A Simple Step-by-Step Decoding of Binary BCH Codes

Ching-Lung CHR*, Szu-Lin SU†, Members, and Shao-Wei WU†, Nonmember

SUMMARY In this letter, we propose a simplified step-by-step decoding algorithm for t-error-correcting binary Bose-Chaudhuri-Hocquenghem (BCH) codes based on logical analysis. Compared to the conventional step-by-step decoding algorithm, the computation complexity of this decoder is much less, since it significantly reduces the matrix calculation and the operations of multiplication.

key words: BCH code, step-by-step decoding, matrix computation, computational complexity

1. Introduction

Among the most well-known error-correcting codes, the Bose-Chaudhuri-Hocquenghem (BCH) codes are a class of powerful random-error-correcting cyclic codes [1]–[3]. The popular binary BCH decoder architecture can be summarized into three steps: a) Calculate the syndromes from the received codeword. b) Compute the error locator polynomial. c) Find the error location, and then correct errors.

There is another decoding method, called the step-by-step decoding algorithm [4], which can decode a cyclic code in a serial manner with a low hardware complexity. The method can directly determine whether any bit in received word is correct or not without finding the error-location polynomial. Since the requirement for calculation of determinant of the syndrome matrix, the conventional step-by-step decoding algorithm has not been widely employed for BCH codes with large error-correcting capability. To reduce the number of matrix-calculations, [6] proposed a low-complexity step-by-step decoding algorithm for t-error-correcting binary BCH codes. However, it is not the most simplified decoder.

This letter presents a novel step-by-step decoding algorithm for t-error-correcting binary BCH codes. Based on logical analysis, the determination whether a received bit is erroneous in the method as proposed in [6] can be further simplified into a simple equation. The novel decoder significantly reduces the matrix calculations and the operations of multiplication compared with the algorithms in [5], [6].

2. Decoding Algorithm

An \((n, k)\) t-error-correcting binary BCH code of block length \(n = 2^m - 1\) can be defined in terms of the roots of its generator polynomial. Let \(\alpha\) be a primitive element in the Galois field \(GF(2^m)\), where \(m\) is an integer with \(m \geq 3\). Then the generator polynomial \(g(x)\) of the code is the lowest degree polynomial over \(GF(2)\), which has \(\alpha, \alpha^2, \ldots, \alpha^{2t}\) as its roots. Let \(\Phi_1(x), \Phi_2(x), \ldots, \Phi_{2t-1}(x)\) be the distinct minimal polynomials of \(\alpha, \alpha^2, \ldots, \alpha^{2t-1}\), respectively. Then, \(g(x)\) is given by \(g(x) = LCM [\Phi_1(x), \Phi_2(x), \ldots, \Phi_{2t-1}(x)]\).

Now let \(v(x)\), \(e(x)\) and \(r(x)\) be a systematic codeword, the error polynomial and the received polynomial, respectively, so that \(r(x) = v(x) + e(x)\). The weight of the error pattern \(e(x)\) would be the number of errors in the received codeword \(r(x)\). The syndromes can be calculated by \(S_i = r(\alpha^i) = e(\alpha^i)\), for \(i = 1, 2, \ldots, 2t\).

The principle of the conventional step-by-step decoding algorithm is that it involves changing the received bits one at a time by testing to determine whether the number of errors is reduced. For \(1 \leq v \leq t\), the \(v \times v\) syndrome matrix is defined as

\[
L_v = \begin{bmatrix}
S_1 & 1 & 0 & \cdots & 0 \\
S_3 & S_2 & S_1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
S_{2v-1} & S_{2v-2} & S_{2v-3} & \cdots & S_v
\end{bmatrix}
\]

The relationship between the syndrome matrices and the number of errors in \(r(x)\) can be given by theorem 9.11 in [1]. The theorem is rewritten as follows:

**Theorem 1**: For any binary BCH code and any \(v\) such that \(1 \leq v \leq t\), the \(v \times v\) syndrome matrix \(L_v\) is singular if the number of errors is \(v-1\) or less, and is nonsingular if the number of errors is \(v\) or \(v+1\).

Using the theorem, the number of errors in \(r(x)\) can be determined from the values of the determinants of the syndrome matrices \(\det(L_v), v = 1, 2, \ldots, t\). For instance, if \(\det(L_1) \neq 0, \det(L_2) \neq 0, \ldots, \det(L_v) = 0, \ldots, \det(L_t) \neq 0\), then two errors have occurred. Thus, in order to acquire the number of errors, the step-by-step decoding algorithm is concerned with the values of \(\det(L_v), v = 1, 2, \ldots, t\), equal to zero, and a decision vector \(l\) composed of decision bits \(l_v\), is defined as [5]

\[
l = (l_1, l_2, \ldots, l_t)
\]

where \(l_v = 0\) if \(\det(L_v) = 0\) and \(l_v = 1\) if \(\det(L_v) \neq 0\).
If the received word $r(x)$ is modified by changing temporarily a selected bit at position $x^p$, $0 \leq p \leq n-1$, then the modified decision vector $l_p$ can also be defined as

$$l_p = (l_1, l_2, \ldots, l_n) \quad \text{(3)}$$

where the subscript “$\overline{p}$” in $l_p$ indicates that the magnitude of the $x^p$ position of the error pattern $e(x)$ is temporarily changed. Whether the bit at position $x^p$ of $r(x)$ is erroneous can be determined from the difference between $l$ and $l_p$.

Using logical analysis, a novel step-by-step decoding algorithm for $t$-error-correcting binary BCH codes is proposed as follows. The logical analysis is based on the fact that only $2t+1$ cases are possible when we want to determine the value at the $x^p$ position of the error pattern $e(x)$ for a $t$-error-correcting binary BCH code. For example, for a binary $(15, 5)$ BCH code ($t = 3$), all seven possible cases of $e_p$ are expressed in Table 1 and described as below:

1. If no error occurs, then the value of $e_p$ must be correct, i.e., $e_p = 0$.
2. If one error occurs, then the value of $e_p$ can be erroneous or correct, i.e., $e_p = 1$ or $e_p = 0$.
3. If two errors occur, then the value of $e_p$ can be erroneous or correct, i.e., $e_p = 1$ or $e_p = 0$.
4. If three errors occur, then the value of $e_p$ can be erroneous or correct, i.e., $e_p = 1$ or $e_p = 0$.

**Theorem 2:** For a $t$-error-correcting binary BCH code, if $v$ ($1 \leq v \leq t$) errors occur, then the estimated error value at position $p$ ($0 \leq p \leq n-1$) of the error pattern can be given by

$$\hat{e}_p(t) = \overline{l_{v,p}}, \quad \text{for} \quad 1 \leq v \leq t \quad \text{(4)}$$

**Proof:**

**Case $t = 1$:** Let $v$ be the number of errors, and $e_p(1)$ be the value at the $x^p$ position of the error pattern. There are three possible cases apply in determining the value of $e_p(1)$, as shown in row 2 of Table 2. $v_p$ and $l_{1,p}$ indicates the weight of the error pattern and the decision bit for changing the received digit $r_p$, respectively. Row 4 is the decision bit $l_{1,\overline{p}}$. If no error occurs ($v_p = 0$), $det(L_{1,\overline{p}}) = 0$, then $l_{1,\overline{p}} = 1$. If one error occurs ($v_p = 1$), $det(L_{1,\overline{p}}) \neq 0$, then $l_{1,\overline{p}} = 1$. If two errors occur ($v_p = 2$), $det(L_{1,\overline{p}}) \neq 0$, then $l_{1,\overline{p}} = 1$. It is easy to see that $\hat{e}_p(1) = \overline{l_{1,\overline{p}}}$.

**Case $t \geq 2$:** Only $2t+1$ possible cases when we determine the value of position $x^p$ of the error pattern, as shown in row 2 of Table 3. As the case of $t = 1$, the decision bits, $l_{2,p}, l_{3,p}, \ldots$, and $l_{t,\overline{p}}$ in Table 3 can be determined by Theorem 1. It is easy to see that all $2t+1$ possible estimated values of $e_p(t)$, as shown in row 7, 10, and 12 of Table 3, are equal to $\overline{l_{v,p}}$. Hence, in a similar way, we can get

$$\hat{e}_p(t) = \overline{l_{v,p}}, \quad \text{for} \quad 2 \leq v \leq t$$

Q.E.D.

The matrix calculations and the operations of multiplication needed for the algorithms in [5], [6] and the proposed algorithm are given in Table 4. Since the probability of a large number of errors is small, only a low-scale matrix calculation is needed in the decoder. Therefore, the proposed algorithm significantly reduces the matrix calculations and the operations of multiplication.

According to Theorem 2, the novel step-by-step decoding procedure can be presented as follows:

1. Determine the number of errors $v$. Detect consecutively $det(L_0), det(L_{t-1}), \ldots$, until a nonzero determinant is found [1]. Then the number of errors is $v$.
2. Let $p = n - 1$.
3. Change the magnitude at position $p$ of $r(x)$ temporarily and determine the modified syndromes $S_{t,\overline{p}}(i = 1, 2, \ldots, 2t; n - k \leq p \leq n - 1)$.
4. Estimate the error value $\hat{e}_p(t) = \overline{l_{v,p}}$.
5. Send the output bit $\hat{r}_p = r_p + \hat{e}_p(t)$.
6. Let $p = p - 1$. If $p = n-k-1$ or all $v$ errors have been found, then this decoding algorithm is completed. Otherwise,
Table 4 The matrix calculations and multiplications of the estimated error value for $t = 3$.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$(l, l_1, p, l_2, p, l_3, p)$</td>
<td>$(l, l_1, p, l_2, p, l_3, p)$</td>
<td>$(l, l_1, p, l_2, p, l_3, p)$</td>
</tr>
</tbody>
</table>

$v=1$

<table>
<thead>
<tr>
<th>multipliers</th>
<th>12</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix calculations</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
</tr>
<tr>
<td>$\hat{e}_p(3)$</td>
<td>$\hat{e}_p(3)$</td>
<td>$\hat{e}_p(3)$</td>
<td></td>
</tr>
</tbody>
</table>

$v=2$

<table>
<thead>
<tr>
<th>multipliers</th>
<th>12</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix calculations</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
</tr>
<tr>
<td>$\hat{e}_p(3)$</td>
<td>$\hat{e}_p(3)$</td>
<td>$\hat{e}_p(3)$</td>
<td></td>
</tr>
</tbody>
</table>

$v=3$

<table>
<thead>
<tr>
<th>multipliers</th>
<th>12</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix calculations</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
<td>det($l_1$), det($l_2$), det($l_3$)</td>
</tr>
<tr>
<td>$\hat{e}_p(3)$</td>
<td>$\hat{e}_p(3)$</td>
<td>$\hat{e}_p(3)$</td>
<td></td>
</tr>
</tbody>
</table>

go to step 3.

3. Decoder Architecture

Based on the above step-by-step decoding algorithm, a hardware structure of the decoder is presented. Figure 1 shows the functional block diagram of the decoder. It consists of four parts:

1. n-stage buffer register: The buffer register is used to store the received vector and to correct received vector bit by bit.
2. Syndrome generator: The syndrome generator is used to obtain the original syndromes $S_i (i = 1, 2, 3, \ldots, 2t)$ and the temporarily changed syndromes $S_i, p (i = 1, 2, 3, \ldots, 2t, n - k \leq p \leq n - 1)$.
3. Error-number calculator: The error-number calculator is used to calculate the determinants $det(L_v), 1 \leq v \leq t$, and then determine the number of errors $v$.
4. Error-corrector: The error-corrector is used to estimate the error value $\hat{e}_p(t) = \hat{l}_n, p$. If the corresponding bit is judged to be an erroneous bit, the decoder sends a correcting bit $\hat{e}_p(t) = 1$ to change its magnitude.

Fig. 1 Functional block diagram of the binary step by step decoder.

4. Example of Decoding of Binary (15, 5) BCH Code

Let $\alpha$ be a primitive element of $GF(2^4)$, satisfying $\alpha^4 + \alpha + 1 = 0$. The generator polynomial of the binary (15, 5) BCH code is defined as the least common multiple of the minimal polynomials of $1, \alpha, \alpha^2, \ldots, \alpha^6$. Assume $c(x) = 0$ and $e(x) = x^3 + x^4$ are the codeword and the error polynomial, respectively. There are two errors in the received polynomial $r(x)$. The initial syndromes are

$$S_1 = 1, S_2 = 1, S_3 = \alpha^8, S_4 = 1, S_5 = \alpha^5, S_6 = \alpha^4.$$ 

Detect consecutively $det(L_v)$:

Since

$$det(L_3) = \begin{vmatrix} S_1 & 1 & 0 \\ S_3 & S_2 & S_1 \\ S_5 & S_4 & S_3 \end{vmatrix} = 0$$

and

$$det(L_2) = \begin{vmatrix} S_1 & 1 \\ S_3 & S_2 \\ S_5 & S_4 \end{vmatrix} = \alpha^2 \neq 0,$$

so $l_3 = 0$ and $l_2 = 1$.

Then, we confirm that two errors ($v = 2$) occur and the estimated error value of $e_p(3)$ is

$$\hat{e}_p(3) = \hat{l}_2, p.$$ 

For $10 \leq p \leq 14$, the temporarily changed syndrome values can be given by

$$S_i, p = S_i + \alpha^ip.$$ 

Then,
\[ S_{1,\hat{f}_4} = \alpha^3, S_{1,\hat{f}_3} = \alpha^6, S_{1,\hat{f}_2} = \alpha^{11}, S_{1,\hat{f}_1} = \alpha^{12}, S_{1,\hat{f}_0} = \alpha^5, S_{2,\hat{f}_4} = \alpha^6, S_{2,\hat{f}_3} = \alpha^{12}, S_{2,\hat{f}_2} = \alpha^7, S_{2,\hat{f}_1} = \alpha^9, S_{2,\hat{f}_0} = \alpha^{10}, S_{3,\hat{f}_4} = \alpha^9, S_{3,\hat{f}_3} = \alpha^{12}, S_{3,\hat{f}_2} = \alpha^{14}, S_{3,\hat{f}_1} = \alpha^{13}, S_{3,\hat{f}_0} = \alpha^2. \]

By calculating the value of
\[
\det(L_{2,\hat{p}}) = \begin{vmatrix}
S_{1,\hat{p}} & 1 \\
S_{3,\hat{p}} & S_{2,\hat{p}}
\end{vmatrix},
\] (7)

we can determine
\[ l_{2,\hat{f}_4} = 0, l_{2,\hat{f}_3} = 1, l_{2,\hat{f}_2} = 1, l_{2,\hat{f}_1} = 1, l_{2,\hat{f}_0} = 1, \]
and
\[ \hat{e}_{14} = \tilde{l}_{2,\hat{f}_4} = 1, \hat{e}_{13} = \tilde{l}_{2,\hat{f}_3} = 0, \hat{e}_{12} = \tilde{l}_{2,\hat{f}_2} = 0, \]
\[ \hat{e}_{11} = \tilde{l}_{2,\hat{f}_1} = 0, \hat{e}_{10} = \tilde{l}_{2,\hat{f}_0} = 0. \]

Hence, the estimated error values of the message part are
\[ (\hat{e}_{10}, \hat{e}_{11}, \hat{e}_{12}, \hat{e}_{13}, \hat{e}_{14}) = (0, 0, 0, 0, 1). \]

5. Conclusions

A modified step-by-step decoding algorithm for t-error-correcting binary BCH codes has been presented. Based on logical analysis, we obtain a simple rule for estimating the value of the error pattern. The proposed algorithm significantly reduces the matrix calculations and the operations of multiplication as the algorithms in [5], [6]. Thus, the computational complexity of this decoder is much less, since the most complex element of the step-by-step decoder is the “matrix-computing” element. Furthermore, the simple and regular structure also makes it suitable for hardware implementation.

References