# The Decoding of Reed-Solomon Codes

R. J. McEliece
Communications Systems Research Section
Electrical Engineering Department
California Institute of Technology

*Reed-Solomon (RS) codes form an important part of the high-rate downlink telemetry system for the Magellan mission, and the RS decoding function for this project will be done by the DSN. Although the basic idea behind all Reed-Solomon decoding algorithms was developed by Berlekamp in 1968, there are dozens of variants of Berlekamp's algorithm in current use. This paper attempts to restore order by presenting a mathematical theory which explains the working of almost all known RS decoding algorithms. The key innovation that makes this possible is the unified approach to the solution of the key equation, which simultaneously describes the Berlekamp, Berlekamp-Massey, Euclid, and continued fractions approaches. Additionally, a detailed analysis is made of what can happen to a generic RS decoding algorithm when the number of errors and erasures exceeds the code's designed correction capability, and it is shown that while most published algorithms do not detect as many of these error-erasure patterns as possible, by making a small change in the algorithms, this problem can be overcome.*

## I. Decoding Reed-Solomon Codes

In this article we will give a general definition of Reed-Solomon codes, state the abstract Reed-Solomon decoding problem, describe the two main classes of decoding algorithms (time- and frequency-domain decoders), and then give three theorems. Theorem 1 explains why the RS decoding algorithms work, and Theorems 2 and 3 delineate exactly what happens if the number of errors and erasures exceeds the codes' designed correction capability. The article concludes with three appendices, which give the mathematical background needed for the proofs of the theorems presented.

Let $F$ be a field which contains a primitive $n$th root of unity $\alpha$. (We assume that the characteristic of the field does not divide $n$.) If $L$ and $r$ are fixed integers between 0 and $n$, the set of all codewords (vectors) $\mathbf{C} = (C_0, \ldots, C_{n-1})$ over $F$ such that

$$\sum_{i=0}^{n-1} C_i \alpha^{ij} = 0 \qquad j = L, L+1, \ldots, L+r-1 \qquad (1)$$

is called a Reed-Solomon code. The parameters $n, r$, and $L$ are called the *length*, *redundancy*, and *offset* of the code. The

parameters $k = n - r$ is called the code's dimension. (Commonly $L = 0$, 1, or $(n - r + 1)/2$.) The polynomial $g(x)$, defined by

$$g(x) = \prod_{j=L}^{L+r-1} (x - \alpha^j) \tag{2}$$

is called the code's *generator polynomial*. Note that $\mathbf{C}$ is a codeword if and only if the generating function for $\mathbf{C}$, viz., $C(x) = C_0 + C_1 x + \cdots + C_{n-1}x^{n-1}$ is a multiple of $g(x)$. If $n$ is odd and $r$ is even, and $L = (n - r + 1)/2$, then the roots of $g(x)$ come in reciprocal pairs $(\alpha^j, \alpha^{-j})$, for $j = L, \ldots, L + (r/2) - 1$, and $g(x)$ is a "palindrome."

The *basic metric property* of an RS code is that any two codewords must differ in at least $r + 1$ positions. Thus if $d_{\mathbf{H}}(\mathbf{C}, \mathbf{C}')$ denotes the Hamming distance between the codewords $\mathbf{C}$ and $\mathbf{C}'$, and $\mathbf{C} \neq \mathbf{C}'$, then it follows that

$$d_{\mathbf{H}}(\mathbf{C}, \mathbf{C}') \geqslant r + 1 \tag{3}$$

The *basic combinatorial property* of an RS code is that given any subset $I \subseteq \{0, 1, \ldots, n - 1\}$ of at most $k$ coordinate positions, and an arbitrary set $\{\alpha_i : i \in I\}$ of elements from $F$, there exists an RS codeword $\mathbf{C}$ such that $C_i = \alpha_i$ for all $i \in I$. (Proofs of these basic properties can be found in [2], Section 7.3.)

Suppose we transmit a codeword $\mathbf{C}$ over a channel, which can, on occasion, change any symbol from $F$ into any other, and which in addition can "erase" any symbol, i.e., make it completely unintelligible. To model erasures, we introduce an "erasure symbol" $*$ and add it to $F : \bar{F} = F \cup \{*\}$. Thus we send a codeword, and receive a vector $\mathbf{R} = (R_0, R_1, \ldots, R_{n-1})$ from $\bar{F}^n$. The RS *decoding problem*, ideally, would be this: given $\mathbf{R} \in \bar{F}^n$, find the nearest RS codeword. However, that proves to be too hard, and we must be satisfied with the solution to an easier but closely related problem.

To state the decoding problem precisely, we must define a distance between vectors over $\bar{F}$, the RS *decoding distance*. If $\mathbf{V} = (V_0, \ldots, V_{n-1})$ and $\mathbf{V}' = (V_0', \ldots, V_{n-1}')$ are vectors with components in $\bar{F}$, we define

$$d_{\mathbf{RS}}(\mathbf{V}, \mathbf{V}') = \sum_{i=0}^{n-1} d_{\mathbf{RS}}(V_i, V_i') \tag{4}$$

where if $x$ and $y$ are elements of $\bar{F}$,

$$d_{\mathbf{RS}}(x, y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \text{ and either } x \text{ or } y \text{ is } * \\ 2 & \text{if } x \neq y \text{ but neither } x \text{ nor } y \text{ is } * \end{cases} \tag{5}$$

One way to think about this metric is shown in Fig. 1, for $F = GF(3)$. The elements of $\bar{F}$ are the vertices of a graph, with every element of $F$ connected to $*$ by an edge. Then $d_{\mathbf{RS}}(x, y)$ is just the distance between $x$ and $y$ in the graph. Note that if $\mathbf{V}$ and $\mathbf{V}'$ are vectors with components in $F$ (i.e., with no $*$s) we have

$$d_{\mathbf{RS}}(\mathbf{V}, \mathbf{V}') = 2 d_{\mathbf{H}}(\mathbf{V}, \mathbf{V}') \tag{6}$$

The *decoding problem* for an RS code of redundancy $r$ can now be stated. Given an arbitrary vector $\mathbf{R} = (R_0, \ldots, R_{n-1})$ with components from $\bar{F}$, find all RS codewords $\mathbf{C}$ such that

$$d_{\mathbf{RS}}(\mathbf{C}, \mathbf{R}) \leqslant r \tag{7}$$

First, note that there can be at most one codeword $\mathbf{C}$ such that Eq. (7) holds. This is because if $d_{\mathbf{RS}}(\mathbf{C}, \mathbf{R}) \leqslant r$ and $d_{\mathbf{RS}}(\mathbf{C}', \mathbf{R}) \leqslant r$, then by the triangle inequality

$$d_{\mathbf{RS}}(\mathbf{C}, \mathbf{C}') \leqslant d_{\mathbf{RS}}(\mathbf{C}, \mathbf{R}) + d_{\mathbf{RS}}(\mathbf{R}, \mathbf{C}') \leqslant 2r \tag{8}$$

which implies by Eq. (5) that $d_{\mathbf{H}}(\mathbf{C}, \mathbf{C}') \leqslant r$, violating Eq. (3), unless $\mathbf{C} = \mathbf{C}'$.

We now describe an efficient algorithm, essentially due to Elwyn Berlekamp ([1], Chapter 7), for solving the decoding problem.

For a given $\mathbf{R}$, its *erasure set* $I_0$ is defined as

$$I_0 = \{i : R_i = *\} \qquad t_0 = |I_0| \tag{9}$$

(The notation $|S|$ denotes the number of elements in the set $S$.) The decoder's first step is to calculate the *erasure locator polynomial* $\sigma_0(x)$, defined by

$$\sigma_0(x) = \prod_{i \in I_0} (1 - \alpha^i x) \tag{10}$$

(If there are no erasures in $\mathbf{R}$, $\sigma_0(x)$ is defined to be 1.) If the number of erasures $t_0$ exceeds $r$, there can be no solutions to Eq. (7); in this case the decoder should simply print "too many erasures!" and stop. We will assume that $t_0 \leqslant r$ in the rest of the discussion of the decoding algorithm.

Once the erasure locator polynomial is calculated, the decoder replaces the *s in $\mathbf{R}$ with symbols from $F$. Usually these symbols are chosen to be 0s, but if the decoder has "side information" about the original values of the $C_i$s corresponding to the erased indices $i \in I_0$, these values can be used. In any case, the result is a new vector $\mathbf{R}' = (R'_0, \ldots, R'_{n-1})$, defined by

$$R'_i = \begin{cases} R_i & \text{if } i \notin I_0 \\ Z_i & \text{if } i \in I_0 \end{cases} \tag{11}$$

where $Z_i = 0$ is the usual choice.

Next, the *syndrome* is computed, i.e., the $r$ values

$$S_j = \sum_{i=0}^{n-1} R'_i \alpha^{ij} \qquad \text{for } j = L, L+1, \ldots, L+r-1 \tag{12}$$

which are used as coefficients in the *syndrome polynomial*

$$S(x) = S_L + S_{L+1}x + \cdots + S_{L+r-1}x^{r-1} \tag{13}$$

If erasures are present the decoder continues by calculating the *modified syndrome polynomial* $S_0(x)$, defined by

$$S_0(x) = S(x)\,\sigma_0(x) \pmod{x^r} \tag{14}$$

Now comes the key step. Define the numbers $\mu$ and $\nu$ by

$$\begin{aligned} \mu &= \lfloor (r - t_0)/2 \rfloor \\ \nu &= \lceil (r + t_0)/2 \rceil - 1 \end{aligned} \tag{15}$$

(If $x$ is a real number, $\lfloor x \rfloor$ is the greatest integer less than or equal to $x$, and $\lceil x \rceil$ is the least integer greater than or equal to $x$.) It is an easy exercise to show that $\mu + \nu = r - 1$. The decoder now solves the $(x^r, S_0(x), \mu, \nu)$ problem, i.e., it finds the unique lowest degree pair of polynomials $\sigma_1(x)$ and $\omega(x)$ such that $\deg(\sigma_1) \leqslant \mu$, $\deg(\omega) \leqslant \nu$, and

$$\sigma_1(x)S_0(x) \equiv \omega(x) \pmod{x^r}$$

(see Appendix C). The polynomial $\sigma_1(x)$ is traditionally called the *error locator polynomial*, and $\omega(x)$ is called the *error-and-erasure evaluator polynomial*. Now the decoder multiplies $\sigma_0(x)$ and $\sigma_1(x)$, obtaining a polynomial $\sigma(x)$, called the erasure/error locator polynomial.

Once the polynomials $\sigma(x)$ and $\omega(x)$ are known, there are two essentially different ways to complete the algorithm. These are usually called the *time-domain* approach and the *frequency-domain* approach.

The *time-domain* approach can be described by the following pseudocode fragment.

```
/* Time domain fragment */
{
  if (σ0 = 0 or deg(ω) ≥ t0 + deg(σ1))
    decode = FALSE;
  else {
    count = 0;
    for (i = 0 to n - 1) {
      if (σ(α^-i) = 0 and σ'(α^-i) ≠ 0) {
        count = count + 1;
        Ei = -α^-i(L-1) ω(α^-i) / σ'(α^-i) ;
      }
      else
        Ei = 0;
    }
  if (count = deg(σ))
    decode = TRUE;
  else
    decode = FALSE;
  }
}
```

After execution, if "decode" is "TRUE," $\mathbf{C} = (C_0, \ldots, C_{n-1})$, where $C_i = R'_i - E_i$ for $i = 0, 1, \ldots, n-1$ is the unique codeword within RS distance $r$ of $\mathbf{R}$. On the other hand, if "decode" is "FALSE," the decoder just prints the warning "no codeword within RS distance r." All early RS decoders used an algorithm much like this; such an algorithm is described as a "hybrid decoder" in Figure 9.2 in Blahut [2].

The frequency-domain approach can be described by the following pseudocode fragment. (In this listing, $d$ denotes the degree of $\sigma(x)$.)

```
/* Frequency Domain Fragment */
{
   if (σ0 = 0)
      decode = FALSE;
   else {
      decode = TRUE;
      for (j = L + r to n + L + d - 1)
         S_j = -(1/σ0) Σ_{k=1}^{d} σ_k S_{j-k};
      for (j = n + L to n + L + d - 1)
         if (S_j ≠ S_{j-n})  {
            decode = FALSE;
            break;
         }
   }
   if (decode = TRUE)
      for (i = 0 to n - 1)
         E_i = α^{-Li} · (1/n) Σ_{j=0}^{n-1} S_{L+j} α^{-ij};
}
```

The decoder now finishes exactly as the time-domain decoder did. The "frequency-domain decoders" described in [2, Figure 9.2] and the decoder described in [5] follow this general description. (The "time-domain decoder" described in Figure 9.7 in [2] is a rare example of an RS decoding algorithm which is apparently not closely related to the descriptions in this section. See Whiting [6] for a survey of Reed-Solomon decoding algorithms.)

In each case, the algorithm will locate the codeword within RS distance $r$ of $\mathbf{R}$, if there is one, and will print the message "no codeword within RS distance r" if there is not. The following theorem explains why.

**Theorem 1**. There is a codeword within RS distance $r$ of $\mathbf{R}$ if and only if the following three conditions are satisfied:

(A) $\deg(\omega(x)) < t_0 + \deg(\sigma_1(x))$

(B) $\sigma_1(0) \neq 0$

(C) $\sigma_0(x)\,\sigma_1(x) \mid (1 - x^n)$

**Proof**: First, we suppose there is a codeword $\mathbf{C} = (C_0, \ldots, C_{n-1})$ within RS distance $r$ of $\mathbf{R}$. We will show that conditions (A), (B), and (C) are satisfied. To do this, we define $I_0$, $\sigma_0(x)$, and $\mathbf{R}'$ as in Eq. (11) (the erasure fills $Z_i$ can be arbitrary). Next, we define the *error set* $I_1$ and *error locator polynomial* $\sigma_1(x)$ as

$$I_1 = \{i \notin I_0 : R_i \neq C_i\} \tag{16}$$

$$\sigma_1(x) = \prod_{i \in I_1} (1 - \alpha^i x) \tag{17}$$

and the *error-and-erasure pattern* as $\mathbf{E} = (E_0, \ldots, E_{n-1})$, where

$$E_i = R'_i - C_i \qquad \text{for } i = 0, \ldots, n-1 \tag{18}$$

Finally we define the *error-and-erasure set* $I$ and the *error-and-erasure locator polynomial* $\sigma(x)$ by

$$I = I_0 \cup I_1 \tag{19}$$

$$\sigma(x) = \prod_{i \in I} (1 - \alpha^i x) \tag{20}$$

It follows from Eqs. (12), (18), and (1) that the syndromes $S_j$ satisfy

$$S_j = \sum_{i=0}^{n-1} E_i \alpha^{ij} \qquad \text{for } j = L, \ldots, L + r - 1 \tag{21}$$

which implies that

$$S_{j+L} = \sum_{i=0}^{n-1} (E_i \alpha^{iL})\, \alpha^{ij} \qquad \text{for } j = 0, \ldots, r-1 \tag{22}$$

Thus $S_L, S_{L+1}, \ldots, S_{L+r-1}$ are the first $r$ components of the DFT $\hat{\mathbf{V}} = (\hat{V}_0, \ldots, \hat{V}_{n-1})$ of the "twisted error pattern" $\mathbf{V} = (V_0, V_1, \ldots, V_{n-1})$, defined by

$$V_i = E_i \alpha^{iL} \qquad \text{for } i = 0, \ldots, n-1 \tag{23}$$

It follows then from Eqs. (13) and (22) that if we define $\hat{V}(x) = \hat{V}_0 + \hat{V}_1 x + \cdots + \hat{V}_{n-1} x^{n-1}$, then $\hat{V}(x) \equiv S(x) \pmod{x^r}$, and indeed, if we define $\hat{V}_0(x) = \sigma_0(x)\, \hat{V}(x) \bmod x^r$, that

$$\hat{V}_0(x) = S_0(x) \tag{24}$$

where $S_0(x)$ is defined in Eq. (14). If we now define the *error-and-erasure evaluator polynomial* as

$$\omega(x) = \sum_{i \in I} V_i \sigma^i(x) \tag{25}$$

where $\sigma^i(x) = \sigma(x)/(1 - \alpha^i x)$, (compare this to Eq. (B-5) in Appendix B), it follows from Theorem B-6 in Appendix B that

$$\sigma(x)\,\widehat{V}(x) = \omega(x)\,(1 - x^n) \tag{26}$$

and so, since $\sigma(x) = \sigma_0(x)\,\sigma_1(x)$ and $\widehat{V}_0(x) = \sigma_0(x)\,\widehat{V}(x)$,

$$\sigma_1(x)\,\widehat{V}_0(x) \equiv \omega(x) \pmod{x^r} \tag{27}$$

Furthermore, since **C** is assumed to have RS distance $r$ or less from **R**, it follows that $t_0 + 2\deg(\sigma_1) \leqslant r$, which in turn implies $\deg(\sigma_1) \leqslant \mu$, and $\deg(\omega) < \deg(\sigma) = t_0 + \deg(\sigma_1) \leqslant \nu$, where $\mu$ and $\nu$ are defined in Eq. (15). Furthermore, $\sigma_1$ and $\omega$ are relatively prime, since by Lemma B-2 in Appendix B for each $i \in I_1$, $\omega(\alpha^{-i}) \neq 0$. Therefore $(\sigma_1, \omega)$ is the solution to the $(x^r, \widehat{V}_0(x), \mu, \nu)$ problem, which by Eq. (24) is the same as the $(x^r, S_0(x), \mu, \nu)$ problem. Thus the polynomials produced by the decoding algorithm must be the error locator polynomial and the error-and-erasure evaluator polynomial, and these polynomials satisfy conditions (A), (B), and (C): Equation (25) implies (A); Equation (10) implies (B); Equation (20) implies (C).

To complete the proof, we suppose that conditions (A), (B), and (C) are satisfied. We will show that this implies that there is a codeword within RS distance $r$ of **R**. To do this we define $\sigma(x) = \sigma_0(x)\,\sigma_1(x)$; note that condition (A) says that $\deg(\omega) < \deg(\sigma)$, and condition (C) says that $\sigma(x) | (1 - x^n)$. Hence by Theorem B-5 in Appendix B, there exists a vector $\mathbf{V} = (V_0, V_1, \ldots, V_{n-1})$ and a support set $I$ for **V** such that

$$\sigma(x) = \lambda \sigma_I(x) \tag{28}$$

$$\omega(x) = \lambda \omega_{\mathbf{V},I}(x) \tag{29}$$

We claim now that the vector $\mathbf{C} = (C_0, \ldots, C_{n-1})$, defined by

$$C_i = R_i' - V_i \alpha^{-iL} \qquad \text{for } i = 0, \ldots, n-1 \tag{30}$$

is a codeword within RS distance $r$ of **R**. First we show that $d_{\mathbf{RS}}(\mathbf{C}, \mathbf{R}) \leqslant r$. This is because **R** has $t_0$ erasure symbols, and apart from these, differs from **C** only in those indices $i$ for which $V_i \neq 0$, i.e., $\sigma_1(\alpha^{-i}) = 0$. But by condition (C), $\sigma_1$ has exactly $\deg(\sigma_1)$ roots in $\{1, \alpha, \ldots, \alpha^{n-1}\}$, and $\deg(\sigma_1) \leqslant \mu$, and so

$$d_{\mathbf{RS}}(\mathbf{C}, \mathbf{R}) = t_0 + 2\deg(\sigma_1) \leqslant t_0 + 2\mu \leqslant t_0 + 2\lfloor (r - t_0)/2 \rfloor \leqslant r \tag{31}$$

All that remains is to show that **C**, as defined in Eq. (30), is a codeword. Since $\sigma_1$ and $\omega(x)$ solve the $(x^r, S_0(x), \mu, \nu)$ problem, we know that

$$\sigma_1(x)\,S_0(x) \equiv \omega(x) \pmod{x^r} \tag{32}$$

But by Eq. (14), $S_0(x) = S(x)\,\sigma_0(x) \bmod x^r$; and since $\sigma(x) = \sigma_0(x)\,\sigma_1(x)$, by Eq. (32) we have

$$\sigma(x)\,S(x) \equiv \omega(x) \pmod{x^r} \tag{33}$$

On the other hand, by Eqs. (28) and (29), together with Theorem B-6 in Appendix B, we have

$$\sigma(x)\,\widehat{V}(x) \equiv \omega(x) \pmod{x^r} \tag{34}$$

Now $\gcd(\sigma_0(x), x^r) = 1$ (see Eq. 10), and condition (B) guarantees that $\gcd(\sigma_1(x), x^r) = 1$, and so $\gcd(\sigma(x), x^r) = 1$. Thus by Eqs. (33) and (34) we have

$$S(x) \equiv \widehat{V}(x) \pmod{x^r} \tag{35}$$

Equating coefficients of $x^j$ for $j = 0, 1, \ldots, r-1$ on both sides of Eq. (35), we see that

$$S_{j+L} = \widehat{V}_j \qquad \text{for } j = 0, 1, \ldots, r-1 \tag{36}$$

But this implies that

$$\sum_{i=0}^{n-1} R_i' \alpha^{ij} = \sum_{i=0}^{n-1} V_i \alpha^{-iL} \alpha^{ij} \qquad \text{for } j = L, \ldots, L + r - 1 \tag{37}$$

which says that **C**, as defined by Eq. (30), is a codeword. ■

With the help of Theorem 1, we can now explain why the time-domain and frequency-domain decoders work. First, we discuss the time-domain decoder. The first line checks condition (A) and (B) of Theorem 1. The "for" loop checks condition (C), by evaluating the polynomial $\sigma(x)$ for $x = \alpha^{-i}$, for $i = 0, 1, \ldots, n-1$. Notice that there is a check for $\sigma'(\alpha^{-i}) = 0$; this is necessary, since as we will see, it is possible for $\sigma(x)$ to have a double root. The formula for $E_i$ follows from Eq. (26) and Corollary B-7 in Appendix B, and the fact that $E_i = V_i \alpha^{-Li}$ (see Eq. 23).

Next, we consider the frequency-domain decoder. The first line checks condition (B) of Theorem 3. The first "for" loop extends the sequence $S_L, S_{L+1}, \ldots, S_{L+r-1}$ recursively, using $\sigma(x)$ as the characteristic polynomial, and the second "for" loop checks to see whether or not this extension has period $n$. If the sequence is not periodic, then either condition (A) or (C) must fail, by the first part of Theorem B-10 in Appendix B. On the other hand, if the sequence is periodic, then

$$\sum_{k=0}^{d} \sigma_k S_{j-k} = 0 \qquad \text{for all } j \geq L + d$$

and so if $u(x) = S_L + S_{L+1}x + \cdots$, then $\sigma(x)\,u(x)$ has degree $< d$. But $\sigma(x)\,u(x) \equiv \omega(x) \pmod{x^r}$, and so $\deg(\omega) < d$, i.e., condition (A) is satisfied. Next, since condition (B) insures that $\gcd(\sigma_1, \omega) = 1$, the second part of Theorem B-10 shows that condition (C) holds. Finally, the formula given for the error vector $E_i$ follows from the fact that $(S_L, \ldots, S_{L+n-1})$ is the DFT $\hat{\mathbf{V}}$ of the twisted error $\mathbf{V}$ vector defined in Eq. (23). This follows from the basic Theorem B-6, which says that the components of $\hat{\mathbf{V}}$ satisfy the homogeneous difference equation whose characteristic polynomial is $\sigma(x)$.

All published RS decoding algorithms correctly locate the codeword within RS distance $r$ of the received word, if there is one. However, almost all of these algorithms (including all algorithms in [2], [3], [5], and [6]) can fail badly when there is no such word. By this we mean that there usually exist words $\mathbf{R}$ which are not within RS distance $r$ of any codeword, and yet which cause the decoder to output a vector $\mathbf{C}$ rather than to print the message "no RS codeword within RS distance r." One naive way to avoid this problem is simply to test any word produced by the decoder to see if it is a codeword, and then to see if it is within RS distance $r$ of the received word. However, this method is not quite foolproof (division by zero is possible in either the time- or frequency-domain approaches), and more complex than necessary.

The difficulty is that the decoders typically do not check all of the conditions (A), (B) and (C) of Theorem 1. The next two theorems explain why it is essential to make this check. Theorem 2 gives conditions on the polynomials $\sigma_1$ and $\omega$ that must always be satisfied, whether $\mathbf{R}$ is within RS distance $r$ of a codeword, or not. Theorem 3, on the other hand, shows that the conditions imposed on $\sigma_1$ and $\omega$ in Theorem 2 are sufficient to guarantee the existence of a vector $\mathbf{R}$ which will produce these polynomials. Together, these two theorems show that there are $\mathbf{R}$s that will produce $\sigma_1$s and $\omega$s satisfying some but not all of conditions (A), (B), and (C). Actually, condition (C) implies condition (B), but it is worthwhile to check condition (B) anyway since it is so easy to do so; if (B) fails, no further work is necessary. Theorems 2 and 3 show conditions (A) and (C) are independent, however, so that they both must be checked.

**Theorem 2.** For any word $\mathbf{R} \in \bar{F}^n$, the polynomials $\sigma_1(x)$ and $\omega(x)$ must satisfy the following three conditions:

(D) $$\deg(\sigma_1) \leq \mu$$

(E) $$\deg(\omega) \leq \nu$$

(F) $$\gcd(\sigma_1, \omega) = x^i$$

where $x^i$ is the highest power of $x$ dividing $\sigma_1(x)$.

**Proof:** Conditions (D) and (E) follow from the definition of the $(a, b, \mu, \nu)$ problem. Condition (F) follows from Lemma C-3 in Appendix C. ■

**Theorem 3.** Conversely, given a set $I_0$ of $t_0$ erasure locations and any pair of polynomials $\sigma_1(x)$ and $\omega(x)$ satisfying conditions (D), (E), and (F), there exists a vector $\mathbf{R} \in \bar{F}^n$, and a choice of "erasure fills" $Z_i$ (see Eq. 11) which will produce $\lambda\sigma_1(x)$ and $\lambda\omega(x)$ as the solution to the $(x^r, S_0(x), \mu, \nu)$ problem. Indeed, if $\sigma_1(x)$ and $\omega(x)$ are relatively prime, let $\mathbf{S} = [S_0, S_1, \ldots, S_{n-1}]$ be any vector such that

$$S_L + S_{L+1}x + \cdots + S_{L+r-1}x^{r-1} = \frac{\omega(x)}{\sigma(x)} \bmod x^r \tag{38}$$

where $\sigma(x) = \sigma_0(x)\,\sigma_1(x)$, and if the vector $\mathbf{R}' = [R'_0, \ldots, R'_{n-1}]$ is defined to be the inverse DFT of $\mathbf{S}$, i.e.,

$$R'_i = \frac{1}{n}\sum_{j=0}^{n-1} S_j \alpha^{-ij} \qquad \text{for } i = 0, \ldots, n-1 \tag{39}$$

and if $\mathbf{R}$ is defined by

$$R_i = \begin{cases} R'_i & \text{if } i \notin I_0 \\ * & \text{if } i \in I_0 \end{cases} \tag{40}$$

then if the RS decoding algorithm is applied to $\mathbf{R}$, (using the components of $\mathbf{R}'$ to fill in the erasures) $\sigma_1(x)$ and $\omega(x)$ will

be the error locator and error-and-erasure evaluator polynomials. Similarly, if $\gcd(\sigma_1, \omega) = x^j$ with $j > 0$, any **S** satisfying

$$S_L + S_{L+1}x + \cdots + S_{L+r-j-1}x^{r-j-1} = \frac{\omega(x)}{\sigma(x)} \bmod x^{r-j} \tag{41a}$$

$$S_L + S_{L+1}x + \cdots + S_{L+r-j-1}x^{r-j-1} + S_{L+r-j}x^{r-j} \neq \frac{\omega(x)}{\sigma(x)} \bmod x^{r-j+1} \tag{41b}$$

again with $\mathbf{R}'$ and **R** defined as in Eqs. (39) and (40), will do.

**Proof**: We distinguish two cases: $\gcd(\sigma_1, \omega) = 1$ and $\gcd(\sigma_1, \omega) \neq 1$. If $\gcd(\sigma_1, \omega) = 1$, and $S_0(x)$ is defined to be the polynomial $\omega(x)/\sigma_1(x) \bmod x^r$, then by Theorem C-4 in Appendix C and the example following it, $(\sigma_1, \omega)$ is the solution to the $(x^r, S_0(x), \mu, \nu)$ problem. Now if the decoding algorithm starts with **R**, and fills in the erasures to produce $\mathbf{R}'$, by Eq. (39) the syndrome polynomial $S(x)$ will be $S_L + S_{L+1}x + \cdots + S_{L+r-1}x^{r-1}$, which, by Eq. (38), is the same as $\omega(x)/\sigma(x) \bmod x^r$. Thus the modified syndrome $\sigma_0(x)\, S(x)$ will be $S_0(x) = \omega(x)/\sigma_1(x) \bmod x^r$, and, as we have seen, this means that the decoding algorithms will produce $s_1(x)$ and $\omega(x)$ as the error evaluator and error-and-erasure evaluator polynomials. The case when $\gcd(\sigma_1, \omega) \neq 1$ is handled similarly. ■

**Corollary**. If $t_0 \leqslant k$, then there is a vector $\mathbf{R}_0$ that produces $(\sigma_1, \omega)$ as the error locator polynomial and error-and-erasure evaluator with zeros as the erasure fills.

**Proof**: Let **R** be the vector defined by Eq. (40), and let **C** be any RS codeword that agrees with **R** on the set $I_0$. (There will be such a codeword, by the basic combinatorial property of RS codes, since $k > t_0$.) Then the vector $\mathbf{R} = \mathbf{R}_0 - \mathbf{C}$ will have the same syndrome as $\mathbf{R}_0$, and has zeros on the erasure set $I_0$. ■

# References


[1] E. R. Berlekamp, *Algebraic Coding Theory*, Laguna Hills, California: Aegean Park Press, 1984. (Reprint of the 1968 McGraw-Hill original).

[2] R. E. Blahut, *Theory and Practice of Error Control Codes*, Reading, Massachusetts: Addison-Wesley, 1983.

[3] R. J. McEliece, *The Theory of Information and Coding*, Reading, Massachusetts: Addison-Wesley, 1977.

[4] R. J. McEliece and L. Swanson, "On the Decoder Error Probability for Reed-Solomon Codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 145–158, 1986.

[5] I. S. Reed, T. K. Troung, and R. L. Miller, "Simplified Algorithm for Correcting Both Errors and Erasures of Reed-Solomon Codes," *Proc. IEE*, vol. 126, No. 10, pp. 961–963, October 1979.

[6] D. Whiting, "Bit Serial Reed-Solomon Decoders in VLSI," Ph.D. thesis, California Institute of Technology, 1984.

[7] J. Yuen (ed.), *Deep Space Telecommunications Systems Engineering*, JPL Publication 82-76, Jet Propulsion Laboratory, Pasadena, California, 1982.

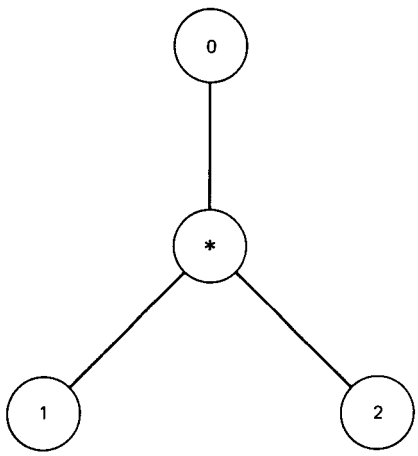


**Fig. 1. Illustration of the RS decoding metric for GF(3).**

# Appendix A
# The Discrete Fourier Transform

Let $F$ be a field which contains a primitive $n$th root of unity $\alpha$. (If the characteristic of $F$ is finite, we assume that it does not divide $n$.) We first note

$$1 - x^n = \prod_{i=0}^{n-1} (1 - \alpha^i x) \tag{A-1}$$

This is because the polynomials on both sides of Eq. (A-1) have degree $n$, constant term 1, and roots $\alpha^{-i}$, for $i = 0, 1, \ldots, n-1$.

Next, let

$$\mathbf{V} = (V_0, V_1, \ldots, V_{n-1}) \tag{A-2}$$

be an $n$-dimensional vector over $F$, and let

$$\widehat{\mathbf{V}} = (\widehat{V}_0, \widehat{V}_1, \ldots, \widehat{V}_{n-1}) \tag{A-3}$$

be its discrete Fourier transform (DFT), defined by

$$\widehat{V}_j = \sum_{i=0}^{n-1} V_i \alpha^{ij} \qquad \text{for } j = 0, 1, \ldots, n-1 \tag{A-4}$$

The components of $\mathbf{V}$ can be recovered from those of $\widehat{\mathbf{V}}$ via the inverse DFT

$$V_i = \frac{1}{n} \sum_{j=0}^{n-1} \widehat{V}_j \alpha^{-ij} \qquad \text{for } i = 0, 1, \ldots, n-1 \tag{A-5}$$

If we interpret the components of $\mathbf{V}$ and $\widehat{\mathbf{V}}$ as the coefficients of polynomials, i.e., if we define

$$V(x) = V_0 + V_1 x + \cdots + V_{n-1} x^{n-1} \tag{A-6}$$

and

$$\widehat{V}(x) = \widehat{V}_0 + \widehat{V}_1 x + \cdots + \widehat{V}_{n-1} x^{n-1} \tag{A-7}$$

then the DFT and IDFT relationships, Eqs. (A-4) and (A-5) become

$$\widehat{V}_j = V(\alpha^j) \tag{A-8}$$

and

$$V_i = \frac{1}{n} \widehat{V}(\alpha^{-i}) \tag{A-9}$$

# Appendix B

# Some Important Polynomials and the Fundamental Identity

Throughout this section $I$ will denote a fixed subset of $\{0, 1, \ldots, n-1\}$. We associate several polynomials with this set. For example, the *locator polynomial* for $I$ is

$$\sigma_I(x) = \prod_{i \in I} (1 - \alpha^i x) \tag{B-1}$$

The *co-locator polynomial* is

$$\tau_I(x) = \prod_{i \notin I} (1 - \alpha^i x) \tag{B-2}$$

In view of Eq. (A-1) we plainly have

$$1 - x^n = \sigma_I(x)\,\tau_I(x) \tag{B-3}$$

For each value of $i \in I$ we also define

$$\sigma_I^i(x) = \frac{\sigma_I(x)}{(1 - \alpha^i x)} = \prod_{\substack{j \in I \\ j \neq i}} (1 - \alpha^j x) \tag{B-4}$$

Finally, let $\mathbf{V} = (V_0, V_1, \ldots, V_{n-1})$ be a vector, such that $I$ is a *support set* for $\mathbf{V}$, i.e., $V_i = 0$ if $i \notin I$. Then the $(\mathbf{V}, I)$ *evaluator polynomial* is defined as

$$\omega_{\mathbf{V},I}(x) = \sum_{i \in I} V_i \sigma_I^i(x) \tag{B-5}$$

We will need several lemmas about these polynomials.

**Lemma B-1.** For all $i, j \in I$,

$$\sigma_I^i(\alpha^{-j}) = \begin{cases} 0 & \text{if } j \neq i \\ \prod_{\substack{k \in I \\ k \neq i}} (1 - \alpha^{k-i}) & \text{if } j = i \end{cases}$$

In particular, $\sigma_I^I(\alpha^{-i}) \neq 0$.

**Proof:** Follows immediately from Eq. (B-4). ■

**Lemma B-2.** If $i \in I$, $\omega_{\mathbf{V},I}(\alpha^{-i}) = V_i \sigma_I^i(\alpha^{-i})$. In particular, $\omega_{\mathbf{V},I}(\alpha^{-i}) = 0$ if and only if $V_i = 0$.

**Proof:** This follows from Eq. (B-5) and Lemma B-1. ■

**Lemma B-3.** The polynomials $\sigma_I^i(x)$ are linearly independent, and therefore form a basis for the set of all polynomials of degree $< |I|$.

**Proof:** If

$$\sum_{i \in I} \lambda_i \sigma_I^i(x) = 0$$

on setting $x = \alpha^{-i}$, we would get by Lemma B-1, $\lambda_i \sigma_I^i(\alpha^{-i}) = 0$, but since (again by Lemma B-1) $\sigma_I^i(\alpha^{-i}) \neq 0$, this implies that $\lambda_i = 0$. The last statement of the lemma now follows from the facts that (a) each $\sigma_I^i(x)$ has degree exactly $|I| - 1$ and (b) there are exactly $|I|$ of them. ■

The next lemma deals with the *minimal* support set $I(\mathbf{V})$ of $\mathbf{V}$, which is defined by

$$I(\mathbf{V}) = \{i \in I : V_i \neq 0\} \tag{B-6}$$

In what follows, the corresponding polynomials will be denoted by $\sigma_{\mathbf{V}}(x)$, $\tau_{\mathbf{V}}(x)$, and $\omega_{\mathbf{V}}(x)$, rather than $\sigma_{I(\mathbf{V})}(x)$, etc.

**Lemma B-4.**

$$\gcd(\sigma_I(x), \omega_{\mathbf{V},I}(x)) = \prod_{I - I(\mathbf{V})} (1 - \alpha^i x)$$

In particular, $\gcd(\sigma_{\mathbf{V}}(x), \omega_{\mathbf{V}}(x)) = 1$.

**Proof:** If $\omega_{\mathbf{V},I}(x)$ had a factor in common with $\sigma_I(x)$, then by Eq. (B-3) $\omega_{\mathbf{V},I}(\alpha^{-i}) = 0$ for some $i \in I$. But by Lemma B-2, this is true if and only if $V_i = 0$, i.e., if $i \in I - I(\mathbf{V})$. ■

We note that for any $\mathbf{V}$ and support set $I$, $\sigma_I(x)$ divides $1 - x^n$ and $\deg \omega_{\mathbf{V},I}(x) < \deg \sigma_I(x)$. The following theorem is a kind of a converse to this.

**Theorem B-5.** Suppose $\sigma(x)$ and $\omega(x)$ are polynomials such that $\sigma(x) | 1 - x^n$ and $\deg(\omega) < \deg(\sigma)$. Then there exists a vector $\mathbf{V}$ and a support set $I$ for $\mathbf{V}$ such that

$$\sigma(x) = \lambda \sigma_I(x) \tag{B-7}$$

$$\omega(x) = \lambda\omega_{\mathbf{V},I}(x) \tag{B-8}$$

for a nonzero constant $\lambda$. Furthermore, if in addition $\gcd(\sigma(x), \omega(x)) = 1$, then in fact there exists a vector $\mathbf{V}$ such that

$$\sigma(x) = \lambda\sigma_{\mathbf{V}}(x) \tag{B-9}$$

$$\omega(x) = \lambda\omega_{\mathbf{V}}(x) \tag{B-10}$$

**Proof:** Suppose $\sigma(x) \mid 1 - x^n$. Then Eq. (B-7) must hold for some subset $I$ of $\{0, 1, \ldots, n-1\}$ and some nonzero constant $\lambda$. Since $\deg(\omega) < \deg(\sigma) = \deg(\sigma_I)$, and since the polynomials $\sigma_I^i(x)$ are linearly independent by Lemma B-3,

$$\omega(x) = \lambda \sum_{i\in I} u_i \sigma_I^i(x) \tag{B-11}$$

for certain constants $u_i$. Thus if we define

$$V_i = \begin{cases} u_i & \text{if } i \in I \\ 0 & \text{if } i \notin I \end{cases} \tag{B-12}$$

Eq. (B-10) follows on comparing Eq. (B-11) to Eq. (B-5). Finally, by Lemma B-4, $\gcd(\sigma_I(x), \omega_{\mathbf{V},I}(x)) = 1$ if and only if $I = I(\mathbf{V})$, and so if $\gcd(\sigma(x), \omega(x)) = 1$, Eqs. (B-7) and (B-8) become Eqs. (B-9) and (B-10). ■

The next theorem is the most important result in this section.

**Theorem B-6.** If $I$ is a support set for $\mathbf{V}$, then the polynomials $V(x)$, $\sigma_I(x)$, and $\omega_{\mathbf{V},I}(x)$ satisfy

$$\sigma_I(x)\,\widehat{V}(x) = \omega_{\mathbf{V},I}(x)\,(1 - x^n) \tag{B-13}$$

**Proof:** Using the definitions in Eqs. (A-4) and (A-7), together with the fact that $I$ is a support set for $\mathbf{V}$, we find that

$$\widehat{\mathbf{V}}(x) = \sum_{i\in I} V_i \sum_{j=0}^{n-1} x^j \alpha^{ij} \tag{B-14}$$

According to Eq. (B-4), $\sigma_I(x) = \sigma_I^i(x)\,(1 - \alpha^i x)$ for all $i \in I$, and so from Eq. (B-14) we have

$$\sigma_I(x)\,\widehat{\mathbf{V}}(x) = \sum_{i\in I} V_i \sigma_I^i(x)\,(1 - \alpha^i x) \sum_{j=0}^{n-1} x^j \alpha^{ij}$$

$$= \sum_{i\in I} V_i \sigma_I^i(x)\,(1 - x^n)$$

$$= \omega_{\mathbf{V},I}(x)\,(1 - x^n)$$ ■

The following Corollary to Theorem B-6 tells us how to reconstruct the nonzero components of $\mathbf{V}$ from $\sigma_I(x)$ and $\omega_{\mathbf{V},I}(x)$. It involves the *formal derivative* $\sigma_I'(x)$ of the polynomial $\sigma_I(x)$.

**Corollary B-7.** If $I$ is a support set for $\mathbf{V}$, then for each $i \in I$, we have

$$V_i = -\alpha^i \frac{\omega_{\mathbf{V},I}(\alpha^{-i})}{\sigma_I'(\alpha^{-i})} \tag{B-15}$$

**Proof:** If we differentiate the fundamental identity in Eq. (B-13) we obtain

$$\sigma_I(x)\,\widehat{V}'(x) + \sigma_I'(x)\,\widehat{V}(x) = \omega_{\mathbf{V},I}(x)\,(-nx^{n-1}) + \omega_{\mathbf{V},I}'(x)\,(1 - x^n) \tag{B-16}$$

Note that if $x = \alpha^{-i}$ with $i \in I$, from Eqs. (B-3) and (A-1) we see that both $\sigma_I(x)$ and $1 - x^n$ vanish. Thus if $x = \alpha^{-i}$, Eq. (B-16) becomes

$$\sigma_I'(\alpha^{-i})\,\widehat{V}(\alpha^{-i}) = -n\alpha^i \omega_{\mathbf{V},I}(\alpha^{-i}) \tag{B-17}$$

But from Eq. (A-9), $\widehat{V}(\alpha^{-i}) = nV_i$. This fact, combined with Eq. (B-17), completes the proof. ■

**Corollary B-8.** $\gcd(\widehat{V}(x), 1 - x^n) = \tau_{\mathbf{V}}(x)$.

**Proof:** From Eq. (B-3) with $I = I(\mathbf{V})$, we have $1 - x^n = \sigma_{\mathbf{V}}(x)\,\tau_{\mathbf{V}}(x)$. Then, if we divide both sides of Eq. (B-13) by $\sigma_{\mathbf{V}}(x)$, we get $\widehat{V}(x) = \omega_{\mathbf{V}}(x)\,\tau_{\mathbf{V}}(x)$. Since by Lemma B-4, $\gcd(\sigma_{\mathbf{V}}(x), \omega_{\mathbf{V}}(x)) = 1$, the Corollary follows. ■

Now we can prove a kind of converse to Theorem B-6.

**Theorem B-9.** Suppose that the vector $\mathbf{V}$ is given, and that for certain polynomials $\sigma(x)$ and $\omega(x)$ we have

$$\sigma(x)\,\widehat{\mathbf{V}}(x) = \omega(x)\,(1 - x^n) \tag{B-18}$$

Then there exists a polynomial $\lambda(x)$ such that

$$\sigma(x) = \lambda(x)\,\sigma_V(x) \tag{B-19}$$

$$\omega(x) = \lambda(x)\,\omega_V(x) \tag{B-20}$$

**Proof**: By Eq. (B-18) we have

$$\sigma(x)\,\hat{V}(x) \equiv 0 \pmod{1 - x^n} \tag{B-21}$$

This implies that

$$\sigma(x) \equiv 0 \left(\mod \frac{1 - x^n}{\gcd(1 - x^n, \hat{V}(x))}\right) \tag{B-22}$$

But by Corollary B-8, $\gcd(1 - x^n, \hat{V}(x)) = \tau_V(x)$, and from Eq. (B-3),

$$\frac{(1 - x^n)}{\tau_V(x)} = \sigma_V(x)$$

and so Eq. (B-22) is equivalent to Eq. (B-19), for a suitable polynomial $\lambda(x)$. Then Eq. (B-18) becomes

$$\lambda(x)\,\sigma_V(x)\,\hat{V}(x) = \omega(x)\,(1 - x^n) \tag{B-23}$$

but multiplying Eq. (B-13) by $\lambda(x)$ we obtain

$$\lambda(x)\,\sigma_V(x)\,\hat{V}(x) = \lambda(x)\,\omega_V(x)\,(1 - x^n) \tag{B-24}$$

Comparing Eq. (B-23) to Eq. (B-24), we see that

$$\omega(x) = \lambda(x)\,\omega_V(x) \tag{B-25}$$

as asserted. This completes the proof of Theorem B-9. ■

The next results in this section deal with homogeneous difference equations (HDEs). We say that the infinite sequence $u_0, u_1, \ldots$ satisfies a $d$th-order HDE if there exist constants $\sigma_0, \ldots, \sigma_d$, with $\sigma_0 \neq 0$ and $\sigma_d \neq 0$ such that

$$\sum_{k=0}^{d} \sigma_k u_{j-k} = 0 \qquad \text{for } j \geq d \tag{B-26}$$

The polynomial $\sigma(x) = \sigma_0 + \cdots + \sigma_d x^d$ is called the *characteristic polynomial* of the HDE, and the degree $d$ of $\sigma(x)$ is called its *order*. If we define

$$\omega_j = \sum_{k=0}^{j} \sigma_k u_{j-k} \qquad \text{for } j = 0, \ldots, d-1 \tag{B-27}$$

and $\omega(x) = \omega_0 + \cdots + \omega_{d-1} x^{d-1}$, then it follows from Eqs. (B-26) and (B-27) that $(u_j)$ satisfies an HDE with characteristic polynomial $\sigma(x)$ if and only if

$$\sigma(x)\,u(x) = \omega(x) \tag{B-28}$$

where $\deg(\omega(x)) < \deg(\sigma(x))$. In particular, the sequence $(u_j)$ is periodic of period $n$, i.e., $u_j = u_{j-n}$ for $j \geq n$, if and only if there is a polynomial $\Omega(x)$ of degree $< n$ such that

$$(1 - x^n)\,u(x) = \Omega(x) \tag{B-29}$$

where $\deg \Omega < n$. The following theorem is needed in the discussion of the frequency-domain decoder. It assumes that $(u_j)$ is a sequence that satisfies a $d$th-order HDE with characteristic polynomial $\sigma(x)$, as described by Eq. (B-28).

**Theorem B-10.** If $\sigma(x)$ divides $1 - x^n$, then the sequence $(u_j)$ has period $n$. Conversely, if $(u_j)$ has period $n$ and if $\sigma(x) = \sigma_0(x)\,\sigma_1(x)$, where $\sigma_0(x)$ divides $1 - x^n$ and $\gcd(\sigma_1(x), \omega(x)) = 1$, then $\sigma(x)$ divides $1 - x^n$. In particular, if $\gcd(\sigma(x), 1 - x^n) = 1$, then $\sigma(x) \mid 1 - x^n$.

**Proof**: If $\sigma(x)$ divides $1 - x^n$, then $1 - x^n = \sigma(x)\,\tau(x)$ for some polynomial $\tau(x)$. If we multiply both sides of Eq. (B-28) by $\tau(x)$, we obtain $(1 - x^n)\,u(x) = \omega(x)\,\tau(x)$. But $\deg(\omega(x) \cdot \tau(x)) < \deg(\sigma(x)\,\tau(x)) = n$, and so by Eq. (B-29) $(u_j)$ has period $n$.

Conversely, if Eq. (B-29) holds, and we multiply Eq. (B-28) by $1 - x^n$ and Eq. (B-29) by $\sigma(x)$, then we find that $\Omega(x) \cdot \sigma(x) = \omega(x)\,(1 - x^n)$. Therefore $\sigma(x) \mid \omega(x)\,(1 - x^n)$. Since $\sigma(x) = \sigma_0(x)\,\sigma_1(x)$ and $\sigma_0(x) \mid 1 - x^n$, it follows that $\sigma_1(x) \mid \omega(x)\,(1 - x^n)/\sigma_0(x)$. But $\gcd(\sigma_1(x), \omega(x)) = 1$, and this means that $\sigma_1(x) \mid (1 - x^n)/\sigma_0(x)$, which implies that $\sigma(x) = \sigma_0(x)\,\sigma_1(x) \mid 1 - x^n$. ■

Having briefly discussed homogeneous difference equations, we are now in a position to discuss *circular homogeneous difference equations* (CHDEs). We say that the finite sequence $(u_0, \ldots, u_{n-1})$ satisfies a $d$th-order CHDE if there are constants $\sigma_0, \sigma_1, \ldots, \sigma_d$ with $\sigma_0 \neq 0$ and $\sigma_d \neq 0$, such that

$$\sum_{k=0}^{d} \sigma_k u_{j-k} = 0 \qquad \text{for } j = 0, \ldots, n-1 \tag{B-30}$$

where the subscripts must be interpreted mod $n$. The polynomial $\sigma(x) = \sigma_0 + \sigma_1 x + \cdots + \sigma_d x^d$ is called the characteristic polynomial of the CHDE, and $d$ is its order. If we define

$u(x) = u_0 + u_1 x + \cdots + u_{n-1}x^n$, then Eq. (B-30) holds if and only if

$$\sigma(x)\,u(x) \equiv 0 \pmod{1 - x^n} \tag{B-31}$$

Equivalently, $(u_0, \ldots, u_{n-1})$ satisfies a CHDE if and only if there is a polynomial $\omega(x)$ such that

$$\sigma(x)\,u(x) = \omega(x)\,(1 - x^n) \tag{B-32}$$

Plainly, the $\sigma(x)$ of smallest degree such that Eq. (B-32) holds is

$$\sigma_{\min}(x) = \frac{1 - x^n}{\gcd(u(x), 1 - x^n)} \tag{B-33}$$

which is a divisor of $1 - x^n$. Thus Theorem B-6 says that $\hat{\mathbf{V}}$ satisfies a CHDE of order $|I|$, where $I$ is any support set for $\mathbf{V}$. Conversely, Theorem B-8 says that $\hat{\mathbf{V}}$ does not satisfy a CHDE of order lower than $|I(\mathbf{V})|$. But we know from Eq. (B-6) that $|I(\mathbf{V})|$ = weight $(\mathbf{V})$ and so we have proved Theorem B-11.

**Theorem B-11.** The weight of $\mathbf{V}$ is the degree of the least-order CHDE satisfied by $\hat{\mathbf{V}}$.

# Appendix C
# The $(a(x), b(x), \mu, \nu)$ Problem

Given polynomials $a(x)$, $b(x)$, with $\deg(b) < \deg(a) = m$, and nonnegative integers $\mu$, $\nu$ with $\mu + \nu = m - 1$, consider the set $S = S(a, b, \mu, \nu)$ of all pairs of polynomials $(\sigma(x), \omega(x))$ such that

$$\deg(\sigma) \leqslant \mu \qquad \deg(\omega) \leqslant \nu \tag{C-1}$$

$$\sigma(x)\, b(x) \equiv \omega(x) \pmod{a(x)} \tag{C-2}$$

**Theorem C-1.** If $\mu + \nu = m - 1$, the set $S(a, b, \mu, \nu)$ is not empty. Indeed, there exists a pair $(\sigma_0, \omega_0) \in S(a, b, \mu, \nu)$ such that every pair $(\sigma(x), \omega(x)) \in S(a, b, \mu, \nu)$ is of the form

$$\sigma(x) = k(x)\, \sigma_0(x) \tag{C-3}$$

$$\omega(x) = k(x)\, \omega_0(x) \tag{C-4}$$

Furthermore, $(\sigma_0, \omega_0)$ is unique up to multiplication by scalars. We summarize this by saying that $(\sigma_0, \omega_0)$ "solves the $(a(x), b(x), \mu, \nu)$ problem."

**Proof:** A proof is given in [3], Theorem 8.5, where it is also pointed out that Euclid's algorithm can be used to find $(\sigma_0, \omega_0)$. Specifically, if one applies Euclid's algorithm as described there to the pair $(a(x), b(x))$, and stops when the degree of the remainder $r_j(x)$ becomes $\leqslant \nu$ for the first time, then $(t_j(x), r_j(x))$ is the solution to the $(a(x), b(x), \mu, \nu)$ problem. ■

**Lemma C-2.** $(\sigma, \omega) \in S(a, b, \mu, \nu)$ solves the $(a, b, \mu, \nu)$ problem if and only if there exists a polynomial $\tau(x)$ such that

$$\omega = \sigma b + \tau a \tag{C-5}$$

where

$$\gcd(\omega, \sigma, \tau) = 1 \tag{C-6}$$

**Proof:** Suppose that $(\sigma, \omega)$ solves the problem. Then by Eq. (C-2), there exists a polynomial $\tau(x)$ such that Eq. (C-5) holds. If $\omega(x)$, $\sigma(x)$, and $\tau(x)$ had a common factor $d(x)$, then with $\omega' = \omega/d$, $\sigma' = \sigma/d$, and $\tau' = \tau/d$, Eq. (C-5) implies $\omega' = \sigma' b + \tau' a$, which means $\sigma' b \equiv \omega' \pmod{a}$ is a smaller degree solution to Eqs. (C-1) and (C-2), contradicting the minimality of $(\sigma, \tau)$.

Conversely, suppose Eqs. (C-5) and (C-6) hold, and that $(\sigma_0, \omega_0)$ solves the $(a, b, \mu, \nu)$ problem. Then by Theorem C-1, $\omega = k\omega_0$, $\sigma = k\sigma_0$, and Eq. (C-5) becomes

$$k\omega_0 = k\sigma_0 b + \tau a \tag{C-7}$$

But since $(\sigma_0, \omega_0) \in S(a, b, \mu, \nu)$, we know that $\omega_0 = \sigma_0 b + \tau_0 a$ for some polynomial $\tau_0(x)$, and so we have

$$k\omega_0 = k\sigma_0 b + k\tau_0 a \tag{C-8}$$

Comparing Eqs. (C-7) and (C-8), we see that $\tau = k\tau_0$, and so $k \mid \gcd(\omega_1, \sigma_1, \tau_1)$. This implies by Eq. (C-6) that $k$ is a scalar and so $(\sigma, \omega)$ is a scalar multiple of $(\sigma_0, \omega_0)$, i.e., $(\sigma_1, \omega_1)$ solves the $(a, b, \mu, \nu)$ problem. ■

**Lemma C-3.** If $(\sigma_0, \omega_0)$ solves the $(a, b, \mu, \nu)$ problem, then

$$\gcd(\sigma_0, \omega_0) = \gcd(\sigma_0, a) \tag{C-9}$$

**Proof:** By Lemma C-2 we know that

$$\omega_0 = \sigma_0 b + \tau_0 a \tag{C-10}$$

with

$$\gcd(\omega_0, \sigma_0, \tau_0) = 1 \tag{C-11}$$

Now by Eq. (C-10) any common divisor of $\sigma_0$ and $a$ must divide $\omega_0$, i.e., $\gcd(\sigma_0, a) \mid \gcd(\sigma_0, \omega_0)$. On the other hand, Eq. (C-10) also says that any common divisor of $\sigma_0$ and $\omega_0$ must divide $\tau_0 a$ and so by Eq. (C-11) must divide $a$. Thus $\gcd(\sigma_0, \omega_0) \mid \gcd(\sigma_0, a)$. ■

**Theorem C-4.** Conversely, given polynomials $a(x)$, $\sigma_0(x)$, and $\omega_0(x)$ such that Eqs. (C-1) and (C-9) hold, there exists a polynomial $b(x)$ of degree $\leqslant m - 1$ such that $(\sigma_0, \omega_0)$ solves the $(a(x), b(x), \mu, \nu)$ problem.

**Proof:** Let $d(x) = \gcd(\sigma_0(x), \omega_0(x)) = \gcd(\sigma_0(x), a(x))$, and $\sigma_1 = \sigma_0/d$, $\omega_1 = \omega_0/d$, $a_1 = a/d$. Since $\gcd(\sigma_1, a_1) = 1$ we can define

$$b' = \frac{\omega_1}{\sigma_1} \bmod a_1 \tag{C-12}$$

It follows that

$$\sigma_1 b' \equiv \omega_1 \pmod{a_1} \tag{C-13}$$

i.e.,

$$\omega_1 = \sigma_1 b' + \tau_1 a_1 \tag{C-14}$$

for a suitable polynomial $\tau_1(x)$. We note that

$$\gcd(\sigma_1, \tau_1) = 1 \tag{C-15}$$

since a common factor of $\sigma_1$ and $\tau_1$ would by Eq. (C-14) also divide $\omega_1$; but $\gcd(\sigma_1, \omega_1) = 1$. We now distinguish two cases, according to whether $d$ and $\tau_1$ have a factor in common or not.

*Case 1:* $\gcd(d, \tau_1) = 1$. In this case if we multiply Eq. (C-14) by $d$ we obtain

$$\omega_0 = \sigma_0 b' + \tau_1 a \tag{C-16}$$

with $\gcd(\sigma_0, \omega_0, \tau_1) = \gcd(d, \tau_1) = 1$, and so by Lemma C-2, $(\sigma_0, \omega_0)$ solves the $(a, b', \mu, \nu)$ problem, where $b'$ is defined by Eq. (C-12).

*Case 2:* $\gcd(d, \tau_1) \neq 1$. In this case, let $\lambda(x)$ be the product of all the irreducible polynomials which divide $p$ but do not divide $\tau_1$, i.e.,

$$\lambda = \prod \{p : p \text{ is irreducible}, p \mid d, p \nmid \tau_1\} \tag{C-17}$$

Next, we define

$$\begin{aligned} b &= b' + \lambda a_1 \\ \tau_0 &= \tau_1 - \lambda \sigma_1 \end{aligned} \tag{C-18}$$

Then from Eq. (C-14) it follows that

$$\omega_1 = \sigma_1 b + \tau_0 a_1 \tag{C-19}$$

If $p$ is an irreducible divisor of $d$, then $p$ cannot divide $\tau_0$, because if $p \mid \tau_1$ then $p \nmid \sigma_1$ by Eq. (C-15) and $p \nmid \lambda$ by Eq. (C-17), so that $p \nmid \tau_0 = \tau_1 - \lambda\sigma_1$. On the other hand, if $p \nmid \tau_1$, then by Eq. (C-17) $p \mid \lambda$ and so again $p \nmid \tau_0 = \tau_1 - \lambda\sigma_1$. This shows that $\gcd(d, \tau_0) = 1$. Thus if we multiply Eq. (C-19) by $d$ we obtain

$$\omega_0 = \sigma_0 b + \tau_0 a \tag{C-20}$$

with $\gcd(\omega_0, \sigma_0, \tau_0) = \gcd(d, \tau_0) = 1$, and so by Lemma C-2, $(\sigma_0, \omega_0)$ solves the $(a, b, \mu, \nu)$ problem, with $b$ defined in Eq. (C-18). ■

**Example**: Suppose $a(x) = x^m$. Then the construction of Theorem C-4 simplifies considerably. Indeed, suppose we are given polynomials $\sigma_0(x)$ and $\omega_0(x)$ such that Eqs. (C-1) and (C-9) hold, with $a(x) = x^m$. Then $\gcd(\sigma_0, \omega_0) = \gcd(\sigma_0, x^m) = x^j$ for some value of $j$. If $b(x)$ is such that Eq. (C-2) holds, then dividing by $x^j$ we obtain

$$\sigma_1 b \equiv \omega_1 \pmod{x^{m-j}} \tag{C-21}$$

where $\gcd(\sigma_1, x^{m-j}) = 1$. Thus if $(\sigma_0, \omega_0)$ is to solve the $(x^m, b, \mu, \nu)$ problem, then it must be true that

$$b(x) \equiv \frac{\omega_1}{\sigma_1} \pmod{x^{m-j}} \tag{C-22}$$

If $j = 0$, i.e., if $\gcd(\sigma_0, \omega_0) = 1$, then Eq. (C-22) is both necessary and sufficient. On the other hand, if $j \geq 1$, it is easy to see that Lemma C-2 implies that $(\sigma_0, \omega_0)$ solves the $(x^m, b, \mu, \nu)$ problem if and only if Eq. (C-22) holds and if in addition

$$b(x) \not\equiv \frac{\omega_1}{\sigma_1} \pmod{x^{m-j+1}} \tag{C-23}$$

Thus if we expand $\omega_1/\sigma_1$ as a power series, viz.

$$\frac{\omega_1}{\sigma_1} = a_0 + a_1 x + \cdots \tag{C-24}$$

then any $b(x) = b_0 + b_1 x + \cdots + b_{m-1} x^{m-1}$ will do, provided that

$$\begin{aligned} b_k &= a_k \qquad \text{for } k = 0, \ldots, m-j-1 \\ b_{m-j} &\neq a_{m-j} \end{aligned} \tag{C-25}$$