Example 4.4 (RS code, 3-error decoding, $M = 3$, $q = 2^4, u = 1, N = 15, r_1 = 4, r_2 = 4, r_3 = 4, l_1 = 1, l_2 = 5, l_3 = 14$). Consider the extension field

$$\mathrm{GF}(2^4) = \{\gamma_0 + \gamma_1\alpha + \gamma_2\alpha^2 + \gamma_3\alpha^3, \gamma_j \in \mathrm{GF}(2), j = 0,1,2,3\} = (\gamma_0, \gamma_1, \gamma_2, \gamma_3),$$

where $\alpha \in \mathrm{GF}(2^4)$ is the root of the irreducible polynomial $x^4 + x + 1$ over $\mathrm{GF}(2)$. Recall that we have $\alpha^{15} = 1$.

Assume that we have three generator polynomials

$$G_1(x) = \prod_{i=1}^{4}(x - \alpha^i) = x^4 + \alpha^{13}x^3 + \alpha^6x^2 + \alpha^3x + \alpha^{10},$$

$$G_2(x) = \prod_{i=5}^{8}(x - \alpha^i) = x^4 + \alpha^2x^3 + \alpha^{14}x^2 + x + \alpha^{11},$$

$$G_3(x) = \prod_{i=14}^{17}(x - \alpha^i) = x^4 + \alpha^{11}x^3 + \alpha^2x^2 + \alpha^{12}x + \alpha^2,$$

and the sent codewords are

$$c_1 = (\alpha^{10}, \alpha^3, \alpha^6, \alpha^{13}, 1,0,0,0,0,0,0,0,0,0,0),$$
$$c_2 = (\alpha^{11}, 1, \alpha^{14}, \alpha^2, 1,0,0,0,0,0,0,0,0,0,0),$$
$$c_3 = (\alpha^2, \alpha^{12}, \alpha^2, \alpha^{11}, 1,0,0,0,0,0,0,0,0,0,0).$$

If the received words are

$$b_1 = (\alpha^9, \alpha^3, \alpha^5, \alpha^{10}, 1,0,0,0,0,0,0,0,0,0,0),$$
$$b_2 = (1, 1, \alpha, 0, 1,0,0,0,0,0,0,0,0,0,0),$$
$$b_3 = (\alpha^9, \alpha^{12}, \alpha^7, \alpha^7, 1,0,0,0,0,0,0,0,0,0,0),$$

then the syndromes known from remainder are

$$\begin{array}{llll} S_{0,1}=\alpha^6, & S_{1,1}=1, & S_{2,1}=\alpha^2, & S_{3,1}=\alpha^8, \\ S_{0,2}=\alpha^{12}, & S_{1,2}=\alpha^9, & S_{2,2}=\alpha^5, & S_{3,2}=\alpha^2, \\ S_{0,3}=\alpha^{12}, & S_{1,3}=\alpha^2, & S_{2,3}=\alpha^{14}, & S_{3,3}=\alpha^8. \end{array}$$

It follows that g_j are

$$\begin{array}{llll} g_0=\alpha^6, & g_3=1, & g_6=\alpha^2, & g_9=\alpha^8, \\ g_1=\alpha^{12}, & g_4=\alpha^9, & g_7=\alpha^5, & g_{10}=\alpha^2, \\ g_2=\alpha^{12}, & g_5=\alpha^2, & g_8=\alpha^{14}, & g_{11}=\alpha^8, \end{array}$$

and the determinants are

$$\begin{array}{llll} G_0^{(0)}=1, & G_1^{(0)}=\alpha^6, & G_2^{(0)}=\alpha^{11}, & G_3^{(0)}=\alpha^{14}, \\ G_0^{(1)}=1, & G_1^{(1)}=\alpha^{12}, & G_2^{(1)}=\alpha^8, & G_3^{(1)}=\alpha, \\ G_0^{(2)}=1, & G_1^{(2)}=\alpha^{12}, & G_2^{(2)}=\alpha^{13}, & G_3^{(2)}=1, \\ G_0^{(3)}=1, & G_1^{(3)}=1, & G_2^{(3)}=\alpha^3, & G_3^{(3)}=\alpha^4. \end{array}$$

Next, by calculating a_j one can obtain the matrix L

$$L=\begin{pmatrix} 0 & 0 & 0 & 1 & & & & & & & & \\ \alpha^6 & 0 & 0 & 0 & 1 & & & & & & & \\ & 1 & 0 & 0 & 0 & 1 & & & & & & \\ & & \alpha^3 & 0 & 0 & 0 & 1 & & & & & \\ & & & \alpha^5 & 0 & 0 & 0 & 1 & & & & \\ & & & & \alpha^6 & 0 & 0 & 0 & 1 & & & \\ & & & & & \alpha^5 & 0 & 0 & 0 & 1 & & \\ & & & & & & \alpha^2 & 0 & 0 & 0 & 1 & \\ & & & & & & & 1 & 0 & 0 & 0 & 1 \\ & & & & & & & & \alpha^5 & 0 & 0 & 0 \\ & & & & & & & & & \alpha^9 & 0 & 0 \\ & & & & & & & & & & \alpha^{14} & 0 \end{pmatrix}.$$

The rational functions $f_v(z)$ are

$$f_1(z)=\frac{z^{11}+\alpha^{10}z^7+\alpha^{12}z^3}{z^{12}+\alpha^{13}z^8+\alpha^9z^4+\alpha^5},$$

$$f_2(z)=\frac{z^{11}+\alpha z^7+\alpha^3z^3}{z^{12}+\alpha^{13}z^8+\alpha^9z^4+\alpha^5},$$

$$f_3(z)=\frac{z^{11}+\alpha^7z^7+\alpha^5z^3}{z^{12}+\alpha^{13}z^8+\alpha^9z^4+\alpha^5}.$$

By setting $x=z^4$ we get

$$\frac{\omega_1(x)}{\sigma(x)}\triangleq S_{0,1}\frac{f_1(z)}{z^3}=\frac{\alpha^6(x^2+\alpha^{10}x+\alpha^{12})}{x^3+\alpha^{13}x^2+\alpha^9x+\alpha^5}=\frac{\alpha^{13}}{x-1}+\frac{\alpha^{11}}{x-\alpha^2}+\frac{\alpha^{12}}{x-\alpha^3},$$

$$\frac{\omega_2(x)}{\sigma(x)} \triangleq S_{0,2}\frac{f_2(z)}{z^3} = \frac{\alpha^{12}(x^2+\alpha x+\alpha^3)}{x^3+\alpha^{13}x^2+\alpha^9 x+\alpha^5} = \frac{\alpha^{12}}{x-1}+\frac{\alpha^2}{x-\alpha^2}+\frac{\alpha^2}{x-\alpha^3},$$

$$\frac{\omega_3(x)}{\sigma(x)} \triangleq S_{0,3}\frac{f_3(z)}{z^3} = \frac{\alpha^{12}(x^2+\alpha^7 x+\alpha^5)}{x^3+\alpha^{13}x^2+\alpha^9 x+\alpha^5} = \frac{\alpha^{11}}{x-1}+\frac{\alpha^{10}}{x-\alpha^2}+\frac{\alpha^5}{x-\alpha^3}.$$

By use of the formula (31), we get the error values

$$e_{0,1} = \alpha^{13}, e_{2,1} = \alpha^9, e_{3,1} = \alpha^9,$$

$$e_{0,2} = \alpha^{12}, e_{2,2} = \alpha^7, e_{3,2} = \alpha^2,$$

$$e_{0,3} = \alpha^{11}, e_{2,3} = \alpha^{12}, e_{3,3} = \alpha^8$$

resulting in

$$e_1 = (\alpha^{13}, 0, \alpha^9, \alpha^9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$e_2 = (\alpha^{12}, 0, \alpha^7, \alpha^2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$e_3 = (\alpha^{11}, 0, \alpha^{12}, \alpha^8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0).$$

Finally, we give an example to demonstrate our algorithm may also work for the case that the error-locations are different.

Example 4.5 (RS code, 4-error decoding, $M = 3$, $q = 2^4, u = 1, N = 15, r_1 = 6$, $r_2 = 5, r_3 = 5, l_1 = l_2 = l_3 = 1$). Consider the extension field

$$\mathrm{GF}(2^4) = \{\gamma_0 + \gamma_1\alpha + \gamma_2\alpha^2 + \gamma_3\alpha^3, \gamma_j \in \mathrm{GF}(2), j = 0, 1, 2, 3\} = (\gamma_0, \gamma_1, \gamma_2, \gamma_3),$$

where $\alpha \in \mathrm{GF}(2^4)$ is the root of the irreducible polynomial $x^4 + x + 1$ over $\mathrm{GF}(2)$. Again, we have $\alpha^{15} = 1$.

Assume that the three generator polynomials are

$$G_1(x) = \prod_{j=1}^{6}(x-\alpha^j),$$

$$G_2(x) = G_3(x) = \prod_{j=1}^{5}(x-\alpha^j),$$

in other words, we consider the sent codes

$$c_1 = (\alpha^6, \alpha^9, \alpha^6, \alpha^4, \alpha^{14}, \alpha^{10}, 1, 0, 0, 0, 0, 0, 0, 0, 0),$$

$$c_2 = c_3 = (1, \alpha, \alpha^5, \alpha^2, \alpha^7, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0).$$

If the received words are

$$b_1 = (\alpha^6, \alpha^9, \alpha^6, \alpha, \alpha^{14}, \alpha^5, 1, 0, 0, 0, 0, 0, 1, 0, 0),$$

$$b_2 = (1, \alpha^9, \alpha^5, \alpha^2, \alpha^7, 1, 0, 0, 0, 0, 0, 0, \alpha^{13}, 0, 0),$$

$$b_3 = (1, 1, \alpha^5, \alpha^{10}, \alpha^7, \alpha^4, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$

we first calculate the syndromes and obtain

$$\begin{array}{lllll} S_{0,1} = 1, & S_{1,1} = 1, & S_{2,1} = \alpha^{10}, & S_{3,1} = 1, & S_{4,1} = \alpha^{10}, \quad S_{5,1} = \alpha^5, \\ S_{0,2} = \alpha^2, & S_{1,2} = \alpha^{13}, & S_{2,2} = \alpha^{12}, & S_{3,2} = \alpha^{14}, & S_{4,2} = \alpha^3, \\ S_{0,3} = 1, & S_{1,3} = \alpha^8, & S_{2,3} = \alpha^2, & S_{3,3} = \alpha^7, & S_{4,3} = \alpha^{10}. \end{array}$$

It follows that g_j are

$$\begin{array}{llllll} g_0 = 1, & g_3 = 1, & g_6 = \alpha^{10}, & g_9 = 1, & g_{12} = \alpha^{10}, & g_{15} = \alpha^5, \\ g_1 = \alpha^2, & g_4 = \alpha^{13}, & g_7 = \alpha^{12}, & g_{10} = \alpha^{14}, & g_{13} = \alpha^3, & \\ g_2 = 1, & g_5 = \alpha^8, & g_8 = \alpha^2, & g_{11} = \alpha^7, & g_{14} = \alpha^{10}. & \end{array}$$

Subsequently, we obtain the determinants

$$\begin{array}{lllll} G_0^{(0)} = 1, & G_1^{(0)} = 1, & G_2^{(0)} = \alpha^{14}, & G_3^{(0)} = \alpha^3, & G_4^{(0)} = \alpha^8, \\ G_0^{(1)} = 1, & G_1^{(1)} = \alpha^2, & G_2^{(1)} = \alpha^9, & G_3^{(1)} = \alpha^{12}, & G_4^{(1)} = \alpha^3, \\ G_0^{(2)} = 1, & G_1^{(2)} = 1, & G_2^{(2)} = \alpha, & G_3^{(2)} = \alpha^6, & G_4^{(2)} = \alpha^{10}, \\ G_0^{(3)} = 1, & G_1^{(3)} = 1, & G_2^{(3)} = \alpha^9, & G_3^{(3)} = \alpha^4, & G_4^{(3)} = \alpha^{14}. \end{array}$$

Then we have the matrix L by calculating a_j

$$L = \begin{pmatrix} 0 & 0 & 0 & 1 & & & & & & & & & & & \\ \alpha^2 & 0 & 0 & 0 & 1 & & & & & & & & & & \\ & \alpha^{13} & 0 & 0 & 0 & 1 & & & & & & & & & \\ & & 1 & 0 & 0 & 0 & 1 & & & & & & & & \\ & & & \alpha^{14} & 0 & 0 & 0 & 1 & & & & & & & \\ & & & & \alpha^8 & 0 & 0 & 0 & 1 & & & & & & \\ & & & & & \alpha^9 & 0 & 0 & 0 & 1 & & & & & \\ & & & & & & \alpha^8 & 0 & 0 & 0 & 1 & & & & \\ & & & & & & & \alpha^{10} & 0 & 0 & 0 & 1 & & & \\ & & & & & & & & \alpha^{14} & 0 & 0 & 0 & 1 & & \\ & & & & & & & & & \alpha^2 & 0 & 0 & 0 & 1 & \\ & & & & & & & & & & \alpha^5 & 0 & 0 & 0 & 1 \\ & & & & & & & & & & & \alpha^{10} & 0 & 0 & 0 \\ & & & & & & & & & & & & \alpha & 0 & 0 \\ & & & & & & & & & & & & & \alpha^{13} & 0 \\ & & & & & & & & & & & & & & \alpha^6 \end{pmatrix}.$$

The rational functions $f_v(z)$ are

$$f_1(z) = \frac{z^{15} + \alpha z^{11} + \alpha^5 z^7 + \alpha^6 z^3}{z^{16} + \alpha^4 z^{12} + \alpha z^8 + \alpha^5 z^4 + \alpha^6},$$

$$f_2(z) = \frac{z^{15} + \alpha^{13} z^{11} + \alpha^2 z^7 + \alpha^{12} z^3}{z^{16} + \alpha^4 z^{12} + \alpha z^8 + \alpha^5 z^4 + \alpha^6},$$

$$f_3(z) = \frac{z^{15} + \alpha^5 z^{11} + \alpha^{14} z^7 + \alpha^7 z^3}{z^{16} + \alpha^4 z^{12} + \alpha z^8 + \alpha^5 z^4 + \alpha^6}.$$

By setting $x = z^4$ we get

$$\frac{\omega_1(x)}{\sigma(x)} \triangleq S_{0,1}\frac{f_1(z)}{z^3} = \frac{x^3 + \alpha x^2 + \alpha^5 x + \alpha^6}{x^4 + \alpha^4 x^3 + \alpha x^2 + \alpha^5 x + \alpha^6} = \frac{\alpha^3}{x - \alpha^3} + \frac{\alpha^5}{x - \alpha^5} + \frac{\alpha^{12}}{x - \alpha^{12}},$$

$$\frac{\omega_2(x)}{\sigma(x)} \triangleq S_{0,2}\frac{f_2(z)}{z^3} = \frac{\alpha^2(x^3 + \alpha^{13} x^2 + \alpha^2 x + \alpha^{12})}{x^4 + \alpha^4 x^3 + \alpha x^2 + \alpha^5 x + \alpha^6} = \frac{\alpha^4}{x - \alpha} + \frac{\alpha^{10}}{x - \alpha^{12}},$$

$$\frac{\omega_3(x)}{\sigma(x)} \triangleq S_{0,3}\frac{f_3(z)}{z^3} = \frac{x^3 + \alpha^5 x^2 + \alpha^{14} x + \alpha^7}{x^4 + \alpha^4 x^3 + \alpha x^2 + \alpha^5 x + \alpha^6} = \frac{\alpha^5}{x - \alpha} + \frac{\alpha^7}{x - \alpha^3} + \frac{\alpha^6}{x - \alpha^5}.$$

By use of the formula (31), we get the error values

$$e_{3,1} = 1, e_{5,1} = 1, e_{12,1} = 1,$$

$$e_{1,2} = \alpha^3, e_{12,2} = \alpha^{13},$$

$$e_{1,3} = \alpha^4, e_{3,3} = \alpha^4, e_{5,3} = \alpha,$$

leading to

$$e_1 = (0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0),$$
$$e_2 = (0, \alpha^3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \alpha^{13}, 0, 0),$$
$$e_3 = (0, \alpha^4, 0, \alpha^4, 0, \alpha, 0, 0, 0, 0, 0, 0, 0, 0, 0).$$

5. Conclusion

In this paper, the moment problem related to a category of extended Lotka–Volterra systems is used for decoding multiple BCH-Goppa codes. When the errors of the multiple codes happen at the same locations, our algorithm needs fewer known syndromes and lower complexity than applying the single BCH-Goppa decoding algorithm in [2] multiple times.

As a future work, we would try to introduce the full-discrete version of the extended Lotka–Volterra systems over finite fields to design a new decoding algorithm involving the recursion relation.

Acknowledgments

We thank Dr Shihao Li for his useful discussions. XC was supported in part by the National Natural Science Foundation of China (Grant Nos. 11688101, 11731014, 11701550) and the Youth Innovation Promotion Association CAS. XH was supported in part by the National Natural Science Foundation of China (Grant Nos.11931017 and 11871336).

ORCID iDs

Xiang-Ke Chang https://orcid.org/0000-0003-0056-8619

References

[1] Faybusovich L 1994 On the Rutishauser's approach to eigenvalue problems *Linear Algebra for Control Theory* (*IMA Vol. Math. Appl.* vol 62) ed P Van Dooren and B Wyman (New York: Springer) pp 87–102
[2] Nakamura Y 1996 The BCH-Goppa decoding as a moment problem and a tau function over finite fields *Phys. Lett.* A **223** 75–81
[3] Nakamura Y and Mukaihira A 1998 Dynamics of the finite Toda molecule over finite fields and a decoding algorithm *Phys. Lett.* A **249** 295–302
[4] Doliwa A, Bialecki M and Klimczewski P 2003 The Hirota equation over finite fields: algebro-geometric approach and-multisoliton solutions *J. Phys. A: Math. Gen.* **36** 4827–39
[5] Rutishauser H 1954 Ein infinitesimales analogon zum quotienten-differenzen-algorithmus *Arch. Math.* **5** 132–7
[6] Shirota N 1995 Relationship of decoding method of Reed–Solomon codes to the inverse scattering method in Toda's exponential lattice *The 18th Symp. on Information Theory and its Applications* pp 751–2 (in Japanese)
[7] Berlekamp E R 1968 *Algebraic Coding Theory* (New York: McGraw-Hill)
[8] Nakamura Y 1998 Calculating Laplace transforms in terms of the Toda molecule *SIAM J. Sci. Comput.* **20** 306–17
[9] Dai Z D and Zeng K C 1990 Continued fractions and the Berlekamp–Massey algorithm *Advances in Cryptology—AUSCRYPT* vol 453, ed J Seberry and J Pieprzyk (Berlin: Springer) pp 23–31
[10] Tamm U 2001 Some aspects of Hankel matrices in coding theory and combinatorics *Electron. J. Combin.* **8** 1–31
[11] Gashkov S B and Gashkov I B 2004 The Berlekamp–Massey algorithm. A sight from theory of Padé approximants and orthogonal polynomials *Computational Science—ICCS* ed M Bubak *et al* (Berlin: Springer) pp 561–4
[12] Ahlswede R 2018 Orthogonal polynomials in information theory *Combinatorial Methods and Models: Rudolf Ahlswede's Lectures on Information Theory* 4 (*Foundations in Signal Processing, Communications and Networking* vol 13) ed A Ahlswede *et al* (Berlin: Springer) pp 307–73
[13] Gashkov S B and Gashkov I B 2006 Berlekamp–Massey algorithm, continued fractions, Padé approximations, and orthogonal polynomials *Math. Notes* **79** 41–54
[14] Moser J 1975 Three integrable Hamiltonian systems connected with isospectral deformations *Adv. Math.* **16** 197–220
[15] Nakamura Y and Kodama Y 1995 Moment problem of Hamburger, hierarchies of integrable systems, and the positivity of tau-functions *Acta Appl. Math.* **39** 435–43
[16] Nakamura Y, Kajiwara K and Shiotani H 1998 On an integrable discretization of the Rayleigh quotient gradient system and the power method with a shift *J. Comput. Appl. Math.* **96** 77–90
[17] Narita K 1982 Soliton solution to extended Volterra equation *J. Math. Soc. Japan* **51** 1682–5
[18] Itoh Y 1987 Integrals of a Lotka–Volterra system of odd number of variables *Theor. Phys.* **78** 507–10
[19] Bogoyavlensky O I 1988 Integrable dynamical systems associated with the KdV equation *Math. USSR Izv.* **31** 435–54

[20] Bogoyavlensky O I 1988 Some constructions of integrable dynamical systems *Math. USSR Izv.* **31** 47–75
[21] Bogoyavlensky O I 1988 Integrable discretizations of the KdV equation *Phys. Lett.* A **134** 34–8
[22] Moser J 1975 Finitely many mass points on the line under the influence of an exponential potential—an integrable system *Dynamical Systems, Theory and Applications* (*Lect. Notes Phys.* vol 38) ed J Moser (Berlin: Springer) pp 467–97
[23] Akhiezer N I and Kemmer N 1965 The classical moment problem and some related questions in analysis *University Mathematical Monographs* vol 5 (Edinburgh: Oliver & Boyd)
[24] Achiezer N I and Krein M G 1962 *Some Questions in the Theory of Moments* (Providence, RI: American Mathematical Society)
[25] van Lint J H 1999 Introduction to coding theory *GTM* vol 86 (Berlin: Springer)
[26] Chang X K, Chen X M, Hu X B and Tam H W 2014 About several classes of bi-orthogonal polynomials and discrete integrable systems *J. Phys. A Math. Theor.* **48** 015204
[27] Wang B, Chang X K, Hu X B and Li S H 2020 Discrete invariant curve flows, orthogonal polynomials and moving frame *Int. Math. Res. Not.* (accepted)
[28] Bogoyavlensky O I 1988 Five constructions of integrable dynamical systems connected with the Korteweg–de Vries equation *Acta Appl. Math.* **13** 227–66