Reed–Solomon List Decoding From a System-Theoretic Perspective

Margreta Kuijper and Jan Willem Polderman

***Abstract*—In this paper, the Sudan–Guruswami approach to list decoding of Reed–Solomon (RS) codes is cast in a system-theoretic framework. With the data, a set of trajectories or time series is associated which is then modeled as a so-called behavior. In this way, a connection is made with the behavioral approach to system theory. It is shown how a polynomial representation of the modeling behavior gives rise to the bivariate interpolating polynomials of the Sudan–Guruswami approach. The concept of "weighted row reduced" is introduced and used to achieve minimality. Two decoding methods are derived and a parametrization of all bivariate interpolating polynomials is given.**

***Index Terms*—Behaviors, bivariate interpolation, list decoding, Reed–Solomon (RS) codes, system theory.**

I. Introduction

A. Context

FOR several decades Reed–Solomon (RS) codes could be counted among the most frequently used block codes. Their algebraic structure allows for a range of hard-decision decoding algorithms, such as the Berlekamp–Massey (B-M) algorithm, the Euclidean algorithm, and the Welch–Berlekamp algorithm. All these classical decoding algorithms decode up to the conventional limit of half the minimum-distance number of errors. In more recent years, a novel approach toward RS decoding was introduced in [1]–[3] that opened up the possibility of decoding beyond this conventional limit. The approach employs list decoding in which a list of codewords that are nearest to the received word is produced. This major breakthrough was followed by several publications, such as [4]–[9]. In particular, it gave rise to innovative ideas on algebraic soft-decision RS decoding, as presented in [10]–[12].

Parallel to the developments described above, there has been a growing interest to establish relationships between the area of coding theory and the area of system theory, particularly the behavioral approach, see, e.g., [13], [14] as well as [15] and references therein. In the behavioral approach, the system theorist's focus is on the set of all possible time trajectories of a system, which is called the *behavior*. In previous work [16], [13], [17], [18], [14], [19], [20] classical RS decoding algorithms are formulated in terms of behavioral modeling. To effectuate the translation to system theory the decoding data is transformed into trajectories on the time axis $\mathbb{Z}_+$. Decoding is then shown to be equivalent to modeling of these trajectories. In this setup, autonomous behaviors play a key role. In the behavioral literature [21], an iterative procedure is available for building autonomous behaviors that requires the specification of an update matrix at each step. In [19] it is shown that the B-M algorithm is a special instance of this procedure with cleverly chosen update matrices. In a sense, this work gives a behavioral foundation to the matrix formulation of the B-M algorithm, as in Blahut [22].

In this paper, we seek to put the novel list decoding approach of [1]–[3] in a system-theoretic framework. Our motivation for this is threefold. First, this framework unifies different list decoding methods. We show that one of these, namely the Nielsen–Høholdt decoding algorithm from [8], can be interpreted as a special instance of the iterative modeling procedure of [21] where the update matrix is chosen cleverly (see also [23], [24] for a restricted case). In this paper, we derive a second decoding method which requires a postprocessing stage. In a sense, the postprocessing stage generalizes the Euclidean algorithm as used for classical decoding (see [25]).

Our second motivation for using a system-theoretic approach is that it naturally gives rise to decoding methods that produce more than just the sought after solution: it is shown that it also produces a parametrization of all solutions. We elaborate on this aspect of the approach in Section V.

A third motivation for using our system-theoretic framework is that we believe that it brings about conceptual clarity that facilitates the understanding of decoding algorithms. Some preliminary knowledge on system theory is of course required to achieve this understanding. This paper is meant to be self-contained. In fact, we aim to equip the reader with just enough system-theoretic knowledge to be able to understand the paper's contributions. In the remainder of this section, we present an informal first introduction to our system-theoretic approach to give the reader a first idea of the approach. This is followed by more stringent system-theoretic preliminaries in Section II. We pay specific attention to the impact of finite fields on behavioral results. The outline of the rest of the paper is as follows. In Section III, we formulate the decoding interpolation problem in behavioral terms as the modeling of certain trajectories. In Section IV, we present algorithms that perform multivariable interpolation through the iterative calculation of a square polynomial matrix $R(x)$. Solutions $Q(x, y)$ of minimal weighted degree are produced in this way. In addition, our approach provides a complete characterization of all such bivariate polynomials, as shown in Section V. Finally, conclusions are presented in Section VI.

Manuscript received January 16, 2003; revised October 14, 2003.

M. Kuijper is with the Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: m.kuijper@ee.mu.oz.au).

J. W. Polderman is with the Department of Applied Mathematics, University of Twente, 7500 AE Enschede, The Netherlands (e-mail: J.W.Polderman@math.utwente.nl).

Communicated by R. Koetter, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2003.822593

B. Introduction to a System-Theoretic Approach

As mentioned above, the translation to system theory is effectuated by reformulating decoding equations in terms of behavioral modeling. The basic idea is contained in the following. Consider an equation of the type

$$D(\lambda)\mu = N(\lambda) \tag{1}$$

where λ and μ are elements of a field $\mathbb{F}$ and $D(x)$ and $N(x)$ are polynomials over $\mathbb{F}$. In terms of bivariate polynomials we then have $Q(\lambda,\mu)=0$ for $Q(x,y) := D(x)y - N(x)$. With the data pair (λ,μ) we associate the trajectory $\boldsymbol{b}: \mathbb{Z}_+ \mapsto \mathbb{F}^2$ given by

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \mu \end{bmatrix} (1, \lambda, \lambda^2, \lambda^3, \ldots). \tag{2}$$

An observation that is crucial to our approach is that the polynomials $D(x)$ and $N(x)$ are solutions of (1) if and only if $\boldsymbol{b}$ is a solution of the difference equation

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

where σ stands for the backward shift operator, that is, $(\sigma w)(k) = w(k+1)$. The latter difference equation gives rise to a so-called behavior $\mathfrak{B}$ that is linear and shift invariant and given by

$$\mathfrak{B} = \left\{ \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} : \mathbb{Z}_+ \mapsto \mathbb{F}^2 \,|\, [-N(\sigma) \quad D(\sigma)] \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} = 0 \right\}. \tag{3}$$

We see that (λ,μ) satisfies (1) if and only if the trajectory $\boldsymbol{b}$ defined in (2) is in $\mathfrak{B}$.

The behavior $\mathfrak{B}$ specified in (3) is spanned by infinitely many trajectories $\boldsymbol{w}: \mathbb{Z}_+ \to \mathbb{F}^2$. Behavioral modeling is concerned with finding the *smallest* linear shift-invariant behavior $\mathfrak{B}$ that contains $\boldsymbol{b}$. This behavior $\mathfrak{B}^*$ is called the Most Powerful Unfalsified Model (MPUM) for the data set $\{\boldsymbol{b}\}$. It is clear that $\mathfrak{B}^*$ is necessarily contained in the behavior $\mathfrak{B}$ defined above. For $\mathfrak{B}^*$ we can immediately write down a representation, namely

$$\mathfrak{B}^* = \left\{ \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} : \mathbb{Z}_+ \mapsto \mathbb{F}^2 \,|\, \begin{bmatrix} \mu & -1 \\ \sigma - \lambda & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} = 0 \right\}$$

i.e., $\mathfrak{B}^* = \ker R(\sigma)$, where

$$R(x) = \begin{bmatrix} \mu & -1 \\ x - \lambda & 0 \end{bmatrix}.$$

In fact, in this rather trivial case $\mathfrak{B}^*$ is one-dimensional and spanned by $\boldsymbol{b}$. A two-dimensional example is provided next.

Example 1.1: Take $\mathbb{F} = \mathbb{Z}_5$. Define

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} (1,1,1,1,\ldots)$$

and

$$\tilde{\boldsymbol{b}} = \begin{bmatrix} \tilde{\boldsymbol{b}}_1 \\ \tilde{\boldsymbol{b}}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix} (1, 2, 2^2, 2^3, \ldots).$$

Then the MPUM for the data set $\{\boldsymbol{b}, \tilde{\boldsymbol{b}}\}$ is $\mathfrak{B}^*$ given by

$$\mathcal{B}^* = \left\{ \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} \Big| \begin{bmatrix} \sigma - 3 & -1 \\ (\sigma-1)(\sigma-2) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} = 0 \right\}.$$

Note that the above representation is not unique: $\mathfrak{B}^*$ can also be represented by, for example

$$\begin{bmatrix} \sigma - 3 & -1 \\ -2 & -\sigma \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} = 0.$$

C. Problem Statement

In this subsection, we formulate the general problem statement. Let $\{x_1, \ldots, x_n\}$ be a subset of a finite field $\mathbb{F}$. For the sake of clarity we make the traditional assumption that the $x_i s$ are mutually distinct. However, the approach is general enough to encompass the case where not all $x_i s$ are distinct, as put forward in recent work [26]. An (n,κ) RS code is defined as the set of codewords of the form $\boldsymbol{c} = (m(x_1), \ldots, m(x_n))$, with $m(x) \in \mathbb{F}[x]$ running through the set of polynomials of degree $< \kappa$. The codeword $\mathbf{c}$ is transmitted through a channel where errors may occur so that the received word $\boldsymbol{r}$ is not necessarily equal to the transmitted codeword $\mathbf{c}$. The decoding problem consists of reconstructing the original polynomial $m(x)$ from the received word $\mathbf{r}$. In list decoding, a list of possible polynomials $m(x)$ is generated. The breakthrough idea of [1] is to use bivariate polynomials for list decoding.

Definition 1.2: Let $Q(x,y) \in \mathbb{F}[x,y]$ be a bivariate polynomial, say

$$Q(x,y) = \sum_{i \in I, j \in J} q_{ij} x^i y^j.$$

The (w_x, w_y) weighted degree of $Q(x,y)$ is defined as

$$\mathrm{wdeg}_{(w_x,w_y)} Q(x,y) = \max_{i \in I, j \in J} \{ i w_x + j w_y \mid q_{ij} \neq 0 \}.$$

Definition 1.3: Let $Q(x,y) \in \mathbb{F}[x,y]$ be a bivariate polynomial, say

$$Q(x,y) = \sum_{i \in I, j \in J} q_{ij} x^i y^j.$$

The pair $(0,0) \in \mathbb{F}^2$ is a root of $Q(x,y)$ of multiplicity $s \in \mathbb{N}$ if $q_{i,s-1-i} = 0$ for all $i = 0, \ldots, s-1$ and $q_{i',s-i'} \neq 0$ for some $i' \in \{0, \ldots, s\}$. The pair $(\lambda,\mu) \in \mathbb{F}^2$ is a root of $Q(x,y)$ of multiplicity $s \in \mathbb{N}$ if $(0,0)$ is a root of $Q(x+\lambda, y+\mu)$ of multiplicity s.

Generally, the concept of multiple root can be expressed in terms of derivatives. In order to enable results that make sense for finite fields we first introduce the concept of Hasse derivative [27], [28] (called *hyperderivative* in [29, p. 303]).

Definition 1.4: Let $P(x) = \sum_{i=0}^{n} p_i x^i$ be a polynomial with coefficients in a field $\mathbb{F}$. Then the polynomial

$$\mathrm{D}_{\mathrm{H}}^{j} P(x) := \sum_{i=j}^{n} \binom{i}{j} p_i x^{i-j}$$

is called the jth *Hasse derivative* of $P(x)$. Let $Q(x,y) \in \mathbb{F}[x,y]$ with $Q(x,y) = q_1(x)q_2(y)$. The (ℓ_1,ℓ_2)th Hasse derivative of $Q(x,y)$, denoted by $\mathrm{D}_{\mathrm{H}}^{\ell_1,\ell_2}Q(x,y)$, is defined as

$$\mathrm{D}_{\mathrm{H}}^{\ell_1,\ell_2}Q(x,y) = \mathrm{D}_{\mathrm{H}}^{\ell_1}q_1(x)\mathrm{D}_{\mathrm{H}}^{\ell_2}q_2(y).$$

The Hasse derivative of a general polynomial $Q(x,y) = Q_1(x,y) + Q_2(x,y)$ is defined through the requirement

$$\mathrm{D}_{\mathrm{H}}^{\ell_1,\ell_2}(Q_1(x,y)+Q_2(x,y)) = \mathrm{D}_{\mathrm{H}}^{\ell_1,\ell_2}Q_1(x,y) + \mathrm{D}_{\mathrm{H}}^{\ell_1,\ell_2}Q_2(x,y).$$

Note that $j!$ times $\mathrm{D}_{\mathrm{H}}^{j}P(x)$ equals the usual jth "formal derivative" $\frac{d^jP(x)}{dx^j}$. The relation with Taylor expansions about a point $x = \lambda$ shows the role of Hasse derivative

$$\begin{aligned} P(x) =& P(\lambda) + (\mathrm{D}_{\mathrm{H}}P)(\lambda)(x-\lambda) \\ &+ (\mathrm{D}_{\mathrm{H}}^2P)(\lambda)(x-\lambda)^2 + \cdots \\ Q(x,y) =& Q(\lambda,\mu) + (\mathrm{D}_{\mathrm{H}}^{1,0}Q)(\lambda,\mu)(x-\lambda) \\ &+ (\mathrm{D}_{\mathrm{H}}^{0,1}Q)(\lambda,\mu)(y-\mu) \\ &+ (\mathrm{D}_{\mathrm{H}}^{2,0}Q)(\lambda,\mu)(x-\lambda)^2 \\ &+ (\mathrm{D}_{\mathrm{H}}^{1,1}Q)(\lambda,\mu)(x-\lambda)(y-\mu) \\ &+ (\mathrm{D}_{\mathrm{H}}^{0,2}Q)(\lambda,\mu)(y-\mu)^2 + \cdots . \end{aligned} \tag{4}$$

From this we also see that $\mathrm{D}_{\mathrm{H}}^{j}P(\lambda)$ is the coefficient of x^j in $P(x+\lambda)$. Similarly, $(\mathrm{D}_{\mathrm{H}}^{i,j}Q)(\lambda,\mu)$ is the coefficient of x^iy^j in $Q(x+\lambda, y+\mu)$. In finite fields, say of characteristic p, the Hasse derivative is much more useful than the formal derivative because whenever $j \geq p$ we have $j! = 0$ and hence all jth formal derivatives vanish. The reader may find it convenient to realize that as long as the order of "differentiation" is strictly less than the characteristic of the field, then we may as well work with the classical formal derivative. Below, we express the concept of multiple root in terms of Hasse derivatives.

Theorem 1.5:

1) The polynomial $(x-\lambda)^m$ divides $P(x)$ if and only if λ is a root of all jth Hasse derivatives of $P(x)$ for $j = 0,\ldots,m-1$.
2) Let $Q(x,y) \in \mathbb{F}[x,y]$ and $(\lambda,\mu) \in \mathbb{F}^2$. Then (λ,μ) is a root of $Q(x,y)$ of multiplicity s if and only if

$$\left(\mathrm{D}_{\mathrm{H}}^{m-\ell,\ell}Q(x,y)\right)(\lambda,\mu) = 0, \qquad m = 0,\ldots,s-1, \quad \ell = 0,\ldots,m$$

$$\left(\mathrm{D}_{\mathrm{H}}^{s-\ell,\ell}Q(x,y)\right)(\lambda,\mu) \neq 0, \qquad \text{for some } 0 \leq \ell \leq s.$$

Proof: This follows immediately from (4) or the "Repeated Factor Test" of [28], see also [29]. □

The next two results can also be found in [3].

Theorem 1.6: Let (x_i,y_i) $(i=1,\ldots,n)$ be elements of $\mathbb{F}^2$ with the x_is mutually distinct. Let $Q(x,y) \in \mathbb{F}[x,y]$ be a bivariate polynomial with $\mathrm{wdeg}_{(1,\kappa-1)}Q(x,y) = L$ such that $Q(x_i,y_i) = 0$ with multiplicity s_i, for $i = 1\ldots,n$. Let $\tilde{m}(x)$ be a polynomial of degree $< \kappa$. Let $G = \{i \mid \tilde{m}(x_i) = y_i\}$ and define $g = \sum_{i\in G} s_i$. If $g > L$ then $y - \tilde{m}(x)$ divides $Q(x,y)$.

Proof: For all $i \in G$, we have that $Q(x_i, m(x_i)) = 0$ with multiplicity at least s_i. This follows by applying the chain rule for D_{H} to $Q(x,m(x))$ and Theorem 1.5. It follows that $Q(x,m(x))$ has, counting multiplicities, at least g and therefore more than L roots. Since $\deg Q(x,m(x)) \leq L$ we conclude that $Q(x,m(x))$ is the zero polynomial. This implies that $y - m(x)$ divides $Q(x,y)$. □

Corollary 1.7: Consider the message polynomial $m(x)$ and let $\boldsymbol{r} = (y_1,\ldots,y_n)$ be the received word. Let $Q(x,y) \in \mathbb{F}[x,y]$ be a bivariate polynomial of weighted degree L such that $Q(x_i,y_i) = 0$ with multiplicity s_i, for $i = 1\ldots,n$. Let $E = \{i \mid m(x_i) \neq y_i\}$ and define $e = \sum_{i\in E} s_i$. Define $S = \sum_{i=1}^{n} s_i$. If $e < S - L$ then $y - m(x)$ divides $Q(x,y)$.

The main concern of Sudan's list decoding approach is to construct a polynomial $Q(x,y)$ such that $Q(x_i,y_i) = 0$ with prescribed multiplicities. To maximize the number of errors that can be corrected, it makes sense to minimize the weighted degree of this polynomial, an approach which is also taken in [8], [10], [11]. In the decoding process all factors of the form $y - \tilde{m}(x)$ are subsequently extracted to produce a list of candidate polynomials $\tilde{m}(x)$ of degree $< \kappa$. The next step is then to produce a sublist of most likely message words by computing the corresponding codewords and comparing with $\boldsymbol{r}$. It has been shown in the literature that the probability that this sublist contains more than one message word is "usually very small" [8]. In this paper, we solely concentrate on the polynomial construction part of this decoding process and do not consider the factorization part.

Our main aim is to place the list decoding approach of [1]–[3] in a behavioral framework. Roughly, our approach is structured as follows. We write the polynomial $Q(x,y)$ to be constructed as

$$Q(x,y) = \sum_{j=0}^{M} q_j(x)y^j$$

for an appropriate choice of M. With each data point (x_i,y_i) $(i=1,\ldots,n)$ we associate $s_i(s_i+1)/2$ trajectories. We then determine the MPUM $\mathfrak{B}$ of these trajectories. Then we construct a weighted row reduced matrix $R(x)$ that represents $\mathfrak{B}$. From $R(x)$ we select a row $q(x)$ of minimal weighted row degree and, finally, we define

$$Q(x,y) = \sum_{j=0}^{M} q_j(x)y^j$$

where the $q_i(x)$'s are the entries of $q(x)$. It turns out that $Q(x,y)$ constructed in this way is a bivariate polynomial of minimal weighted degree that interpolates the data points (x_i,y_i) with multiplicity at least s_i for $i = 1,\ldots,n$.

II. Preliminary Results for Behaviors Over Finite Fields

In this section, we review some basic concepts and results of the behavioral approach to linear systems over a field $\mathbb{F}$. For most of these the underlying field is immaterial. The only exception pertains to the differentiation of polynomials as we shall see below. Results that are obvious analogies of the real or complex case are stated without proof. The reader is referred to [30] for more detailed discussions of the behavioral theory. Results that are specific for the finite field case are stated with proof.

Following [30], a dynamical system is a triple $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$. Here $\mathbb{T}$ can be thought of as the time axis, $\mathbb{W}$ is the signal alphabet, and $\mathfrak{B}$, the behavior of the system, is a subset of $\mathbb{W}^{\mathbb{T}}$. Relevant choices for our purposes are $\mathbb{T} = \mathbb{Z}_+$, $\mathbb{W} = \mathbb{F}^q$, and $\mathfrak{B}$ a linear subspace of $\mathbb{W}^{\mathbb{T}}$.

We define σ, the backward shift operator, acting on elements in $\mathbb{W}^{\mathbb{T}}$ as $(\sigma w)(k) = w(k+1)$. Important systems are those whose behavior is defined as the kernel of a polynomial matrix in σ. Let $R(x) \in \mathbb{F}^{g\times q}[x]$ be a $g \times q$ matrix in the indeterminate x and with coefficients in $\mathbb{F}$. Then we define the behavior corresponding to $R(x)$ as

$$\mathfrak{B} = \{\mathbf{w}\colon \mathbb{Z}_+ \to \mathbb{F}^q \mid R(\sigma)\mathbf{w} = 0\}.$$

It is easy to see that $\mathfrak{B}$ is linear. Moreover, $\mathfrak{B}$ is time invariant, that is, for every trajectory $\mathbf{w}$ in $\mathfrak{B}$ the shifted trajectory $\sigma\mathbf{w}$ is also in $\mathfrak{B}$. The class of behaviors in q variables that admit a representation of the form $R(\sigma)\mathbf{w} = 0$ is denoted by $\mathfrak{L}^q$. Representations of the form $R(\sigma)\mathbf{w} = 0$ are, for obvious reasons, referred to as *kernel representations*. In the general theory of behaviors many other representations are of interest. In this paper, we only use kernel representations.

It appears that different matrices $R_1(x)$ and $R_2(x)$ may define the same behavior. It is possible to give a complete characterization of all matrices that represent a given behavior, as we show next (see [31, Theorem 3.7] for a detailed proof, see also [32]).

Lemma 2.1: For $i = 1, 2$ let $R_i(x) \in \mathbb{F}^{g_i\times q}[x]$ and denote the corresponding behaviors by $\mathfrak{B}_i$. If $\mathfrak{B}_1 \subset \mathfrak{B}_2$, then there exists a matrix $F(x) \in \mathbb{F}^{g_2\times g_1}[x]$ such that $R_2(x) = F(x)R_1(x)$.

A matrix $U(x) \in \mathbb{F}^{g\times g}[x]$ is said to be *unimodular* if there exists $V(x) \in \mathbb{F}^{g\times g}[x]$ such that $U(x)V(x) = V(x)U(x) = I$, equivalently, if $\det U(x)$ is a nonzero constant in $\mathbb{F}$. A direct consequence of the above lemma is the following theorem (see e.g., [31, Theorem 3.9] or [30, Theorem 3.6.2]).

Theorem 2.2: Let $R_i(x) \in \mathbb{F}^{g\times q}[x]$ define the same behavior ($i = 1, 2$), i.e., $R_1(\sigma)\mathbf{w} = 0$ if and only if $R_2(\sigma)\mathbf{w} = 0$. Then there exists a unimodular matrix $U(x) \in \mathbb{F}^{g\times g}[x]$ such that $R_2(x) = U(x)R_1(x)$.

Theorem 2.2 makes it possible to choose out of the many representations of a given behavior one that is particularly convenient for the application at hand. Examples are upper or lower triangular forms. Also, by means of appropriate unimodular premultiplication one may create zero rows to end up with a matrix in which the remaining nonzero rows are independent over $\mathbb{F}[x]$. The nonzero rows then form a matrix with fewer rows that is said to be of full row rank. A form that is crucial in the application of the behavioral approach to coding theory is the row reduced form.

Definition 2.3: Let $R(x) \in \mathbb{F}^{g\times q}[x]$ and denote the rows of $R(x)$ by $r_i(x)$, $i = 1, \ldots, g$. The *row degrees* $d_1, \ldots, d_g$ are defined as $d_i = \max_{j=1,\ldots,q} \deg r_{ij}(x)$. Define the diagonal matrix $D(x) = \operatorname{diag}(x^{d_1}, \ldots, x^{d_g})$ and write $R(x) = D(x)R_0 + R_1(x)$ with $D(x)^{-1}R_1(x)$ strictly proper, meaning that in every entry of $D(x)^{-1}R_1(x)$ the degree of the denominator exceeds the degree of the numerator. Then, $R(x)$ is said to be *row reduced* if R_0 is of full row rank as a matrix in $\mathbb{F}^{g\times q}$. The matrix R_0 is called the leading row coefficient matrix.

The next two theorems are well-known results from behavioral theory.

Theorem 2.4: Let $R(x) \in \mathbb{F}^{q\times q}[x]$ be a square matrix with row degrees $d_1, \ldots, d_q$. Denote the sum of these row degrees by d. Then $R(x)$ is row reduced if and only if $\deg \det R(x) = d$.

Theorem 2.5: Let $R(x) \in \mathbb{F}^{g\times q}[x]$ be of full row rank. There exists a unimodular matrix $U(x)$ such that $U(x)R(x)$ is row reduced.

For $g = 2$, Theorem 2.5 may conveniently be proved using the Euclidean algorithm. The proof for the general case $g > 2$ uses a slightly different technique, dating back to [33, p. 27], see also [31, p. 24]. We provide this proof for the sake of completeness and the convenience of the reader. Moreover, the procedure that is outlined in the proof is used explicitly in Section IV, see also Example 2.9.

Proof: Refer to Definition 2.3 for the notation. Suppose that R_0 is not of full row rank. Then there exists a nonzero vector $v \in \mathbb{R}^g$ such that $v^T R_0 = 0$. Let $r^{(j^*)}(x)$ be a row of $R(x)$ for which the row degree is maximal among all rows of $R(x)$ for which the corresponding component of v is nonzero. Define the unimodular matrix $U(x)$ as

$$U(x) = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ & \ddots & & & \\ v_1 x^{d_{j^*}-d_1} & \cdots & v_{j^*} & \cdots & v_g x^{d_{j^*}-d_g} \\ & & & \ddots & \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}.$$

Premultiplication of $R(x)$ with $U(x)$ leaves all rows unaltered except the j^*th row which is transformed into a linear combination of the rows of $R_1(x)$

$$v_1 x^{d_{j^*}-d_1} r_1^{(1)}(x) + \cdots + v_{j^*} r_1^{(j^*)}(x) + \cdots + v_g x^{d_{j^*}-d_g} r_1^{(g)}(x). \quad (5)$$

Because of $v^T R_0 = 0$, the row degree of (5) is strictly smaller than d_{j^*}. As a consequence, the sum of the row degrees of $U(x)R(x)$ is strictly smaller than the sum of the row degrees of $R(x)$. We can repeat this transformation for as long as the leading row coefficient matrix does not have full row rank. On the other hand, the sum of the row degrees is a nonnegative integer and can therefore only decrease a finite number of times. The conclusion is that after a finite number of steps a row reduced form is reached. □

In the sequel, we use a modified version of row reducedness of which the above is a special case. This is the notion of "weighted row reduced."

Definition 2.6: Let $n_1, \ldots, n_q$ be nonnegative integers. Define

$$N(x) = \operatorname{diag}(x^{n_1}, \ldots, x^{n_q}). \quad (6)$$

The weighted row degrees of a matrix $R(x) \in \mathbb{F}^{g\times q}[x]$ are defined as the row degrees of $R(x)N(x)$. The matrix $R(x)$ is called $(n_1,\ldots,n_q)$ weighted row reduced if $R(x)N(x)$ is row reduced.

Notice that $(0,\ldots,0)$ weighted row reduced is just row reduced. We mainly consider $(0,\kappa{-}1,2(\kappa{-}1),\ldots,(q{-}1)(\kappa{-}1))$ weighted row reduced. We shall refer to this special case as simply weighted row reduced.

The following two theorems are generalizations of Theorems 2.4 and 2.5. They come in useful in Section IV.

Theorem 2.7: Let $R(x) \in \mathbb{F}^{q\times q}[x]$ be a square polynomial matrix of full row rank and let $n_1,\ldots,n_q$ be nonnegative integers. Let $N(x)$ be defined as in (6). Then $R(x)$ is $(n_1,\ldots,n_q)$ weighted row reduced if and only if $\deg\det R(x)+\deg\det N(x)$ equals the sum of the weighted row degrees of $R(x)$.

Proof: It follows from Theorem 2.4 that $R(x)N(x)$ is row reduced if and only if

$$\deg\det R(x)N(x) = \deg\det R(x) + \deg\det N(x)$$

equals the sum of the row degrees of $R(x)N(x)$. By definition, the latter equals the sum of the weighted row degrees of $R(x)$, which proves the theorem. □

Theorem 2.8: Let $R(x)\in\mathbb{F}^{g\times q}[x]$ be of full row rank and let $n_1,\ldots,n_q$ be nonnegative integers. There exists a unimodular matrix $U(x)$ such that $U(x)R(x)$ is $(n_1,\ldots,n_q)$ weighted row reduced.

Proof: Let $N(x)$ be as in (6). According to Theorem 2.5, there exists a unimodular matrix $U(x)$ such that $U(x)R(x)N(x)$ is row reduced. But then, by definition, $U(x)R(x)$ is $(n_1,\ldots,n_q)$ weighted row reduced. □

Example 2.9: Let $\kappa = 2$ and let the matrix $R(x)\in\mathbb{Z}_5^{3\times 3}[x]$ be given by

$$R(x) = \begin{bmatrix} x^5+4x & 0 & 0 \\ 4x^3+4x^2+4x+1 & 1 & 0 \\ 2x^4+4 & 0 & 1 \end{bmatrix}.$$

This matrix is not weighted row reduced. We demonstrate how this matrix is transformed into weighted row reduced form. To that end, we first postmultiply $R(x)$ by $N(x)=\operatorname{diag}(1,x,x^2)$ yielding

$$\begin{bmatrix} x^5{+}4x & 0 & 0 \\ 4x^3{+}4x^2{+}4x{+}1 & x & 0 \\ 2x^4{+}4 & 0 & x^2 \end{bmatrix} = \begin{bmatrix} x^5 & 0 & 0 \\ 0 & x^3 & 0 \\ 0 & 0 & x^4 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 4 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 4x & 0 & 0 \\ 4x^2{+}4x{+}1 & 1 & 0 \\ 4 & 0 & x^2 \end{bmatrix}.$$

Take, for example

$$v = [\,2 \quad 0 \quad -1\,] \qquad \text{so that } U_1(x) = \begin{bmatrix} 2 & 0 & -x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then

$$U_1(x)R(x)N(x) = \begin{bmatrix} 4x & 0 & -x^3 \\ 4x^3+4x^2+4x+1 & x & 0 \\ 2x^4+4 & 0 & x^2 \end{bmatrix}.$$

Repeating the argument we finally obtain

$$U(x)R(x)N(x) = \begin{bmatrix} 4x & 0 & 4x^3 \\ 4x^3+4x^2+4x+1 & x & 0 \\ x+3 & 3x^2+2x & 4x^2 \end{bmatrix}$$

so that

$$U(x)R(x) = \begin{bmatrix} 4x & 0 & 4x \\ 4x^3+4x^2+4x+1 & 1 & 0 \\ x+3 & 3x+2 & 4 \end{bmatrix}$$

which is indeed weighted row reduced.

The next result shows how row reducedness gives rise to the so-called "predictable-degree property" (terminology from Forney's paper [34]). This observation turns out to be crucial in the behavioral interpretation of the decoding scheme of [1].

Theorem 2.10: Let $R(x) \in \mathbb{F}^{g\times q}[x]$ be row reduced and denote the row degrees by $d_1,\ldots,d_g$. Let $a(x)\in\mathbb{F}^{1\times g}[x]$ be a nonzero vector. Then the row degree of $a(x)R(x)$ equals

$$\max_i(d_i+\deg a_i(x)). \tag{7}$$

Proof: See [34] and also [35, Theorem 6.3–13]. □

Corollary 2.11: Let $R(x) \in \mathbb{F}^{g\times q}[x]$ be weighted row reduced and denote the weighted row degrees by $d_1,\ldots,d_g$. Let $a(x) \in \mathbb{F}^{1\times g}[x]$ be a nonzero vector. Then the weighted row degree of $a(x)R(x)$ equals

$$\max_i(d_i+\deg a_i(x)).$$

Proof: Define $r(x)=a(x)R(x)$. Let the diagonal matrix

$$W(x) = \operatorname{diag}(1, x^{\kappa-1},\ldots,x^{(q-1)(\kappa-1)}).$$

Since $R(x)$ is weighted row reduced, $R(x)W(x)$ is row reduced with row degrees $d_1,\ldots,d_g$. By Theorem 2.10, the row degree of $r(x)W(x)$ is given by (7). Since the weighted row degree of $r(x)$ is the row degree of $r(x)W(x)$ the statement follows. □

As remarked, behaviors are represented by polynomial matrices. The question arises how, for a given polynomial matrix, the behavior can be determined explicitly. Our key players are trajectories $\boldsymbol{b}^j\colon\mathbb{Z}_+\to\mathbb{F}$ of the form

$$b^j(k) := \begin{cases} \binom{k}{j}\lambda^{k-j}, & \text{for } k\geq j \\ 0, & \text{for } k<j \end{cases} \tag{8}$$

where $\lambda\in\mathbb{F}$. So far, Hasse derivatives have only been defined for polynomials. We would however like to interpret the trajectory $\boldsymbol{b}^j$ as the jth Hasse derivative of the trajectory $\boldsymbol{b}^0$. To this end, we now extend the definition of Hasse derivative to trajectories as follows.

Definition 2.12: Let $\lambda\in\mathbb{F}$ and $\boldsymbol{w}\in\mathbb{F}^{\mathbb{Z}_+}$ be such that $w(k)=P_k(\lambda)$ for all $k\in\mathbb{Z}_+$ and some $P_k\in\mathbb{F}[x]$. Then the jth Hasse derivative of the trajectory $\boldsymbol{w}$ is defined as the trajectory

$$\mathrm{D}_{\mathrm{H}}^j\boldsymbol{w}\colon k\mapsto \mathrm{D}_{\mathrm{H}}^jP_k(\lambda).$$

With this definition, the trajectory $\boldsymbol{b}^j$ defined in (8) can be written as $\boldsymbol{b}^j = \mathrm{D}_{\mathrm{H}}^j \boldsymbol{b}^0$. In the sequel, for $\boldsymbol{w} \in \mathbb{F}^{\mathbb{Z}_+}$ and constant $Y \in \mathbb{F}^q$ we denote the trajectory $k \mapsto Yw(k)$ by $Y\boldsymbol{w}$. Then obviously

$$\mathrm{D}_{\mathrm{H}}^j(Y\boldsymbol{w}) = Y\mathrm{D}_{\mathrm{H}}^j\boldsymbol{w}. \tag{9}$$

Now, let $R(x)$ be a given polynomial matrix in $\mathbb{F}^{g\times q}[x]$. Then obviously

$$R(\sigma)(Y\mathrm{D}_{\mathrm{H}}^j\boldsymbol{w}) = \mathrm{D}_{\mathrm{H}}^j(R(\sigma)(Y\boldsymbol{w})). \tag{10}$$

Further, it is easy to check that for $\boldsymbol{b}^0$ defined by (8) we have

$$R(\sigma)Y\boldsymbol{b}^0 = R(\lambda)Y\boldsymbol{b}^0. \tag{11}$$

It now follows straightforwardly from (9)–(11) and the product rule for Hasse derivative that

$$\begin{aligned} R(\sigma)(Y\boldsymbol{b}^\ell) &= \mathrm{D}_{\mathrm{H}}^\ell(R(\sigma)(Y\boldsymbol{b}^0)) = \mathrm{D}_{\mathrm{H}}^\ell(R(\lambda)Y\boldsymbol{b}^0) \\ &= \sum_{j=0}^{\ell} \mathrm{D}_{\mathrm{H}}^j R(\lambda)Y\boldsymbol{b}^{\ell-j}. \end{aligned} \tag{12}$$

Using (12), we arrive at the following result which is the finite field equivalent of [30, Theorem 3.2.16].

Theorem 2.13: Let $R(x) \in \mathbb{F}^{q\times q}[x]$, let $\det R(x)$ be a polynomial of degree n, and let $\mathfrak{B} = \{\mathbf{w}\colon \mathbb{Z}_+ \to \mathbb{F}^q \mid R(\sigma)\mathbf{w} = 0\}$. Then $\mathfrak{B}$ is an n-dimensional subspace of $(\mathbb{F}^q)^{\mathbb{Z}_+}$. If

$$\det R(x) = c\prod_{i=1}^{N}(x-\lambda_i)^{m_i}$$

with $c \neq 0$ and $\lambda_i \in \mathbb{F}$, then all trajectories in $\mathfrak{B}$ are of the form

$$\boldsymbol{w} = \sum_{i=1}^{N}\sum_{j=0}^{m_i-1} b_{ij}\mathrm{D}_{\mathrm{H}}^j\boldsymbol{b}_i$$

with $\boldsymbol{b}_i$ defined by $b_i(k) = \lambda_i^k$ for $k \in \mathbb{Z}_+$ and with $b_{ij} \in \mathbb{F}^q$ satisfying the linear restrictions

$$\sum_{j=\ell}^{m_i-1}\left[\mathrm{D}_{\mathrm{H}}^{j-\ell}R(\lambda_i)\right]b_{ij} = 0, \quad \ell = 0,\ldots,m_i-1,\ i=1,\ldots,N.$$

Example 2.14: Let $R(x) \in \mathbb{Z}_3^{2\times 2}[x]$ be given by

$$R(x) = \begin{bmatrix} x^4+x^3+x+1 & x^5+x^4+x^2+2x+1 \\ x^5+x^4+x^2+x & x^6+x^5+x^3+2x+1 \end{bmatrix}.$$

Then

$$\det R(x) = x^6+2x^5+2x^4+2x^3+2x^2+2x+1 = (x-1)^2(x-2)^4.$$

Using Theorem 2.13 it follows that all solutions of $R(\sigma)\boldsymbol{w} = 0$ are of the form

$$\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 0 \\ 2x_{11} \end{bmatrix} k + \begin{bmatrix} x_{21} \\ 0 \end{bmatrix} 2^k + \begin{bmatrix} x_{22} \\ 0 \end{bmatrix} k2^{k-1} + \begin{bmatrix} x_{23} \\ 0 \end{bmatrix} \binom{k}{2} 2^{k-2} + \begin{bmatrix} x_{24} \\ 0 \end{bmatrix} \binom{k}{3} 2^{k-3}.$$

In the sequel we only use a special case of Theorem 2.13 which we therefore state next.

Corollary 2.15: Let $R(x) \in \mathbb{F}^{q\times q}[x]$, with $\det R(x) = x-\lambda$. $\lambda \in \mathbb{F}$. Let $\boldsymbol{w}\colon \mathbb{Z}_+ \to \mathbb{F}^q$. Then $R(\sigma)\boldsymbol{w} = 0$ if and only if $w(k) = Y\lambda^k$ with $R(\lambda)Y = 0$.

In the above we investigated explicit expressions for trajectories satisfying a given polynomial representation. In the sequel, we are interested in the converse, namely, building representations from given trajectories. For this purpose, we use the theory of exact modeling of behaviors as first introduced in [36]. We recall a few of the main ideas. Given a finite number of trajectories $\mathbf{w}_j\colon \mathbb{Z}_+ \to \mathbb{F}^q$ $(j = 1,\ldots,N)$ we may seek to build a system whose behavior contains these specific trajectories. A behavior $\mathfrak{B}$ is called an *unfalsified model* for the *data set* $\mathbf{D} = \{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_N\}$ if $\mathbf{D} \subseteq \mathfrak{B}$. A model $\mathfrak{B}_1$ is called *more powerful* than a model $\mathfrak{B}_2$ if $\mathfrak{B}_1 \subseteq \mathfrak{B}_2$. From a modeling perspective it appears sensible to look for the smallest behavior that contains the N trajectories. A model $\mathfrak{B}^*$ is called the MPUM for $\mathbf{D}$, if $\mathfrak{B}^*$ is unfalsified for $\mathbf{D}$ and $\mathbf{D} \subseteq \mathfrak{B} \Longrightarrow \mathfrak{B}^* \subseteq \mathfrak{B}$. In [21], it is shown that for $\mathbf{D} = \{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_N\}$ such an MPUM exists. In fact, it is characterized as $\mathfrak{B}^* = \bigcap_{\mathfrak{B}\in\mathfrak{M}} \mathfrak{B}$, where $\mathfrak{M}$ is the set of unfalsified models for $\mathbf{D}$. In [21], a general procedure for the iterative construction of a kernel representation for $\mathfrak{B}^*$ is presented. We recall this procedure; its workings can be easily understood from Lemma 2.1.

Procedure 2.16: ([21]) Initially define

$$R_0(x) := I_q \text{ (where } I_q \text{ is the } q\times q \text{ identity matrix).}$$

Proceed iteratively as follows for $k = 1,\ldots,N$. Define, after receiving $\{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_k\}$, the kth error trajectory $\boldsymbol{e}_k$ as

$$\boldsymbol{e}_k := R_{k-1}(\sigma)\boldsymbol{w}_k.$$

Compute a kernel representation $V_k(\sigma)\boldsymbol{w} = 0$ of the MPUM for $\{\boldsymbol{e}_k\}$. Then define

$$R_k(x) := V_k(x)R_{k-1}(x).$$

Theorem 2.17: ([21]) For $k = 1,\ldots,N$, the kernel representation $R_k(\sigma)\boldsymbol{w} = 0$ of the above procedure, represents the MPUM for $\{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_k\}$.

Remark 2.18: For general trajectories, Procedure 2.16 may be cumbersome to run. However, for exponential trajectories, the procedure is easy and convenient to perform. As an illustration, for given $\lambda \in \mathbb{F}$ and $Y \in \mathbb{F}^q$, consider the trajectories $Y\boldsymbol{b}^0$, $Y\boldsymbol{b}^1$, and $Y\boldsymbol{b}^2$ with $\boldsymbol{b}^j$ defined by (8). We want to determine the MPUM of these three trajectories. Set $R_0(x) = I$ and define $\boldsymbol{e}_1 = R_0(\sigma)Y\boldsymbol{b}^0$. So $\boldsymbol{e}_1 = Y\boldsymbol{b}^0$. Choose $V_1(x)$ such that $\det V_1(x) = x-\lambda$ and $V_1(\lambda)Y = 0$. Then, according to Corollary 2.15, $V_1(\sigma)\boldsymbol{w} = 0$ represents the MPUM of $\boldsymbol{e}_1$. Set $R_1(x) = V_1(x)R_0(x)$. Next, define $\boldsymbol{e}_2 = R_1(\sigma)Y\boldsymbol{b}^1$. It follows from (12) that

$$e_2(k) = \underbrace{\mathrm{D}_{\mathrm{H}}R_1(\lambda)Y}_{Y_1}\lambda^k + \underbrace{R_1(\lambda)Y}_{=0}k\lambda^{k-1}.$$

Take $V_2(x)$ such that $\det V_2(x) = x - \lambda$ and $V_2(\lambda)Y_1 = 0$ and define $R_2(x) = V_2(x)R_1(x)$. Define $\boldsymbol{e}_3 = R_2(\sigma)Y\boldsymbol{b}^2$. Then, again from (12), $e_3(k)$ equals

$$\underbrace{\mathrm{D}_{\mathrm{H}}^2 R_2(\lambda)Y}_{Y_2}\lambda^k + \underbrace{\mathrm{D}_{\mathrm{H}} R_2(\lambda)Y}_{=0} k\lambda^{k-1} + \underbrace{R_2(\lambda)Y}_{=0}\frac{k(k-1)}{2}\lambda^{k-2}.$$

The middle term is zero because

$$\begin{aligned}\mathrm{D}_{\mathrm{H}} R_2(\lambda)Y &= \mathrm{D}_{\mathrm{H}} V_2(\lambda)R_1(\lambda)Y + V_2(\lambda)\mathrm{D}_{\mathrm{H}} R_1(\lambda)Y \\ &= V_2(\lambda)Y_1 = 0.\end{aligned}$$

Hence, choose $V_3(x)$ such that $\det V_3(x) = x - \lambda$ and $V_3(\lambda)Y_2 = 0$. Finally, define $R_3(x) = V_3(x)R_2(x)$. Then $R_3(\sigma)\boldsymbol{w} = 0$ represents the MPUM of $Y\boldsymbol{b}^0$, $Y\boldsymbol{b}^1$, and $Y\boldsymbol{b}^2$.

III. Minimal Interpolation as Behavioral Modeling

In this section, we reformulate the problem statement as introduced in Section I in terms of behavioral modeling. In Section IV, we then proceed to solve the problem by employing iterative behavioral modeling. As outlined in Section I, the problem statement in its most general form involves interpolation "with multiplicity."

A. Problem Statement

Given n triples $(x_i, y_i, s_i) \in \mathbb{F}^2 \times \mathbb{Z}_+$ $(i = 1, \ldots, n)$, find a polynomial $Q(x, y) \in \mathbb{F}[x, y]$ of minimal weighted degree such that $Q(x_i, y_i) = 0$ for $i = 1, \ldots, n$ with multiplicity at least s_i.

We formulate this problem statement in terms of behavioral modeling. Analogous to (8), we define trajectories $\boldsymbol{w}_i^j : \mathbb{Z}_+ \to \mathbb{F}$ by

$$w_i^j(k) := \begin{cases} \binom{k}{j} x_i^{k-j}, & \text{for } k \geq j \\ 0, & \text{for } k < j \end{cases}. \tag{13}$$

Also, for $\ell = 0, \ldots, M$ we define

$$Y_i^\ell := \mathrm{D}_{\mathrm{H}}^\ell \begin{bmatrix} 1 & y & \cdots & y^M \end{bmatrix}^T_{y=y_i}. \tag{14}$$

Theorem 3.1: Let $Q(x, y) \in \mathbb{F}[x, y]$ be a bivariate polynomial, written as

$$Q(x, y) = \sum_{j=0}^{M} q_j(x) y^j.$$

Then, for $i = 1, \ldots, n$, $Q(x_i, y_i) = 0$ with multiplicity at least s if and only if

$$\underbrace{\begin{bmatrix} q_0(\sigma) & \cdots & q_M(\sigma) \end{bmatrix}}_{q(\sigma)} \left(Y_i^\ell \boldsymbol{w}_i^{m-\ell}\right) = 0,$$
$$m = 0, \ldots, s-1, \ \ell = 0, \ldots, m. \tag{15}$$

Proof: We give the proof for $s = 2$ only. For $s > 2$ the proof follows the same lines.

First, assume that $Q(x_i, y_i) = 0$ with multiplicity at least two. We need to prove (15) for $(m, \ell) = (0, 0), (1, 0)$, and $(1, 1)$, respectively. According to the second part of Theorem 1.5, we conclude that

$$\mathrm{D}_{\mathrm{H}}^{0,0} Q(x_i, y_i) = \mathrm{D}_{\mathrm{H}}^{1,0} Q(x_i, y_i) = \mathrm{D}_{\mathrm{H}}^{0,1} Q(x_i, y_i) = 0.$$

From (11), we conclude

$$q(\sigma)Y_i^0 \boldsymbol{w}_i^0 = 0 \tag{16a}$$
$$\mathrm{D}_{\mathrm{H}} q(\sigma)Y_i^0 \boldsymbol{w}_i^0 = 0 \tag{16b}$$
$$q(\sigma)Y_i^1 \boldsymbol{w}_i^0 = 0. \tag{16c}$$

Then from (16a) and (16c), we obtain two of the three claims made in (15). Next, from (12) it follows that

$$q(\sigma)Y_i^0 \boldsymbol{w}_i^1 = \mathrm{D}_{\mathrm{H}} q(x_i)Y_i^0 \boldsymbol{w}_i^0 + q(x_i)Y_i^0 \boldsymbol{w}_i^1. \tag{17}$$

From (16a), (11) it follows that $q(x_i)Y_i^0 \boldsymbol{w}_i^0 = 0$ and, therefore, $q(x_i)Y_i^0 = 0$. Once more using (11) we then get from (17) and (16b) that

$$q(\sigma)Y_i^0 \boldsymbol{w}_i^1 = 0. \tag{18}$$

This proves (15) for $(m, \ell) = (1, 0)$.

Second, assume that (16a), (16c), and (18) are true. From (16a), (16c), (11) we get

$$\mathrm{D}_{\mathrm{H}}^{0,0} Q(x_i, y_i) = 0 \qquad \mathrm{D}_{\mathrm{H}}^{0,1} Q(x_i, y_i) = 0. \tag{19}$$

Finally, again using (11), it follows from (17) that also (16b) is true. From the second part of Theorem 1.5 it then follows that $Q(x_i, y_i) = 0$ with multiplicity at least two. □

The problem is now to find an integer M and a vector $q(x) = \begin{bmatrix} q_0(x) & \cdots & q_M(x) \end{bmatrix}$ satisfying the above. Notice that (15) only guarantees interpolation with multiplicity *at least* s_i. At first sight one might, relying on univariate intuition, expect that the additional requirement that $Q(x, y)$ is of minimal weighted degree implies that the multiplicity is *exactly* s_i. This, however, is not true as the following simple example shows.

Example 3.2: Let $\mathbb{F} = \mathbb{Z}_5$ and take as interpolation points $\{(0, 0), (1, 0), (2, 0), (3, 3), (4, 4)\}$. Furthermore, take $\kappa = 2$. The polynomial $Q(x, y) \in \mathbb{F}[x, y]$ of minimal weighted degree that interpolates these points can be proven to be unique up to a scaling factor and equals $Q(x, y) = 4xy + y^2$. Inspection yields that all points are interpolated with multiplicity one, except $(0, 0)$ which has multiplicity two.

For the choice of M we argue as follows. A trivial solution to the interpolation problem is given by $\prod_{i=1}^n (x - x_i)^{s_i}$. With $S := \sum_{i=1}^n s_i$, this polynomial has weighted degree S and, therefore, the following choice of M suffices:

$$M = \max\left\{ j \in \mathbb{N} \mid j \leq \frac{S}{\kappa - 1} \right\}. \tag{20}$$

Remark 3.3: A tighter upper bound for the minimal weighted degree can be expressed in terms of both n and κ. It is based on a counting argument, see [1, Lemma 7]. This upper bound can then be used to derive a possibly smaller choice of M.

We now proceed as follows. We construct a weighted row reduced matrix $R(x)$ that represents the MPUM of the trajectories in (15). From $R(x)$ we select a row $q(x) = \begin{bmatrix} q_0(x) & \ldots & q_M(x) \end{bmatrix}$ of minimal weighted row degree. The desired polynomial

$$Q(x, y) = \sum_{j=0}^{M} q_j(x) y^j$$

interpolates each data point (x_i, y_i) with multiplicity at least s_i and has minimal weighted degree.

Theorem 3.4: Let $\mathfrak{B}$ be the MPUM of

$$\{Y_i^\ell \mathbf{w}_i^{m-\ell} \mid i = 1, \ldots, n; m = 0, \ldots, s_i - 1; \ell = 0, \ldots, m\}$$

defined in (13), (14) with M defined by (20). Let $R(x) \in \mathbb{F}^{(M+1)\times(M+1)}[x]$ be a weighted row reduced representation of $\mathfrak{B}$ and let $q(x) = [\, q_0(x) \quad \cdots \quad q_M(x) \,]$ be a row of $R(x)$ of minimal weighted degree. Define

$$Q(x, y) = \sum_{j=0}^{M} q_j(x) y^j.$$

Then $Q(x, y)$ is a polynomial of minimal weighted degree with $Q(x_i, y_i) = 0$ for $i = 1, \ldots, n$ with multiplicity at least s_i.

Proof: Let $\tilde{Q}(x, y) \in \mathbb{F}[x, y]$ be such that $\tilde{Q}(x_i, y_i) = 0$ with multiplicity at least s_i for $i = 1, \ldots, n$. Write

$$\tilde{Q}(x, y) = \sum_{j=0}^{M} \tilde{q}_j(x) y^j$$

and $\tilde{q}(x) = [\, \tilde{q}_0(x) \quad \cdots \quad \tilde{q}_M(x) \,]$. Then, by Theorem 3.1 , for $i = 1, \ldots, n$

$$[\tilde{q}_0(\sigma) \cdots \tilde{q}_M](\sigma) Y_i^\ell \mathbf{w}_i^{m-\ell} = 0, \quad m = 0, \ldots, s_i \;\; \ell = 0, \ldots, m.$$

It follows from the definition of MPUM that $\tilde{q}(\sigma)\mathbf{w} = 0$ for all $\mathbf{w} \in \mathfrak{B}$. It follows from Lemma 2.1 that there exists $F(x) \in \mathbb{F}^{1\times(M+1)}[x]$ such that $\tilde{q}(x) = F(x)R(x)$. It now follows from Corollary 2.11 that the weighted row degree of $\tilde{q}(x)$ is larger than or equal to the weighted row degree of $q(x)$. This means that the weighted degree of $\tilde{Q}(x, y)$ is larger than or equal to the weighted degree of $Q(x, y)$. □

IV. Decoding Procedure

In this section we show how behavioral modeling, in particular the iterative modeling Procedure 2.16, can be put to work for list decoding. The workings of this procedure in its general context of behavioral modeling were described in Section II. Here we use this theory to first produce a general interpolation procedure in which the matrix $V_i(x)$ in Procedure 2.16, which is the update matrix at step i, is left unspecified. For different choices of update matrix the procedure then turns into different interpolation algorithms.

Procedure 4.1: **General interpolation procedure**
INPUT: interpolation data (x_i, y_i) for $i = 1, \ldots, n$; multiplicities s_i for $i = 1, \ldots, n$; parameter M.

Step 1: **Initialization** $R_0(x) := I_{M+1}$, i.e., the $(M + 1) \times (M + 1)$ identity matrix.
Now proceed iteratively for $i = 1, \ldots, n$ with

Step 2: Process (x_i, y_i): taking the matrix $R_{i-1}(x)$ as input this step outputs the matrix $R_i(x)$.

OUTPUT: matrix $R_n(x)$.

The rows of the final matrix $R_n(x)$ produce bivariate polynomials that are solutions to the interpolation problem as formulated in Section III. In particular, a row $[q_0(x) \cdots q_M(x)]$ of lowest weighted row degree L gives rise to a solution

$$Q(x, y) := q_0(x) + q_1(x)y + \cdots + q_M(x)y^M$$

of minimal weighted degree L. The reason for this is that the algorithm is set up in such a way that $R_n(x)$ is weighted row reduced.

The remainder of this section is organized as follows. In Section IV-A, we present the case $s_i = 1$, whereas Section IV-B assumes general values for the multiplicities $s_i (i = 1, \ldots, n)$. In each of these subsections, we present two choices (Choice I and Choice II) for the type of update matrix $V_i(x)$. Choice I is a rather straightforward choice, leading to an algorithm that produces a final matrix $R_n(x)$ that is not necessarily weighted row reduced. In Section IV-A (all multiplicities set to one), this algorithm builds up the Lagrange interpolating solutions. Minimal interpolation can then be achieved by next applying the procedure indicated in the proof of Theorem 2.5 for making a matrix weighted row reduced (use also Theorem 2.8). In Choice II, the update matrix $V_i(x)$ is chosen in such a way that it creates weighted row reducedness at each step, thus immediately producing minimal solutions. Such an update needs to keep track of weighted row degrees, just as in the minimal interpolation algorithm of [8]. In Section VI, we comment on the relationship of our algorithm with the algorithm of [8]. In Section V, we elaborate on some of the implications of using the behavioral approach. In particular, we find that, in generating a minimal solution, our behavioral algorithm in fact generates a polynomial basis for all interpolating solutions in an explicit fashion.

Throughout the remainder of this section we use a $(5, 2)$ RS code on $\mathbb{Z}_5$ as a running example.

A. Multiplicity One

For $s_i = 1$ $(i = 1, \ldots, n)$ Step 2 of Procedure 4.1 can be specified in more detail as follows (Y_i^0 defined as in (14)):

Step 2.1: Compute

$$\Delta_i := R_{i-1}(x_i) Y_i^0.$$

Step 2.2: Specify the update matrix $V_i(x)$.

Step 2.3: Compute $R_i(x) := V_i(x) R_{i-1}(x)$.

Here the update matrix $V_i(x)$ should be chosen such that

$$V_i(\sigma)\boldsymbol{w} = 0 \tag{21}$$

represents the MPUM of $\{\Delta_i \boldsymbol{w}_i^0\}$. This is due to the fact that, at each step, the error trajectory $R_{i-1}(\sigma) Y_i^0 \boldsymbol{w}_i^0$ equals $\Delta_i \boldsymbol{w}_i^0$.

The above procedure exactly follows the steps of Procedure 2.16. It therefore follows immediately that for $i = 1, \ldots, n$ the matrices $R_i(x)$ in the above algorithm are such that $R_i(\sigma)\boldsymbol{w} = 0$ represents the MPUM of $\{Y_1^0 \boldsymbol{w}_1^0, \ldots, Y_i^0 \boldsymbol{w}_i^0\}$. The only issue that still deserves attention is the choice of update matrix $V_i(x)$. It follows from Corollary 2.15 that for (21) to represent the MPUM of $\{\Delta_i \boldsymbol{w}_i^0\}$ we need to make sure that $V_i(x_i)\Delta_i = 0$ and $\det V_i(x) = x - x_i$. This still leaves room for numerous choices. In the following we focus on two of those choices. In the sequel, e_j denotes the jth unit vector.

1) Nonminimal Choice of $V_i(x)$: A particularly straightforward choice for the update matrix $V_i(x)$ in the above algorithm is

$$V_i(x) := (x - x_i)e_1 e_1^T + \sum_{j \neq 1} e_j \left(e_j^T - \frac{\Delta_i(j)}{\Delta_i(1)} e_1^T \right)$$

that is,

$$V_i(x) = \begin{bmatrix} x - x_i & & & \\ -\Delta_i(2)/\Delta_i(1) & 1 & & \\ -\Delta_i(3)/\Delta_i(1) & & \ddots & \\ \vdots & & & \\ -\Delta_i(M+1)/\Delta_i(1) & & & 1 \end{bmatrix}. \quad (22)$$

It is not difficult to show that, as long as the x_is are distinct, this choice guarantees that $\Delta_i(1) \neq 0$ for all $i = 1, \ldots, n$ so that (22) is well defined.

For this choice of update matrix the final matrix $R_n(x)$ is given by

$$R_n(x) = \begin{bmatrix} \Pi(x) & & & \\ -p_1(x) & 1 & & \\ -p_2(x) & & \ddots & \\ \vdots & & & \\ -p_M(x) & & & 1 \end{bmatrix} \quad (23)$$

where $\Pi(x) = \prod_{i=1}^{n}(x - x_i)$ and $p_j(x)$ denotes the Lagrange polynomial of degree $< n$ that maps x_i to y_i^j for $i = 1, \ldots, n$. This is most easily seen by expressing the Lagrange polynomial in its Newton polynomial form. Thus, we have an iterative procedure for building this model which lends itself to an elegant implementation.

Example 4.2: Consider the $(5, 2)$ RS code C on $\mathbb{Z}_5$ defined by

$$C = \{(m(0), \ldots, m(4)) | m(x) = a + bx,\ a, b \in \mathbb{Z}_5\}.$$

Let the received word be as in [11, Example 1], i.e., $\boldsymbol{r} = (4, 2, 3, 3, 3)$. Running the preceding algorithm for $M = 2$ yields a final matrix $R_5(x)$ given by

$$R_5(x) = \begin{bmatrix} x^5 + 4x & 0 & 0 \\ 4x^3 + 4x^2 + 4x + 1 & 1 & 0 \\ 2x^4 + 4 & 0 & 1 \end{bmatrix}$$

which is easily seen to be of the form (23). This matrix is not weighted row reduced as the leading weighted row coefficient matrix is singular.

As demonstrated by Example 4.2, for the above choice of $V_i(x)$, the resulting matrix is not necessarily weighted row reduced. It follows from Theorem 3.4 that in order to get a solution of our interpolation problem we need to bring the matrix into weighted row reduced form. For this we resort to the method outlined in Section II.

Example 4.2–Continued Applying Theorem 2.8 and the procedure from the proof of Theorem 2.5 to the above matrix $R_5(x)$ yields weighted row reducedness. The resulting matrix is (see Example 2.9)

$$\begin{bmatrix} 4x & 0 & 4x \\ 4x^3 + 4x^2 + 4x + 1 & 1 & 0 \\ x + 3 & 3x + 2 & 4 \end{bmatrix}.$$

The third row has minimal weighted row degree, namely, $L = 2$, and we conclude that the corresponding $Q(x, y)$ given by $Q(x, y) = x + 3 + (3x + 2)y + 4y^2$ is of minimal weighted degree 2. Factorization yields

$$Q(x, y) = 4(y - 3)(y - 3x - 4).$$

From this it follows that $m_1(x) = 3$ and $m_2(x) = 3x + 4$ are candidate message polynomials. Computing the corresponding codewords we find that both are at a Hamming distance of 2 from $\boldsymbol{r} = (4, 2, 3, 3, 3)$ so that both qualify as decoding solutions.

2) Minimal Choice of $V_i(x)$: A different algorithm is obtained by choosing $V_i(x)$ in such a way that the matrix $R_i(x)$ is weighted row reduced at each step. This is achieved by making sure that only one of the row degrees of $R_{i-1}(x)$ is increased when left multiplied by $V_i(x)$. In order to specify $V_i(x)$ we need to keep track of the vector of weighted row degrees

$$L_{i-1} := \begin{bmatrix} L_{i-1}(1) \\ \vdots \\ L_{i-1}(M+1) \end{bmatrix}$$

of the matrix $R_{i-1}(x)$. The following specification satisfies our requirement: Let j_* be the smallest integer for which $L_{i-1}(j_*)$ is minimal among $\{L_{i-1}(j) | \Delta_i(j) \neq 0\}$. Define $V_i(x)$ by

$$V_i(x) = (x - x_i)e_{j_*} e_{j_*}^T + \sum_{j \neq j_*} e_j (\Delta_i(j_*) e_j^T - \Delta_i(j) e_{j_*}^T) \quad (24)$$

and update the vector L_i as

$$L_i = L_{i-1} + e_{j_*}.$$

This choice of $V_i(x)$ produces an iterative algorithm that immediately leads to a weighted row reduced matrix and essentially coincides with the algorithm in [8]. It is interesting to note that for $q = 2$ the algorithm only differs from the Welch–Berlekamp algorithm in the initialization of L_0. This can be seen most easily by comparison with the behavioral interpretation of the Welch–Berlekamp algorithm, as given in [14], [17], [20].

B. General Multiplicities

For general values of s_i the specification of Step 2 is more complicated because there are other than exponential trajectories to be incorporated. For each i $(i = 1, \ldots, n)$ there are $s_i(s_i + 1)/2$ trajectories to be incorporated and thus Step 2 consists of $s_i(s_i + 1)/2$ substeps, indexed by (k, j) with $k = 0, \ldots, s_i - 1$ and $j = 0, \ldots, s_i - 1 - k$. In the following, the notation "$\mathrm{pred}(k, j)$" is used to denote the predecessor of index (k, j) in the lexicographical ordering. As initialization we have $R_i^{\mathrm{pred}(0,0)}(x) := R_{i-1}(x)$. Step 2 now becomes: proceed iteratively for $k = 0, \ldots, s_i - 1$ and $j = 0, \ldots, s_i - 1 - k$ as follows. Recall that the notation $\mathrm{D_H}$ is used for the Hasse derivative.

Step 2.1: Compute

$$\Delta_i^{(k,j)} = \mathrm{D}_{\mathrm{H}}^j R_i^{\mathrm{pred}(k,j)}(x_i) Y_i^k.$$

Step 2.2: Specify the update matrix $V_i^{(k,j)}(x)$.

Step 2.3: Compute $R_i^{(k,j)}(x) := V_i^{(k,j)}(x) R_i^{\mathrm{pred}(k,j)}(x)$.

Finally set $R_i(x) := R_i^{(s_i-1,0)}(x)$.

In the above procedure, the update matrix $V_i^{(k,j)}(x)$ should be chosen such that

$$V_i^{(k,j)}(\sigma)\boldsymbol{w} = 0$$

represents the MPUM of $\{\Delta_i^{(k,j)}\boldsymbol{w}_i^0\}$. This is due to the fact that, at each step, the error trajectory $R_i^{\mathrm{pred}(k,j)}(\sigma) Y_i^k \boldsymbol{w}_i^j$ equals $\Delta_i^{(k,j)}\boldsymbol{w}_i^0$ (use Remark 2.18 to see this).

The above procedure exactly follows the steps of Procedure 2.16 in Section II. It, therefore, follows immediately that for $i = 1, \ldots, n$ the matrices $R_i^{(k,j)}(x)$ in the above algorithm are such that $R_i^{(k,j)}(\sigma)\boldsymbol{w} = 0$ represents the MPUM of

$$\left\{ Y_i^k \boldsymbol{w}_i^j | 1 \le i \le n; 0 \le k \le s_i - 1; 0 \le j \le s_i - 1 - k \right\}.$$

In order to represent the MPUM of $\{\Delta_i^{(k,j)}\boldsymbol{w}_i^0\}$ we need to choose the update matrices $V_i^{(k,j)}(x)$ such that

$$V_i^{(k,j)}(x_i)\Delta_i^{(k,j)} = 0 \quad \text{and} \quad \det V_i^{(k,j)}(x) = x - x_i.$$

Below we focus on two of those choices, in analogy with Section IV-A.

1) Nonminimal Choice of $V_i^{(k,j)}(x)$: Here we aim to present a simple choice of $V_i^{(k,j)}(x)$ analogous to the $s = 1$ case of (22). Let j_* be the smallest integer for which $\Delta_i(j) \neq 0$. Define $V_i^{(k,j)}(x)$ by

$$V_i^{(k,j)}(x) := (x - x_i) e_{j_*} e_{j_*}^T + \sum_{j \neq j_*} e_j \left(e_j^T - \frac{\Delta_i(j)}{\Delta_i(j_*)} e_{j_*}^T \right).$$

As in (22), this specification does not require us to keep track of the row degrees. It may yield a matrix that is not weighted row reduced, as illustrated in the following example.

Example 4.3: Consider again the $(5, 2)$ RS code of Example 4.2 with received word $\boldsymbol{r} = (4, 2, 3, 3, 3)$. Let the multiplicities be specified as in [11] by $(s_1, s_2, s_3, s_4, s_5) = (2, 3, 2, 1, 1)$. Running the above algorithm for $M = 4$ yields a final matrix $R_5(x)$ given in the Appendix by (A1). Its vector of weighted row degrees is given by $L_5 = (9, 7, 7, 8, 8)$. Note that the sum of the weighted row degrees equals 39 which does not add up to

$$\deg \det R_5(x) + \deg \det \mathrm{diag}(1, x, x^2, x^3, x^4) = 14 + 10 = 24.$$

By Theorem 2.7 this shows that $R_5(x)$ is not weighted row reduced.

As illustrated by the preceding example the resulting matrix may not be weighted row reduced. It follows from Theorem 3.4 that in order to get a solution of our interpolation problem we need to bring the matrix into weighted row reduced form. For this we resort to the method outlined in Section II.

Example 4.4: Applying Theorem 2.8 and the procedure from the proof of Theorem 2.5 to the above matrix $R_5(x)$ yields weighted row reducedness. The resulting matrix is given in the Appendix by (A2). Its weighted row degrees are $4, 4, 5, 5$, and 6 which adds up to 24 (compare Theorem 2.7).

2) Minimal Choice of $v_i(x)$: It follows straightforwardly that $V_i^{(k,j)}(x)$ can be chosen in exactly the same way as in (24) to give rise to a weighted row reduced matrix $R_i^{(k,j)}$ at each step. Again, we need to keep track of the vector of weighted row degrees $L_i^{(k,j)}$ at each step. The resulting algorithm essentially coincides with the algorithm of [8]. The exact specification is as follows: Let j_* be the smallest integer for which $L_i^{\mathrm{pred}(k,j)}(j_*)$ is minimal among $\{L_i^{\mathrm{pred}(k,j)}(j) | \Delta_i^{(k,j)}(j) \neq 0\}$. Define $V_i^{(k,j)}(x)$ by

$$(x - x_i) e_{j_*} e_{j_*}^T + \sum_{j \neq j_*} e_j (\Delta_i^{(k,j)}(j_*) e_j^T - \Delta_i^{(k,j)}(j) e_{j_*}^T)$$

and update the vector $L_i^{(k,j)}$ as

$$L_i^{(k,j)} := L_i^{\mathrm{pred}(k,j)} + e_{j_*}.$$

Example 4.5: For the data of Example 4.3 the above algorithm produces a final matrix $R_5(x)$ given in the Appendix by (A.3). Its vector of weighted row degrees is given by $L_5 = (6, 5, 5, 4, 4)$. The sum of the weighted row degrees equals 24 so it follows from Theorem 2.7 that $R_5(x)$ is weighted row reduced. Another way to conclude weighted row reducedness is by observing that the leading weighted row coefficient matrix of $R_5(x)$ has full rank, see Definition 2.6. The last two rows of $R_5(x)$ have minimal weighted degree and yield interpolating polynomials

$$Q_4(x, y) = 3x^4 + 2x^2 + 3x + 3 + (4x^3 + 4x^2 + 3)y + (2x^2 + 3x + 2)y^2 + (x + 2)y^3$$

and

$$Q_5(x, y) = 2x^4 + 4x^3 + x^2 + 2x + 3 + (3x^3 + 2x + 1)y + 3x^2 y^2 + 4y^3 + 2y^4.$$

The polynomial $Q_4(x, y)$ corresponds to the minimal interpolating polynomial in the example of [11]—it is factorized as

$$2(y - 1 - x)(y - 3x - 4)(y + 3xy + 3x^2 + 3x + 1)$$

and yields candidate message polynomials $m_1(x) = 1 + x$ and $m_2(x) = 4 + 3x$ at Hamming distances 3 and 2, respectively. The polynomial $Q_4(x, y)$ was incorrectly labeled *the* minimal interpolating polynomial in [11]. Our algorithm shows that there are other minimal interpolating polynomials, for example, the polynomial $Q_5(x, y)$. This polynomial can be factorized as

$$2(y - 1 - x)(y - 3x - 4)(y^2 + 2y - xy + 2x^2 + x + 1).$$

It yields the same candidate message polynomials as the above polynomial $Q_4(x, y)$. In this example, any linear combination

$aQ_4(x,y)+bQ_5(x,y)$ with $a,b \in \mathbb{Z}_5$ is an interpolating polynomial of minimal weighted degree. On the other hand, any interpolating polynomial of minimal weighted degree can be written as $aQ_4(x,y)+bQ_5(x,y)$. This is discussed in Section V.

V. The Set of All Interpolating $Q(x,y)$'s

Theorem 3.4 provides a desired polynomial $Q(x,y)$ through a row of $R(x)$ of minimal weighted degree. The question arises whether the other rows of $R(x)$ contain valuable information. The first observation is that the rows of $R(x)$ provide a basis over $\mathbb{F}[x]$ of all polynomials

$$Q(x,y)=\sum_{j=0}^{M} q_j(x)y^j$$

that interpolate the data with the prescribed multiplicities and with $q_i(x)=0$ for $i>M$.

Theorem 5.1: Let $\mathfrak{B}$ and $R(x)$ be as in Theorem 3.4. Suppose that

$$Q(x,y)=\sum_{j=0}^{M} q_j(x)y^j$$

and $Q(x_i,y_i)=0$ with multiplicity at least $s_i, i=1,\ldots,n$. Define $q(x)=[\,q_0(x) \;\; \cdots \;\; q_M(x)\,]$. Then there exist a unique polynomial $a(x)\in \mathbb{F}^{1\times(M+1)}[x]$ such that

$$q(x)=a(x)R(x). \tag{25}$$

Conversely, any $q(x)$ as defined in (25) gives rise to a bivariate polynomial

$$Q(x,y)=\sum_{j=0}^{M} q_j(x)y^j$$

for which $Q(x_i,y_i)=0$ with multiplicity at least $s_i, i=1,\ldots,n$. Its weighted degree equals

$$L=\max_{\{i=1,\ldots,M+1\}}\{L_i+\deg a_i(x)\}$$

where $L_1,\ldots,L_{M+1}$ are the weighted row degrees of $R(x)$.

Proof: Since $Q(x_i,y_i)=0$ with multiplicity at least s_i it follows that for all $\boldsymbol{w}\in\mathfrak{B}$ there holds $q(\sigma)\boldsymbol{w}=0$. By Lemma 2.1, there exists a polynomial vector $a(x)$ such that $q(x)=a(x)R(x)$. Since $R(x)$ has full row rank there can be at most one such vector. The converse statement follows immediately from Theorem 3.4 and Corollary 2.11. □

In particular, it follows from Theorem 5.1 that all $Q(x,y)$'s of *minimal* weighted degree are stemming from a linear combination of rows of $R(x)$ of minimal weighted degree.

Let us now, for the sake of clarity, restrict ourselves to interpolation with multiplicity 1. In this case, polynomials $Q(x,y)$ of weighted degree L generate all message words $m(x)$ that correspond to codewords at distance less then $n-L$ to the received word. It may, however, also produce polynomials $m(x)$ that correspond to codewords at a larger distance than $n-L$. The reason that this possibility cannot be excluded is that Corollary 1.7 reflects an implication rather than an equivalence. Intuitively, we would like to use the parametrization in Theorem 5.1 to draw some conclusions about the message polynomials that the interpolating solutions corresponding to the rows of $R(x)$ have in common. Ideally this could help us narrow down the number of suitable message candidates. The following theorem is a first step in this direction. Part 2 of the theorem deals with the classical situation of $t<(n-\kappa+1)/2$ errors in which an interpolating polynomial of degree $\leq t+\kappa-1$ can be found. It shows that the rows of $R(x)$ that give rise to interpolating solutions of degree $\leq t+\kappa-1$ have the true message polynomial as their only intersection. Part 1 of the theorem is more general and shows that a common message polynomial stemming from rows of $R(x)$ of weighted degree $\leq t+\kappa-1$ is either unique or does not exist.

Theorem 5.2: Let (x_i,y_i), $i=1,\ldots,n$, be elements of $\mathbb{F}^2$ with the $x_i s$ mutually distinct. Let $\mathfrak{B}$ and $R(x)$ be as in Theorem 3.4. Define

$$Q_j(x,y)=\sum_{\ell=0}^{M} R_{j,\ell+1}(x)y^{\ell}.$$

Let $m(x)\in\mathbb{F}[x]$ with $\deg m(x)<\kappa$. Define $E=\{i \mid m(x_i)\neq y_i\}$ and $t=|E|$. Define $L=t+\kappa-1$.

1) Let $\tilde{m}(x)\in\mathbb{F}[x]$ with $\deg\tilde{m}(x)<\kappa$ be such that $Q_j(x,\tilde{m}(x))=0$ for all $Q_j(x,y)$ of weighted degree not exceeding L. Then $\tilde{m}(x)=m(x)$.
2) Suppose $L<\frac{n+\kappa-1}{2}$. Then $Q_j(x,\tilde{m}(x))=0$ for all $Q_j(x,y)$ with $\operatorname{wdeg} Q_j(x,y)\leq L$ if and only if $\tilde{m}(x)=m(x)$.

Proof:

1) Define

$$\lambda(x)=\prod_{i\in E}(x-x_i)$$

and

$$Q(x,y)=\lambda(x)(y-m(x)).$$

Then $\operatorname{wdeg} Q(x,y)=L$. By Theorem 5.1 we can write $Q(x,y)=\sum_j a_j(x)Q_j(x,y)$ with $\operatorname{wdeg} Q_j(x,y)\leq L$. Since $Q_j(x,\tilde{m}(x))=0$ we have that $y-\tilde{m}(x)$ divides $Q_j(x,y)$ and, therefore, $y-\tilde{m}(x)$ divides $y-m(x)$. It follows that $m(x)=\tilde{m}(x)$.

2) Sufficiency follows from Part 1. Notice that $n-t>\frac{n+\kappa-1}{2}$. Since

$$\deg Q_j(x,m(x))\leq L<\frac{n+\kappa-1}{2}$$

it follows that the number of zeroes of $Q_j(x,m(x))$ is strictly larger than its degree. Therefore, $Q_j(x,m(x))=0$.

□

Part 2 of the above theorem is illustrated by the next single-error correcting example.

Example 5.3: We take $\mathbb{F} = \mathbb{Z}_5$. The data are given by $(0,1)$, $(1,3)$, $(2,3)$, $(3,4)$, $(4,0)$. Running the algorithm in Section IV-A with all multiplicities set to one, $\kappa = 2$, and $M = 5$ yields $R(x)$ given by

$$\begin{bmatrix} 3x^4+x^2+3x+4 & 3x^2+4x+1 & 0 & 0 & 0 & 0 \\ 4x^2+1 & x+4 & 0 & 0 & 0 & 0 \\ 2x^2+4x+2 & 0 & 3 & 0 & 0 & 0 \\ 2x^3+4x^2+3x+1 & 2x+1 & 0 & 3 & 0 & 0 \\ x^3+4x^2+3x & x+2 & 0 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 & 0 & 3 \end{bmatrix}.$$

The minimal weighted row degree of $R(x)$ equals two. Both the second and the third row of $R(x)$ are of weighted row degree two. In the notation of Theorem 5.2 this yields two interpolating polynomials of minimal weighted degree: $Q_2(x,y) = 4x^2+1+(x+4)y$ and $Q_3(x,y) = 2x^2+4x+2+3y^2$. Simple calculations show that the only $m(x)$ for which $Q_2(x,m(x)) = Q_3(x,m(x)) = 0$ is given by $m(x) = x+1$, which is the true message polynomial.

The above result shows that the algorithm in Section IV-A can easily be used for classical decoding by performing the factorization step for all rows of minimal degree and then outputting the factor that the corresponding polynomials have in common. Note that in this case a minimal interpolating polynomial is not necessarily linear in y. This makes the method substantially different from other classical decoding methods such as the Welch–Berlekamp algorithm. An efficient implementation of the involved multiple factorization would still have to be investigated.

VI. Conclusion

In this paper we formulated the RS list decoding approach in a system-theoretic framework of behavioral modeling over finite fields. Interpolation with multiplicity was dealt with through the use of Hasse derivatives. The modularity of the modeling method allowed for the derivation of two decoding methods. One of these (Procedure 4.1 with Step 2 specified as in Section IV-B1) is noniterative in the data and can be interpreted as a generalization of the Euclidean algorithm as used for classical decoding. It relies on a system-theoretic matrix manipulation procedure, dating back to [33]. The other decoding method (Procedure 4.1 with Step 2 specified as in Section IV-B2) is iterative in the data and essentially coincides with the Nielsen–Høholdt algorithm of [8]. Its presentation is, however, different from [8] in that it explicitly keeps track of an $(M+1)\times(M+1)$ matrix of univariate polynomials, whereas the algorithm in [8] iteratively constructs a set of bivariate polynomials. An advantage of our system-theoretic matrix presentation is that it gives rise to parametrization results and, in our belief, yields conceptual clarity. It is a topic of current investigation how to use the parametrization results to help the decoding process. A first idea is to use them to limit the number of factors of interpolating solutions that need to be validated as valid message polynomials. A few preliminary results on this were presented in Section V. Another idea is to use the parametrization to find an interpolating solution $Q(x,y)$ of minimal weighted degree that also has minimal degree in y, so as to achieve few factors.

Acknowledgment

The authors would like to thank the reviewers for valuable comments and helpful suggestions.

Appendix

$$R_5(x) = \begin{bmatrix} 2x^9+2x^8+x^6+3x^5+3x^4+4x^2 & 0 & 0 & 0 & 0 \\ x^7+2x^6+2x^5+3x^4+2x^3+3x^2+2x & 4x^4+4x^3+2x & 0 & 0 & 0 \\ 2x^7+x^6+x^5+3x^4+4x^2+2x+2 & x^3+4 & 3x+2 & 0 & 0 \\ 4x^8+2x^7+4x^6+3x^4+2x^3+2x^2+3x+1 & x^3+2x^2+3x & 2 & 3 & 0 \\ 4x^8+x^6+x^5+3x^4+3x^3+4x^2+3x+2 & x^2+3x+3 & 3 & 0 & 3 \end{bmatrix}. \tag{A1}$$

The row reduced form of $R_5(x)$:

$$\begin{bmatrix} 3x^4+2x^2+3x+3 & 4x^3+4x^2+3 & 2x^2+3x+2 & x+2 & 0 \\ 4x^4+3x^3+2x^2+4x+1 & x^3+4x+2 & x^2 & 3 & 4 \\ 2x^5+2x^4+x^3+3x^2+3x+3 & 3x^3+2x^2+x+4 & x^3+x^2+3x+4 & 3x^2+3x+3 & 3x \\ 2x^5+4x^4+x^3+4x^2+x+1 & 2x^4+2x^3+4x^2+2x+3 & x^3+x^2+3x+1 & 3x^2+4x+2 & 2x+3 \\ 4x^6+4x^5+3x^4+3x^3+3x^2+x+2 & 2x^5+2x^4+2x^3+2x^2+4x+3 & 4x^3+3x+3 & 0 & 2x^2+3 \end{bmatrix} \tag{A2}$$

$$R_5(x) = \begin{bmatrix} x^6+x^5+4x^2+4 & 3x^4+2x^3+4x^2+4x+2 & x^3+2x^2+2 & x+4 & 0 \\ 2x^5+x^4+3x^3+x+4 & x^4+3x^3+2x^2+4x+1 & 4x^2+3x & 3 & 0 \\ 4x^5+4x^4+2x^3+2 & 3x^4+x^2+x+2 & 3x^3+4x^2+4x+3 & 3x+3 & 0 \\ 3x^4+2x^2+3x+3 & 4x^3+4x^2+3 & 2x^2+3x+2 & x+2 & 0 \\ 2x^4+4x^3+x^2+2x+3 & 3x^3+2x+1 & 3x^2 & 4 & 2 \end{bmatrix}. \tag{A3}$$

REFERENCES

[1] M. Sudan, "Decoding of Reed-Solomon codes beyond the error correction bound," *J. Compl.*, vol. 13, pp. 180–193, 1997.

[2] ——, "Decoding of Reed-Solomon codes beyond the error correction diameter," in *Proc. 35th Allerton Conf. Communication, Control, and Computing*, 1997.

[3] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1757–1768, Sept. 1999.

[4] W. Feng and R. E. Blahut, "Some results on the sudan algorithm," in *Proc. IEEE Int. Symp. Information Theory (ISIT'98)*, Cambridge, MA, Aug. 1998, p. 57.

[5] G.-L. Feng, "A generalization of the Welch-Berlekamp algorithm for weighted curve fitting with application to the sudan decoding procedure," in *Proc. 13th Symp. Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes (AAECC)*, HI, 1999, pp. 88–89.

[6] T. Høholdt and R. R. Nielsen, "Decoding hermitian codes with sudan's algorithm," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, M. Fossorier, H. Imai, S. Lin, and A. Poli, Eds. New York: Springer-Verlag, 1999, pp. 260–270.

[7] R. R. Nielsen, "A class of Sudan-decodable codes," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1564–1572, July 2000.

[8] R. Nielsen and T. Høholdt, "Decoding Reed-Solomon codes beyond half the minimum distance," in *Coding Theory, Cryptography and Related Areas*, J. Buchmann, T. Høholdt, T. Stichtenoth, and H. Tapia-Recillas, Eds. Berlin, Germany: Springer-Verlag, 2000, pp. 221–236.

[9] R. M. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Trans. Inform. Theory*, vol. 46, pp. 246–258, Jan. 2000.

[10] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," in *Proc. 2000 IEEE Int. Symp. Information Theory (ISIT'00)*, Sorrento, Italy, June 2000, p. 61.

[11] ——, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, submitted for publication.

[12] ——, "Decoding of Reed-Solomon codes for additive cost functions," in *Proc. 2002 IEEE Int. Symp. Information Theory (ISIT'02)*, Lausanne, Switzerland, 2002, p. 313.

[13] M. Kuijper, "The Berlekamp–Massey algorithm, error-correction, keystreams and modeling," in *Dynamical Systems, Control, Coding Computer Vision: New trends, Interfaces, and Interplay*. ser. Progress in Systems and Control Theory, G. Picci and D. Gilliam, Eds. Basel, Switzerland: Birkhäuser, 1999, pp. 321–342.

[14] ——, "Algorithms for decoding and interpolation," in *Codes, Systems, and Graphical Models*. ser. The IMA Volumes in Mathematics and its Applications, B. Marcus and J. Rosenthal, Eds. Berlin, Germany: Springer-Verlag, 2001, vol. 123, pp. 265–282.

[15] J. Feigenbaum, G. D. Forney, B. H. Marcus, R. J. McEliece, and A. Vardy, "Introduction to the special issue on codes and complexity," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1649–1657, Nov. 1996.

[16] M. Kuijper and J. C. Willems, "An algorithm for computing a shortest linear recurrence relation for a sequence of matrices: generalizing the Berlekamp-Massey algorithm," in *Proc. 1998 IEEE Int. Symp. Information Theory (ISIT'98)*, Cambridge, MA, Aug. 1998, p. 441.

[17] M. Kuijper, "A system-theoretic derivation of the Welch-Berlekamp algorithm," in *Proc. 2000 IEEE Int. Symp. Information Theory*, Sorrento, Italy, June 2000, p. 418.

[18] ——, "Further results on the use of a generalized B-M algorithm for BCH decoding beyond the designed error-correcting capability," in *Proc. 13th Symp. Applied Algebra Algebraic Algorithms, and Error-Correcting Codes (AAECC)*, HI, 1999, pp. 98–99.

[19] M. Kuijper and J. C. Willems, "On constructing a shortest linear recurrence relation," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 1554–1558, Nov. 1997.

[20] M. Kuijper, M. van Dijk, H. Hollmann, and A. J. Oostveen, "A unifying system-theoretic framework for errors-and-erasures Reed-Solomon decoding," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (Lecture Notes in Computer Science)*, S. Boztas and I. Shparlinski, Eds. Berlin, Germany: Springe-Verlag, 2001, vol. 2227, pp. 343–352.

[21] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Trans. Automat. Contr.*, vol. 36, pp. 259–294, Mar. 1991.

[22] R. E. Blahut, *Theory and practice of error control codes*. Reading, MA: Addison-Wesley, 1983.

[23] M. Kuijper and J. W. Polderman, "A behavioral approach to list decoding," in *Proc. 15th Int. Symp. Mathematical Theory of Networks and Systems*, D. Gilliam and J. Rosenthal, Eds., IN, 2002.

[24] ——, "Behavioral models for list decoding," *J. Math. Comput. Modeling of Dyn. Syst. (MCMDS)*, vol. 8, pp. 429–444, 2003.

[25] M. Kuijper, "Partial realization and the euclidean algorithm," *IEEE Trans. Automat. Contr.*, vol. 44, pp. 1013–1016, May 1999.

[26] R. Koetter and A. Vardy, "A complexity reducing transformation in algebraic list decoding of Reed-Solomon codes," in *Proc. 2003 Information Theory Workshopo*, Paris, France, 2003.

[27] H. Hasse, "Theorie der höheren Differentiale in einem algebraischen Funktionkörper mit volkommenen Konstantenkörper bei beliebiger Charakteristik," *J. Reine Angew. Math.*, vol. 175, pp. 50–54, 1936.

[28] G. Castagnoli, J. L. Massey, P. A. Schoeller, and N. v. Seemann, "On repeated-root cyclic codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 50–54, Jan. 1991.

[29] R. Lidl and H. Niederreiter, *Finite Fields*, 2 ed. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[30] J. W. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory: a Behavioral Approach*, ser. Texts in Applied Mathematics. New York: Springer-Verlag, 1997, vol. 26.

[31] M. Kuijper, *First-Order Representations of Linear Systems*, ser. Systems and Control: Foundations and Applications. Boston, MA: Birkhäuser, 1994.

[32] J. M. Schumacher, "Transformations of linear systems under external equivalence," *Linear Alg. its Applic.*, vol. 102, pp. 1–33, 1988.

[33] W. A. Wolovich, *Linear multivariable systems*. New York: Springer-Verlag, 1974.

[34] G. D. Forney Jr, "Minimal bases of rational vector spaces, with applications to multivariable linear systems," *SIAM J. Control*, vol. 13, pp. 493–520, 1975.

[35] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[36] J. C. Willems, "From time series to linear system. Part II: Exact modeling," *Automatica*, vol. 22, pp. 675–694, 1986.