

A Unifying System-Theoretic Framework for Errors-and-Erasures Reed-Solomon Decoding

Margreta Kuijper¹, Marten van Dijk², Henk Hollmann² and Job Oostveen²

¹ Dept. of EE Engineering, University of Melbourne, VIC 3010, Australia,
m.kuijper@ee.mu.oz.au

² Philips Research, 5656 AA Eindhoven, The Netherlands,
marten.van.dijk,henk.d.l.hollmann,job.oostveen@philips.com

Abstract. In the literature there exist several methods for errors-and-erasures decoding of RS codes. In this paper we present a unified approach that makes use of behavioral systems theory. We show how different classes of existing algorithms (e.g., syndrome based or interpolation based, non-iterative, erasure adding or erasure deleting) fit into this framework. In doing this, we introduce a slightly more general WB key equation and show how this allows for the handling of erasure locations in a natural way.

1 Introduction

Reed-Solomon (RS) codes find applications in storage and communication systems. Their algebraic structure has given rise to several low-complexity algorithms for error correction. The most well known are the Berlekamp-Massey (BM), the Euclidean and the Welch-Berlekamp (WB) algorithm.

The importance of having an errors-and-erasures correcting algorithm became truly apparent in the seminal paper [9] of G.D. Forney Jr., which presents a generalized minimum distance (GMD) decoding method which repeatedly employs errors-and-erasures decoding. In particular, efficient GMD decoding needs a fast iterative processing of Erasures, i.e., a fast way to obtain the solution for f erasures from the solution with either $f + 1$ or $f - 1$ erasures (named *erasure deletion* and *erasure addition*, respectively).

The decoding of corrupted RS code words boils down to solving a key equation. Classical key equations are the BM key equation and the WB key equation. Araki et al.[1] introduced the generalized key equation of which the classical ones are particular examples. As is shown in Section 2, these key equations can be reformulated in terms of behavioral modeling. Behavioral modeling has already been used to provide a good understanding of errors-only decoding [15–18].

The main contribution of this paper is in Section 3, where our approach straightforwardly gives rise to a range of errors-and-erasures decoding algorithms, and where we make connections with the existing literature. We unify several presently known iterative errors-and-erasures decoding algorithms in one conceptually

clear framework. We explain these algorithms and also give a new proof of the correctness of classical noniterative errors-and-erasures decoding in terms of behavioral modeling. Further, we generalize the WB key equation, which gives rise to a variant of the WB algorithm with which we can handle erasures.

2 General Framework

2.1 Preliminaries on RS Codes and Key Equations for Errors-Only Decoding

Let $\{x_1, \dots, x_n\}$ be a subset of a finite field \mathbb{F} with all x_i 's distinct. We define aRS code as a set of codewords of the form $\mathbf{c} = (M(x_1), \dots, M(x_n))$, where $M(x)$ is a polynomial of degree $< k$. RS codes are maximum distance separable (MDS), i.e. the minimum Hamming distance d of a (n, k) RS code equals $n - k + 1$. As a result, t errors and f erasures can be corrected if $2t + f \leq d - 1 = n - k$.

For decoding, the above definition naturally leads to the key equation

$$D(x_i)y_i = N(x_i)\eta_i \quad (1)$$

for $i = 0, \dots, n - k$. Here y_i and η_i are data derived from the received word—in this decoding context all η_i 's are nonzero and $y_{n-k} = 0$. The aim of errors-only decoding is to find polynomials $D(x)$ and $N(x)$ that satisfy (1) and for which $\deg N \leq \deg D$ and $\deg D$ is minimal. The error locations are then computed as the zeros of $D(x)$. A well known algorithm for solving this problem is the WB algorithm, which processes the interpolation data (x_i, y_i, η_i) iteratively for $i = 0, \dots, n - k$. In the literature η_i usually equals 1. In this paper, however, we prefer to leave η_i unspecified (possibly zero) as this allows us to incorporate erasure decoding in Section 3.

Alternatively, a RS code is defined as a set of codewords which have zeros at zero locations z_1, \dots, z_{n-k} . Here the zero locations are prespecified consecutive powers of a primitive element in \mathbb{F} . Decoding methods are then derived on the basis of the syndrome sequence $(S_1, \dots, S_{n-k}) := (r(z_1), \dots, r(z_{n-k}))$, where $r(x)$ denotes the received polynomial. A relevant equation is Berlekamp's classical key equation

$$A(x)S(x) = \Omega(x) \pmod{x^{n-k+1}} \quad (2)$$

Here $S(x) := S_1x + \dots + S_{n-k}x^{n-k}$ is the syndrome polynomial. The aim of errors-only decoding is to find polynomials $A(x)$ and $\Omega(x)$ that satisfy (2) and for which $A(0) \neq 0$ and $\max\{\deg A, \deg \Omega\}$ is minimal. The error locations are then computed as the reciprocals of the zeros of $A(x)$. This problem is solved by the BM algorithm which iteratively processes the syndrome components. Note that $A(x)$ corresponds to a shortest LFSR for the syndrome components S_1, \dots, S_{n-k} .

Both of the above described decoding methods are instances of polynomial interpolation. In the first method the interpolation points x_0, \dots, x_{n-k} are all distinct

whereas the second method performs repeated interpolation at one single point $x = 0$ (these originate from interpolation requirements on derivatives of the key equation). We denote the latter as interpolation at $(0, \{0, S_1, \dots, S_{n-k}\})$. This common interpolation aspect is exploited in recent work [5] by Blackburn who presents a generalized interpolation method that incorporates both types of interpolation.

2.2 Errors-Only Decoding of RS Codes in a Behavioral Framework

Formulation in terms of Behavioral Modeling. Here we recall how decoding in terms of the above key equations is reformulated as behavioral modeling of certain trajectories of time. Let us start with Berlekamp's classical key equation (2). From the syndromes S_1, \dots, S_d we define the trajectory $\mathbf{b} : \mathbb{Z}_+ \mapsto \mathbb{F}^2$ given by

$$\mathbf{b} = \left(\begin{bmatrix} S_{d-1} \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} S_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots \right) . \quad (3)$$

It can now be easily verified that $A(x)$ and $\Omega(x)$ are solutions of (2) if and only if the trajectory \mathbf{b} is a solution of the difference equation

$$[A(\sigma) \quad -\Omega(\sigma)] \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = 0 \quad (4)$$

in the variable $\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} : \mathbb{Z}_+ \mapsto \mathbb{F}^2$. Here σ stands for the backward shift operator.

Let us now consider the WB key equation (1). From the interpolation data we define d trajectories $\mathbf{b}_i : \mathbb{Z}_+ \mapsto \mathbb{F}^2$ given by

$$\mathbf{b}_i = \begin{bmatrix} y_i \\ \eta_i \end{bmatrix} (1, x_i, x_i^2, \dots) \quad \text{for } i = 0, \dots, d-1. \quad (5)$$

Clearly, the polynomials $D(x)$ and $N(x)$ are solutions of (1) if and only if all trajectories \mathbf{b}_i ($i = 0, \dots, d-1$) are solutions of the difference equation

$$[D(\sigma) \quad -N(\sigma)] \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = 0 . \quad (6)$$

For decoding we require in addition that the row degrees of $[A(x) \quad -\Omega(x)]$ and $[D(x) \quad -N(x)]$, respectively, are minimal. Here the row degree of a polynomial row vector is defined as the maximum degree of its entries. Furthermore, for decoding, we require $A(0) \neq 0$ for the solution of (4) and $\deg N \leq \deg D$ for the solution of (6). The fact that these requirements differ is solely due to the fact that Berlekamp's key equation (2) aims at reciprocals of error locations rather than at the locations themselves.

Remark 1. Note that, if we process the syndromes in a reversed order then the requirement that $A(0) \neq 0$ is to be replaced by the requirement that $\deg \Omega \leq \deg A$.

Having reformulated the two decoding problem statements in a behavioral setting, how do we go about solving it? A model of the form (4)

$$[\Lambda(\sigma) \quad -\Omega(\sigma)] \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = 0$$

clearly gives rise to a linear σ -invariant solution space (“behavior”) spanned by infinitely many trajectories from \mathbb{Z}_+ to \mathbb{F}^2 . For our decoding we require that this behavior contains the given trajectory \mathbf{b} , defined by (3). The smallest σ -invariant behavior \mathcal{B}^* that contains \mathbf{b} is clearly finite dimensional and given by the span of $\mathbf{b}, \sigma\mathbf{b}, \dots, \sigma^d\mathbf{b}$. This behavior \mathcal{B}^* is called the *Most Powerful Unfalsified Model (MPUM)* for the data set $\{\mathbf{b}\}$, see [23]. For \mathcal{B}^* we can immediately write down a representation, namely

$$\begin{bmatrix} 1 & -(S_1\sigma + \dots + S_{d-1}\sigma^{d-1}) \\ 0 & \sigma^d \end{bmatrix} \mathbf{w} = 0 . \quad (7)$$

The above representation is not unique—in fact, all other representations of \mathcal{B}^* can be obtained by left multiplying the matrix in (7) by a unimodular polynomial matrix, i.e. a polynomial matrix whose determinant is a nonzero constant. Note that it follows that the degree of the determinant of any matrix that represents \mathcal{B}^* equals $d = \dim \mathcal{B}^*$, whereas the sum of the row degrees of any such matrix is larger than or equal to d . It can be proven [23] that there exists a representation of \mathcal{B}^* for which equality holds. This representation has minimal row degrees and is called “row reduced”. A solution $[\Lambda(x) \quad -\Omega(x)]$ of the decoding problem is simply found by selecting from the two rows in a row reduced representation of \mathcal{B}^* the row of minimal degree that satisfies the additional requirement (here: $\Lambda(0) \neq 0$).

In the case of the WB key equation (1) the approach is completely analogous: simply replace $\{\mathbf{b}\}$ by $\{\mathbf{b}_0, \dots, \mathbf{b}_{d-1}\}$, defined in (5) and find a row reduced representation for its MPUM accordingly, see [18, 19]. In this case we choose the row of minimal degree that satisfies the additional requirement that $\deg N \leq \deg D$.

Algorithms. A well known noniterative algorithm for solving the above decoding problems is the Euclidean algorithm. In [18] it has been explained that the Euclidean algorithm simply brings the matrix in (7) in row reduced form.

Alternatively, the general iterative behavioral modeling procedure of [23, p. 289] can be used. For key equation (2) it is explained in detail in [15] how the BM algorithm can be interpreted as an instance of this procedure.

It has been shown in [18, 19] how the same general iterative behavioral modeling procedure of [23] can also be put to work to produce an iterative algorithm for solving key equation (1). The resulting algorithm closely resembles the WB algorithm but involves a different update parameter [18, Sect. 4.4]. It plays a key role in the sequel of this paper. We believe that the behavioral set-up

enables a particularly transparent explanation. For this reason we now explain the algorithm as clearly as possible, see also [18, Thm. 4.2].

Algorithm 1. As a first step we initialize

$$R_{-1}(x) := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} .$$

Note that the row degrees L_{-1}^1 and L_{-1}^2 of this matrix both equal 0. The behavior represented by $R_{-1}(\sigma)\mathbf{w} = 0$ equals $\{0\}$. We now proceed by processing the data (x_i, y_i, η_i) step by step. At step i ($i = 0, \dots, d-1$) we process the corresponding trajectory \mathbf{b}_i given by (5). For this, we first compute the error trajectory $\mathbf{e}_i := R_{i-1}(\sigma)\mathbf{b}_i$, which is easily shown to be of the form

$$\mathbf{e}_i = \begin{bmatrix} \Delta_i \\ \Gamma_i \end{bmatrix} (1, x_i, x_i^2, \dots) .$$

In fact, Δ_i and Γ_i are computed as

$$\begin{bmatrix} \Delta_i \\ \Gamma_i \end{bmatrix} := R_{i-1}(x_i) \begin{bmatrix} y_i \\ \eta_i \end{bmatrix} .$$

We then choose an update matrix $V_i(x)$ such that $V_i(\sigma)\mathbf{w} = 0$ represents the MPUM for $\{\mathbf{e}_i\}$. Defining $R_i(x) := V_i(x)R_{i-1}(x)$, we then have that $R_i(\sigma)\mathbf{w} = 0$ is a representation that models all data $\mathbf{b}_0, \dots, \mathbf{b}_i$ processed so far. We need to choose $V_i(x)$ carefully, so as to produce a row reduced matrix $R_i(x)$. Recall that this means that the sum of the first row degree L_i^1 and the second row degree L_i^2 of $R_i(x)$ equals the degree of the determinant of $R_i(x)$. This is achieved by making sure that only one of the row degrees of $R_{i-1}(x)$ is increased by one when left multiplied by $V_i(x)$. The following specification satisfies this requirement: if $(\Gamma_i \neq 0$ and $L_{i-1}^1 \geq L_{i-1}^2)$ or $\Delta_i = 0$ then

$$V_i(x) := \begin{bmatrix} \Gamma_i & -\Delta_i \\ 0 & x - x_i \end{bmatrix}; \quad L_i^1 := L_{i-1}^1 \quad \text{and} \quad L_i^2 := L_{i-1}^2 + 1$$

and, if otherwise,

$$V_i(x) := \begin{bmatrix} x - x_i & 0 \\ \Gamma_i & -\Delta_i \end{bmatrix} \quad L_i^1 := L_{i-1}^1 + 1 \quad \text{and} \quad L_i^2 := L_{i-1}^2 .$$

Note that for efficient implementation it is sufficient to update only L_i^1 since $L_i^1 + L_i^2 = i + 1$ at each step i . After processing all data (x_i, y_i, η_i) for $i = 0, \dots, d-1$, the matrix $R_d(x)$ is a row reduced representation of the MPUM \mathcal{B}^* of $\{\mathbf{b}_0, \dots, \mathbf{b}_{d-1}\}$. It can be proven that $R_{d-1}(x)$ also has the property that the degree of its lower left entry is strictly smaller than the degree of the lower right entry. From the row reducedness of $R_d(x)$ it then follows that the upper left entry $D(x)$ and the upper right entry $N(x)$ are a solution of key equation (1) for which $\deg N \leq \deg D$ and $\deg D$ is minimal.

Remark 2. The above algorithm can be easily adapted (see [5]) so as to process repeated interpolations at $x = 0$, say involving the reversed syndrome polynomial $S_{d-1}x + \dots + S_1x^{d-1}$. This straightforwardly gives rise to the algorithm of [18, sect. 4.2] which computes a polynomial whose zeros are the error locations rather than the reciprocals of the error locations, see Remark 1. In this case the discrepancies Δ_i and Γ_i are computed as

$$\begin{bmatrix} \Delta_i \\ \Gamma_i \end{bmatrix} := \text{coeff of } x^i \text{ in } R_{i-1}(x) \begin{bmatrix} S_{d-1}x + \dots + S_1x^{d-1} \\ 1 \end{bmatrix} .$$

3 Errors-and-Erasures RS Decoding

In this section we present various methods for errors-and-erasures decoding, most of which can be found in the literature. The main aim of this section is to cast all methods into one conceptually clear framework by reformulation in behavioral modeling terms.

3.1 Noniterative Processing of Erasures

Here we deal with a situation where $f < d$ erasure locations $\alpha_1, \alpha_2, \dots, \alpha_f$ are a priori specified, for example through the erasure locator polynomial $\Gamma(x) := \prod_{j=1}^f (1 - \alpha_j x)$. We seek to find the corresponding error values (possibly zero) as well as additional errors in the non-erased locations. Substituting zeros in the erased positions we first derive syndrome values S_1, \dots, S_{d-1} . Errors-and-erasures decoding amounts to finding the shortest LFSR $A(x)$ for S_1, \dots, S_{d-1} that contains $\Gamma(x)$ as a factor. Methods for solving this problem are well known and can be found in e.g. [6, 22]. In this subsection we first seek to reformulate the problem in behavioral modeling terms. We then outline how a range of different classical solution methods fits into our framework.

In terms of trajectories, the above requirement that $\Gamma(x)$ is a factor of the errors-and-erasures locator polynomial $A(x)$ is easily reformulated as the requirement to model not only the trajectory $\mathbf{b} : \mathbb{Z}_+ \mapsto \mathbb{F}^2$ given by (3), but also, for $j = 1, \dots, f$, the trajectories

$$\mathbf{b}_j := \begin{bmatrix} 1 \\ 0 \end{bmatrix} (1, \alpha_j^{-1}, \alpha_j^{-2}, \dots) . \quad (8)$$

With $S(x) := S_1x + \dots + S_{d-1}x^{d-1}$, a representation for the MPUM for the set of trajectories $\{\mathbf{b}, \mathbf{b}_1, \dots, \mathbf{b}_f\}$ is readily obtained as

$$\begin{bmatrix} \Gamma(\sigma) & -\bar{S}(\sigma) \\ 0 & \sigma^d \end{bmatrix} \mathbf{w} = 0,$$

where $\bar{S}(x) := \Gamma(x)S(x) \bmod x^d$ is the modified syndrome, see e.g. [22]. The task at hand is now simply to bring the matrix in the above equation in row reduced form. As described below, this can be done in a convenient

way by making use of the next lemma (whose proof is straightforward) and the decomposition $\bar{S}(x) = \bar{S}_1(x) + x^f \bar{S}_2(x)$, where $\bar{S}_1(x) := \bar{S}_1 x + \bar{S}_2 x^2 + \dots + \bar{S}_f x^f$ and $\bar{S}_2(x) := \bar{S}_{f+1} x + \dots + \bar{S}_{d-1} x^{d-1-f}$.

Lemma 1. *Let $a(x)$ and $b(x)$ be polynomials of degree f and let $c(x)$ be a polynomial of degree $\leq f$. Let $F(x)$ be a 2×2 polynomial matrix that is row reduced. Then*

$$F(x) \begin{bmatrix} a(x) & c(x) \\ 0 & b(x) \end{bmatrix}$$

is row reduced.

Theorem 1. *Let*

$$\begin{bmatrix} \Lambda(\sigma) & -\Omega(\sigma) \\ \lambda(\sigma) & -\omega(\sigma) \end{bmatrix} \mathbf{w} = 0$$

be a row reduced representation of the MPUM of the trajectory

$$\left(\begin{bmatrix} \bar{S}_{d-1} \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} \bar{S}_{f+1} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \dots \right).$$

Define

$$R(x) = \begin{bmatrix} \bar{\Lambda}(x) & -\bar{\Omega}(x) \\ \bar{\lambda}(x) & -\bar{\omega}(x) \end{bmatrix} := \begin{bmatrix} \Lambda(x) & -\Omega(x) \\ \lambda(x) & -\omega(x) \end{bmatrix} \begin{bmatrix} \Gamma(x) & -\bar{S}_1(x) \\ 0 & x^f \end{bmatrix}.$$

Then $R(\sigma)\mathbf{w} = 0$ is a row reduced representation of the MPUM of $\{\mathbf{b}, \mathbf{b}_1, \dots, \mathbf{b}_f\}$, as defined in (3) and (8).

Proof. Applying the above lemma for $a(x) = \Gamma(x)$, $b(x) = x^f$ and $c(x) = -\bar{S}_1(x)$, it follows that $R(x)$ is row reduced. It can also be easily seen that $R(\sigma)\mathbf{w} = 0$ represents the MPUM of $\{\mathbf{b}, \mathbf{b}_1, \dots, \mathbf{b}_f\}$. \square

Because of the above theorem we can perform errors-and-erasures decoding by computing the modified syndrome values $\bar{S}_{f+1}, \dots, \bar{S}_{d-1}$ and constructing a shortest LFSR for them. The latter can be done either noniteratively, by applying the Euclidean algorithm on the polynomials x^{d-f} and $\bar{S}_2(x)$ or iteratively by applying the BM algorithm on $\bar{S}_{f+1}, \dots, \bar{S}_{d-1}$. Both methods are classical and can be found in e.g. [6], see also [8]. The BM type method is essentially equivalent to the method recounted in [21, Sect. II-A] and [6, 13]: it can be easily verified that applying BM on $\bar{S}_{f+1}, \dots, \bar{S}_{d-1}$ is the same as applying BM on S_1, \dots, S_N and initializing with

$$\begin{bmatrix} \Gamma(x) & 0 \\ 0 & x \end{bmatrix}.$$

3.2 Iterative Processing of Erasures

Erasur Deletion through Interpolation at Distinct Points. The most natural way [2–4, 20] to deal with erasures is to employ an approach based on interpolation at the code locations. Indeed, in this approach the interpolation

points can be chosen complementary to the erasure locations, which are thus ignored (“erased”). In fact, we can regard the preliminary step of the WB algorithm, in which k entries are re-encoded, as a case of erasures-only decoding in which $n - k = d - 1$ code locations are erased. In each subsequent step of the WB algorithm one erasure is deleted from the full set of $d - 1$ erasures, until at the last $(d - 1)$ st step all erasures have been deleted and errors-only decoding is completed. Thus the WB algorithm and the closely related Algorithm 1 can be regarded as instances of an iterative errors-and-erasures decoding method in which erasures are successively deleted.

Syndrome-Based Erasure Addition. Alternatively, it is possible to formulate a syndrome-based errors-and-erasures decoding method that processes the erasures iteratively, as presented by Kötter in [14]. Indeed, the exposition in Section 3.1 is easily modified to reformulate decoding as the construction of a row reduced representation for the MPUM of the data set $\{\tilde{\mathbf{b}}, \tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_f\}$, where

$$\tilde{\mathbf{b}} := \left(\begin{bmatrix} S_1 \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} S_{d-1} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \dots \right), \quad (9)$$

and, for $j = 1, \dots, f$,

$$\tilde{\mathbf{b}}_j := \begin{bmatrix} 1 \\ 0 \end{bmatrix} (1, \alpha_j, \alpha_j^2, \dots) . \quad (10)$$

In the notation of Section 2.1, the decoding problem is thus an interpolation problem with interpolation data $(0, \{0, S_{d-1}, \dots, S_1\}), (\alpha_1, 1, 0), \dots, (\alpha_f, 1, 0)$. This approach is close to the work by Kötter [14] who, in behavioral terms, first constructs a row reduced representation for the syndromes, then takes its reciprocal model and proceeds by performing interpolation at the erasure locations. In our set-up we process the syndrome components in a reversed order so that a reciprocal model needs not be computed. Note that the order in which erasures are added is not important. In fact, erasures can even be added after any intermediate syndrome processing iteration, an observation which was also made in [13], where a similar algorithm is presented.

Syndrome-Based Erasure Deletion. In [21] Taipale and Seo employ an erasure deleting approach that is syndrome-based. Their algorithm produces a polynomial whose zeros are the reciprocals of the error locations. Below we present an algorithm which resembles the algorithm in [21] but produces a polynomial whose zeros are the error locations. We found that setting up the algorithm in this way rather than in the reciprocal domain enhances its insightfulness. Similar algorithms to ours have been presented in [10–12].

For our syndrome-based erasure deletion approach, we first consider erasures-only decoding, specifying $d - 1$ erasure locations $\alpha_1, \alpha_2, \dots, \alpha_{d-1}$ and defining $\tilde{F}(x) := \prod_{j=1}^{d-1} (x - \alpha_j)$. We initialize our algorithm with

$$R_0(x) = \begin{bmatrix} \tilde{F}(x) & -\tilde{S}(x) \\ 0 & x^d \end{bmatrix},$$

where $\tilde{S}(x) := \tilde{\Gamma}(x)(S_{d-1}x + \dots + S_1x^{d-1}) \bmod x^d$. Note that the representation

$$\begin{bmatrix} \tilde{\Gamma}(\sigma) & -\tilde{S}(\sigma) \\ 0 & \sigma^d \end{bmatrix} \mathbf{w} = 0$$

models $\{\tilde{\mathbf{b}}, \tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_f\}$, given by (9-10). Erasure deletion comes down to removing, one by one, the erasure trajectories $\tilde{\mathbf{b}}_j$ ($j = 1, \dots, d-1$). After erasing all $d-1$ erasures, the output of the algorithm achieves errors-only decoding. Not surprisingly, the algorithm operates inversely to Algorithm 1. For the sake of brevity we omit its proof here.

Algorithm 2. Initialize

$$R_0(x) := \begin{bmatrix} \tilde{\Gamma}(x) & -\tilde{S}(x) \\ 0 & x^d \end{bmatrix}; \quad L_0^1 := d-1 \quad \text{and} \quad L_0^2 := d .$$

At step i , process the erasure α_i ($i = 1, \dots, d-1$) by computing

$$\begin{bmatrix} \Delta_i \\ \Gamma_i \end{bmatrix} := R_{i-1}(\alpha_i) \begin{bmatrix} 0 \\ 1 \end{bmatrix} .$$

Then define $R_i(x) := V_i(x)R_{i-1}(x)$ where if ($L_{i-1}^1 \geq L_{i-1}^2$ and $\Gamma_i \neq 0$) or $\Delta_i = 0$, then

$$V_i(x) := \begin{bmatrix} \frac{\Gamma_i}{x-\alpha_i} & \frac{-\Delta_i}{x-\alpha_i} \\ 0 & 1 \end{bmatrix} \quad L_i^1 := L_{i-1}^1 - 1 \quad \text{and} \quad L_i^2 := L_{i-1}^2$$

and, if otherwise,

$$V_i(x) := \begin{bmatrix} 1 & 0 \\ \frac{\Gamma_i}{x-\alpha_i} & \frac{-\Delta_i}{x-\alpha_i} \end{bmatrix} \quad L_i^1 := L_{i-1}^1 \quad \text{and} \quad L_i^2 := L_{i-1}^2 - 1 .$$

Now, the zeros of the upper left entry of $R_i(x)$ are candidate error locations for errors-and-erasures decoding with $d-1-i$ erasures specified.

Note again that for efficient implementation only L_i^1 needs to be specified since $L_i^1 + L_i^2 = 2d-1-i$ at each step i .

4 Conclusions

In this paper we put behavioral systems theory to work to provide a unified explanation of a range of iterative errors-and-erasures decoding algorithms in the literature. In doing this, we introduced a slightly more general version of the WB key equation (by introducing the η_i in equation (1)) to accommodate the handling of erasure trajectories. We classified several known iterative procedures for errors-and-erasures RS decoding and gave an overview of the relationships between our framework and the currently known schemes.

References

1. Araki, K. and I. Fujita (1992). Generalized syndrome polynomials for decoding Reed-Solomon codes. *IEICE Trans. Fundamentals* **E75-A**, 1026-1029.
2. Araki, K., Takada, M. and M. Morii (1992). On the efficient decoding of Reed-Solomon codes based on GMD criterion. *Proc. 22nd Int. Symp. on Multiple Valued Logic*, 138-142.
3. Araki, K., Takada, M. and M. Morii (1993). The efficient GMD decoders for BCH codes. *IEICE Trans. Inform. and Systems* **E76-D**, 594-604.
4. Berlekamp, E.R. (1996). Bounded distance+1 soft-decision Reed-Solomon decoding. *IEEE Trans. Inform. Theory* **42**, 704-721.
5. Blackburn, S.R. (1997). A generalized rational interpolation problem and the solution of the WB algorithm. *Designs, Codes and Cryptography* **11**, 223-234.
6. Blahut, R.E. (1983). *Theory and Practice of Error Control Codes*, Addison-Wesley.
7. Elias, P. (1954). Error-free coding. *IRE Trans. Inform. Theory* **PGIT-4**, 29-37.
8. Fitzpatrick, P. (1995). On the key equation. *IEEE Trans. Inform. Theory* **41**, 1290-1302.
9. Forney, G.D., Jr. (1966). Generalized minimum distance decoding. *IEEE Trans. Inform. Theory* **12**, 125-131.
10. Fujisawa, M. and S. Sakata (1999). On fast generalized minimum distance decoding for algebraic codes. *Preproc. AAECC-13*, 82-83.
11. Kamiya, N. (1995). On multisequence shift register synthesis and generalized-minimum-distance decoding of Reed-Solomon codes. *Finite Fields Appl.* **1**, 440-457.
12. Kamiya, N. (1999). A unified algorithm for solving key equations for decoding alternant codes. *IEICE Trans. Fundamentals*, **E82-A**, 1998-2006.
13. Kobayashi, Y., Fujisawa, M. and S. Sakata (2000). Constrained shiftregister synthesis: fast GMD decoding of 1D algebraic codes. *IEICE Trans. Fundamentals* **83**, 71-80.
14. Kötter, R. (1996). Fast generalized minimum distance decoding of algebraic geometric and Reed-Solomon codes. *IEEE Trans. Inform. Theory* **42**, 721-738 .
15. Kuijper, M. and J.C. Willems (1997). On constructing a shortest linear recurrence relation. *IEEE Trans. Aut. Control* **42**, 1554-1558.
16. Kuijper, M. (1999). The BM algorithm, error-correction, keystreams and modeling. In *Dynamical Systems, Control, Coding, Computer Vision*, G. Picci, D. S. Gilliam (eds.), Birkhäuser
17. Kuijper, M. (1999). Further results on the use of a generalized BM algorithm for BCH decoding beyond the designed error-correcting capability, *Proc. 13th AAECC*, Hawaii, USA (1999), 98-99.
18. Kuijper, M. (2000). Algorithms for Decoding and Interpolation. In *Codes, Systems, and Graphical Models* , IMA Series Vol. 123, B. Marcus and J. Rosenthal (eds.), pp. 265-282, Springer-Verlag.
19. Kuijper, M. (2000). A system-theoretic derivation of the WB algorithm, in *Proc. IEEE International Symposium on Information Theory (ISIT'00)* p. 418.
20. Sorger, U. (1993). A new Reed-Solomon decoding algorithm based on Newton's interpolation. *IEEE Trans. Inform. Theory* **39**, 358-365.
21. Taipale, D.J. and M.J. Seo (1994). An efficient soft-decision Reed-Solomon decoding algorithm. *IEEE Trans. Info. Theory* **40**, 1130-1139.
22. Wicker, S.B. (1995). *Error control systems*, Prentice Hall.
23. Willems, J.C. (1991). Paradigms and puzzles in the theory of dynamical systems. *IEEE Trans. Aut. Control* **36**, 259-294.