# Behavioral interpolation for coding and control

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

**Abstract**

It is wellknown that classical decoding of Reed-Solomon error-correcting block codes is equivalent to system-theoretic minimal partial realization. In the first part of the paper we show how this type of decoding can also be formulated as minimal polynomial interpolation. We compare this type of interpolation with system-theoretic interpolation techniques that are used for control applications. We then present a procedure that achieves minimal polynomial interpolation by iteratively constructing a row reduced representation of an interpolating behavior.

Motivated by the need for improved decoding techniques, in particular soft-decision decoding, we turn to "list decoding" in the second part of the paper. Here the aim is to construct a list of all codewords that are within a pre-specified Hamming distance from the received word. A connection is made with recent work in the coding-theoretic literature that performs list decoding in two steps: in step one a minimal bivariate interpolating polynomial $Q(\xi, \eta)$ is constructed whereas in step two $Q(\xi, \eta)$ is factorized into factors that are linear in $\eta$.

We point out that this new development opens up yet another connection between coding theory and system theory, namely the connection between list decoding and minimal multivariable interpolation.

## 1 SISO behavioral interpolation, control and classical decoding

Interpolation techniques play an important role in various system-theoretic problems, see [1] and references therein. In particular, internal stability of a closed-loop system can be formulated as a set of interpolation constraints, see e.g. [8]. In its most general form, scalar minimal rational interpolation asks for a rational interpolant $y(\xi)$ of minimal McMillan degree that maps points $\xi_i$ to values $\eta_{i,j}$ with multiplicity $j$ as

$$y^{(j-1)}(\xi_i) = \eta_{i,j}$$

(for $i = 1, \ldots, \nu; j = 1, \ldots, j_i$). Here the McMillan degree of the rational interpolant $y(\xi) = n(\xi)/d(\xi)$ is defined as max $\{$ deg $n$, deg $d\}$. The above problem formulation comprises the minimal partial realization problem which arises as interpolation at one point $\xi_1 = 0$.

In the case where all interpolation points are distinct and all multiplicities equal one, the interpolant $y(\xi) = n(\xi)/d(\xi)$ is simply required to be of minimal McMillan degree and to satisfy

$$y(\xi_i) = \eta_i \qquad i = 1, \ldots, \nu.$$

Note that in this formulation the interpolant is required to be a rational function, i.e. any common factors between the polynomials $n$ and $d$ are to be cancelled. Thus this minimal *rational* interpolation problem differs from the minimal *polynomial* interpolation problem, which requires

$$d(\xi_i)\eta_i = n(\xi_i),$$

with max $\{$ deg $n$, deg $d\}$ minimal. In this paper we will see that it is the minimal polynomial interpolation problem (with the added constraint deg $n \leq$ deg $d$) which is important in a coding theoretic context: common factors between the polynomials $n$ and $d$ turn out to play a crucial role. In this paper we adopt an approach based on behavioral modeling and see that this approach is particularly suitable since it naturally generates solutions with common factors.

For preliminaries on the behavioral approach, in particular the theory of exact modeling, the reader is referred to [12]–[18],[21],[26]–[28]. In the behavioral approach, a system is essentially defined as a set of trajectories, taking values in a field $\mathbb{F}$. For system-theoretic applications, $\mathbb{F}$ is usually infinite ($\mathbb{R}$ or $\mathbb{C}$); in a coding-theoretic

context, $\mathbb{F}$ is a finite field. We will be concerned with linear shift-invariant behaviors on the time-set $\mathbb{Z}_+$ of the form $\mathcal{B} = \ker R(\sigma)$, where $R$ is a polynomial matrix of, say, size $p \times q$ and $\sigma$ is the backward shift operator:

$$\sigma(w_0, w_1, w_2, \ldots)) := (w_1, w_2, \ldots).$$

The behavior $\mathcal{B}$ consists of trajectories $\boldsymbol{w} : \mathbb{Z}_+ \mapsto \mathbb{F}^q$, for which

$$R(\sigma)\boldsymbol{w} = 0. \tag{1}$$

The representation (1) is called a *kernel representation* of $\mathcal{B}$.

In this paper we focus on a particular type of error-correcting block codes, namely Reed-Solomon codes. They are widely used in a range of applications, including the Compact Disc and deep space image transmission. It is wellknown that syndrome-based decoding of these codes has its equivalent in system theory, namely as minimal partial realization. In particular, it was shown in [18] how the Berlekamp-Massey algorithm can be interpreted as a special instance of the modeling procedure of [28, p. 289], involving a clever choice of update matrix at each step. This work formed the basis for the multivariable algorithm of [12] which was then used to achieve improved decoding of the related BCH codes in [13]–[15].

Recently, alternative decoding methods that are based on the original code definition by Reed & Solomon [22] have obtained increased attention. Let $\mathbb{F}$ be a finite field, consisting of $n$ elements, say $\{\xi_1, \ldots, \xi_n\}$. A $(n, \kappa)$ Reed-Solomon code is a $\kappa$-dimensional linear subspace of the space $\mathbb{F}^n$. The encoder maps vectors of length $\kappa$ ("message words") into vectors of length $n$ ("codewords"), thus providing redundancy and the possibility of retrieving the original message from codewords that are perturbed by noise. More particularly, a $(n, \kappa)$ Reed-Solomon code consists of codewords $\boldsymbol{c}$ given by

$$\boldsymbol{c} = (m(\xi_1), m(\xi_2), \ldots, m(\xi_n)), \tag{2}$$

where $m$ is a polynomial of degree $< \kappa$. Denoting the perturbed word by $\boldsymbol{r}$ (received word), error correction then amounts to curve fitting:

> given a received word $\boldsymbol{r} = (r_1, \ldots, r_n)$, find a polynomial $m$ of degree $< \kappa$ such that
>
> $$\begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} m(\xi_1) \\ \vdots \\ m(\xi_n) \end{bmatrix}$$
>
> for $n - \tau$ entries, with $\tau$ minimal.

In Reed & Solomon's original paper [22] this was solved by repeated Lagrange interpolation followed by majority voting, a method which is conceptually clear but computationally inefficient.

An alternative approach is readily obtained by reformulating the above curve fitting problem as an interpolation problem:

> given a received word $\boldsymbol{r} = (r_1, \ldots, r_n)$, find polynomials $e$ and $d$ such that
>
> $$d(\xi_i) r_i = e(\xi_i) \tag{3}$$
>
> for $i = 1, \ldots, n$ with
>
> 1. deg $d$ minimal
> 2. $e = dm$ with deg $m < \kappa$.

Note that the common factor $m$ between $d$ and $e$ is of crucial importance. In this section our aim is to reformulate this problem in a behavioral setting, see also [17] which is a summarized version of this section. In particular, we would like to simplify the above constraints 1) and 2) on the polynomials $d$ and $e$. For this purpose we re-encode the last $\kappa$ entries of $\boldsymbol{r}$, resulting in a codeword $\boldsymbol{c} = (c_1, \ldots, c_n)$ such that (recall that $\nu = n - \kappa$)

$$c_i = r_i \qquad \text{for } i = \nu + 1, \ldots, n.$$

Defining $\tilde{\boldsymbol{r}} := \boldsymbol{r} - \boldsymbol{c}$ we can, without restrictions, perform decoding on $\tilde{\boldsymbol{r}}$ since $\boldsymbol{r}$ and $\tilde{\boldsymbol{r}}$ differ by a codeword and are therefore disturbed by the same error pattern. In the following we use the fact that the last $\kappa$ entries of $\tilde{\boldsymbol{r}}$ are zero to reformulate our problem statement as a case of behavioral minimal modeling.

For $i = 1, \ldots, n$, we first introduce trajectories $\tilde{\boldsymbol{b}}_i : \mathbb{Z}_+ \mapsto \mathbb{F}^2$ given by

$$\tilde{\boldsymbol{b}}_i := \left( \begin{bmatrix} \tilde{r}_i \\ 1 \end{bmatrix}, \begin{bmatrix} \tilde{r}_i \xi_i \\ \xi_i \end{bmatrix}, \begin{bmatrix} \tilde{r}_i {\xi_i}^2 \\ {\xi_i}^2 \end{bmatrix}, \ldots \right).$$

Next, we define the polynomial $g$ as $g(\xi) := (\xi - \xi_{\nu+2}) \cdots (\xi - \xi_n)$ and we define $\mathcal{B}$ as the behavior spanned by trajectories

$$\bar{\boldsymbol{b}}_i := \begin{bmatrix} 1 & 0 \\ 0 & g(\sigma) \end{bmatrix} \tilde{\boldsymbol{b}}_i \qquad (i = 1, \ldots, n)$$

(recall that $\sigma$ denotes the backward shift operator). As a result of the fact that $g(\xi_i) = 0$ as well as $\tilde{r}_i = 0$, it follows that $\bar{\boldsymbol{b}}_i = 0$ for $i = \nu + 2, \ldots, n$, so that $\mathcal{B}$ is of

dimension $\nu+1$. Furthermore, for $i=1,\ldots,\nu+1$, we have

$$\bar{\boldsymbol{b}}_i := \left( \begin{bmatrix} \tilde{r}_i \\ g(\xi_i) \end{bmatrix}, \begin{bmatrix} \tilde{r}_i\xi_i \\ \xi_i g(\xi_i) \end{bmatrix}, \begin{bmatrix} \tilde{r}_i{\xi_i}^2 \\ {\xi_i}^2 g(\xi_i) \end{bmatrix}, \ldots \right).$$

Let us now define, for $i=1,\ldots,\nu+1$,

$$\eta_i := \frac{\tilde{r}_i}{g(\xi_i)} \quad \text{and} \quad \boldsymbol{b}_i := \frac{1}{g(\xi_i)}\bar{\boldsymbol{b}}_i$$

(note that $\eta_{\nu+1}=0$). Then for $i=1,\ldots,\nu+1$

$$\boldsymbol{b}_i = \left( \begin{bmatrix} \eta_i \\ 1 \end{bmatrix}, \begin{bmatrix} \eta_i\xi_i \\ \xi_i \end{bmatrix}, \begin{bmatrix} \eta_i{\xi_i}^2 \\ {\xi_i}^2 \end{bmatrix}, \ldots \right)$$

and the decoding problem is now readily formulated as the problem of finding a minimal representation

$$R(\sigma)\boldsymbol{w} = 0 \tag{4}$$

for $\mathcal{B}=\{\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{\nu+1}\}$. Indeed, the row $[d \quad -n]$ of the polynomial matrix $R$ that has minimal row degree and for which deg $n \leq$ deg $d$ gives rise to error locations and values as follows: the error locations $\alpha_1,\ldots,\alpha_\ell$ are the zeros of $d$, whereas the error values are obtained from the polynomial $\tilde{n} := ng$, namely by writing $\tilde{n} = \tilde{m}d$ with deg $\tilde{m} < \kappa$ and calculating $e_j := \tilde{r}_j - \tilde{m}(\alpha_j)$ $(j=1,\ldots,\ell)$. In other words, the decoding problem can be reformulated as follows:

> Given a received word $\boldsymbol{r} = (r_1,\ldots,r_n)$, compute $\eta_i$ as above for $i=1,\ldots,\nu+1$. Now find polynomials $d$ and $n$ such that
>
> $$d(\xi_i)\eta_i = n(\xi_i) \tag{5}$$
>
> for $i=1,\ldots,\nu+1$ with deg $d$ minimal and deg $n \leq$ deg $d$.

Observe that we are thus concerned with minimal polynomial interpolation at distinct points $\xi_1,\ldots,\xi_{\nu+1}$ and that common factors between $d$ and $n$ are of importance.

Let us now construct an algorithm that computes a minimal, i.e. row reduced representation (4) for the behavior $\mathcal{B}$. We outline an iterative algorithm that, at each step $k$, constructs a row reduced representation $R_k(\sigma)\boldsymbol{w}=0$ of the MPUM corresponding to the interpolation data $(\xi_i,\eta_i)$ $(i=1,\ldots,k)$, i.e. the interpolation data processed so far. The algorithm is structured along the lines of the general iterative modeling procedure of [28, p. 289] and strongly resembles the Welch-Berlekamp algorithm [25, 4, 5]. Like the Berlekamp-Massey algorithm [2, 3, 19, 18], our algorithm makes use of the solution's degree $\ell$ at each step to determine which type of update matrix is used. In this respect it differs from the Welch-Berlekamp algorithm which uses a different integer parameter. At each step $k$, our algorithm produces a row reduced matrix $R_k$ whose first row contains the solution corresponding to the interpolation data processed so far.

**Algorithm 1.1**

*For* $k=0,\ldots,\nu$ *denote* $R_k := \begin{bmatrix} d_k & -n_k \\ \omega_k & -q_k \end{bmatrix}$. *Initially define*

$$R_0 := \begin{bmatrix} 1 & 0 \\ 0 & \xi-\xi_{\nu+1} \end{bmatrix}, \text{ and } \ell_0 := 0.$$

*Proceed iteratively as follows for* $k=1,\ldots,\nu$. *Compute, after processing* $(\xi_i,\eta_i)$ *for* $i=0,\ldots,k$, *the numbers* $\Delta_k$ *and* $\Gamma_k$ *as follows:*

$$\begin{aligned} \Delta_k &:= d_{k-1}(\xi_k)\eta_k - n_{k-1}(\xi_k) \\ \Gamma_k &:= \omega_{k-1}(\xi_k)\eta_k - q_{k-1}(\xi_k). \end{aligned}$$

*Compute the matrix* $R_k$ *and the integer* $\ell_k$ *as follows:*

$$R_k := V_k R_{k-1},$$

*where, if* $\Delta_k \neq 0$ *and* $(\ell_{k-1} < k/2 \quad \text{or } \Gamma_k = 0)$,

$$V_k(s) := \begin{bmatrix} \xi-\xi_k & 0 \\ -\Gamma_k & \Delta_k \end{bmatrix}; \quad \ell_k := \ell_{k-1}+1,$$

*and, if otherwise,*

$$V_k(s) := \begin{bmatrix} \Gamma_k & -\Delta_k \\ 0 & \xi-\xi_k \end{bmatrix}; \quad \ell_k := \ell_{k-1}.$$

The next theorem shows that the above algorithm produces a solution to equation (5) under accompanying constraints.

**Theorem 1.1** *Let the above algorithm operate on the interpolation data* $(\xi_i,\eta_i)$ $(i=1,\ldots,\nu+1)$, *as given above. Then for* $k=1,\ldots,\nu$ *the polynomials* $d_k$ *and* $n_k$ *satisfy*

$$d_k(\xi_i)\eta_i = n_k(\xi_i) \qquad \text{for } i=1,\ldots,k,\nu+1$$

*with $\ell_k = \deg d_k$ minimal and $\deg n_k \leq \deg d_k$. In particular, $d_\nu$ and $n_\nu$ are a solution of equation (5) with $\ell_\nu = \deg d_\nu$ minimal and $\deg n_\nu \leq \deg d_\nu$.*

*Furthermore, a parametrization of all solutions of row degree $\ell$ is obtained from*

$$[d \quad -n] := \begin{bmatrix} q_1 & q_2 \end{bmatrix} R_\nu,$$

*where $q_1$ is a polynomial of degree $\ell - \ell_\nu$ and $q_2$ is a polynomial of degree $\ell - ((\nu+1)/2 - \ell_\nu)$. In particular, uniqueness occurs if and only if*

$$\ell_\nu < (\nu+1)/2.$$

**Proof** Define trajectories $\boldsymbol{b}_i : \mathbb{Z}_+ \mapsto \mathbb{F}^2$ as above ($i = 1, \ldots, \nu+1$):

$$\boldsymbol{b}_i = \left( \begin{bmatrix} \eta_i \\ 1 \end{bmatrix}, \begin{bmatrix} \eta_i \xi_i \\ \xi_i \end{bmatrix}, \begin{bmatrix} \eta_i \xi_i{}^2 \\ \xi_i{}^2 \end{bmatrix}, \ldots \right).$$

In the following we show that Algorithm 1.1 is a special instance of the general iterative modeling procedure of [28, p. 289] applied to the data set $\{\boldsymbol{b}_{\nu+1}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_\nu\}$, starting with $R_{-1} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Then the error trajectory $\boldsymbol{e}_0 = \boldsymbol{b}_{\nu+1}$, whereas for $k = 1, \ldots, \nu$, $\boldsymbol{e}_k = R_{k-1}(\sigma)\boldsymbol{b}_k$ is given by

$$\boldsymbol{e}_k = \left( \begin{bmatrix} \Delta_k \\ \Gamma_k \end{bmatrix}, \begin{bmatrix} \Delta_k \xi_k \\ \Gamma_k \xi_k \end{bmatrix}, \begin{bmatrix} \Delta_k \xi_k^2 \\ \Gamma_k \xi_k^2 \end{bmatrix}, \ldots \right).$$

Here $\Delta_k$ and $\Gamma_k$ are given as in Algorithm 1.1. In particular, $\Delta_0 = 0$, $\Gamma_0 = 1$ and $R_0(\sigma)\boldsymbol{w} = 0$ is a minimal representation of the MPUM of $\{\boldsymbol{e}_0\}$. Further, the update matrices $V_k$, defined in Algorithm 1.1, represent the MPUM for $\{\boldsymbol{e}_k\}$ ($k = 1, \ldots, \nu$), so that $R_k(\sigma)\boldsymbol{w} = 0$ represents the MPUM for $\{\boldsymbol{b}_{\nu+1}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$. In the algorithm the integer $\ell_k$ denotes the first row degree of $R_k$. Denoting the second row degree of $R_k$ by $\tilde{\ell}_k$, we have $\ell_k + \tilde{\ell}_k = k+1$, so that the algorithm's condition $\ell_{k-1} < k/2$ is nothing else than $\ell_{k-1} < \tilde{\ell}_{k-1}$. It now follows by induction that the choice of $V_k$'s ensures that each $R_k$ has minimal row degrees. Indeed, this holds trivially for $k = 1$ and the assumption that $R_{k-1}$ has minimal row degrees implies that $R_k$ has minimal row degrees because of the fact that $V_k$ increases the degree of only one row of $R_{k-1}$ by 1. It also follows by induction that $\deg q_k > \deg \omega_k$, so that, because of row reducedness of $R_k$,

$$\deg n_k \leq \deg d_k = \ell_k \qquad k = 1, \ldots, \nu.$$

Now any solution of (5) of smaller degree that satisfies the accompanying constraints leads to a row reduced MPUM representation with smaller sum of row degrees. This contradicts the minimality of the row degrees of the matrix $R_k$ and proves the theorem (the parametrization follows straightforwardly from Theorem 2.5.4 in [21]).

## 2 List decoding and multivariable behavioral interpolation

The type of decoding that we considered up till now is a classical type of decoding in the sense that the method aims at the derivation of a unique solution $(d(\xi), n(\xi))$. By Theorem 1.1 this solution is the true solution whenever the number of errors that actually occurred is less than $(\nu+1)/2$, i.e. $(n-\kappa+1)/2$.

In recent years, a new approach to Reed-Solomon decoding has been initiated by Sudan [23, 24] who, inspired by the Welch-Berlekamp algorithm, introduced an interpolation-based method for performing so-called *list decoding* for Reed-Solomon codes. In this type of decoding the decoding radius around a received word is increased beyond $(n-\kappa+1)/2$, thus allowing for multiple solutions. Decoding $\tau$ errors may then result in a *list* of several codewords, all within distance $\tau$ from the received word. List decoding is particularly important as it naturally leads to soft-decision decoding, enabling the decoder to pick the most likely codeword based on reliability information of all codeword components.

Sudan's list decoding method consists of two parts, namely *interpolation* and *factorization*. To understand the method, we should first realize that condition (3) is readily reformulated as

$$Q(\xi_i, \eta_i) = 0 \qquad \text{for } i = 1, \ldots, n$$

for $Q(\xi, \eta) := d(\xi)\eta - e(\xi)$. The most straightforward way to get from classical decoding to list decoding would be to use the parametrization of Theorem 1.1. We are then essentially creating bivariate polynomials of the type $d(\xi)\eta - e(\xi)$ of increasing degree and arrive at feasible candidates by checking the zeros of each $d(\xi)$. Apart from being computationally intensive, it turns out that this method does not use the available information in an efficient way, as shown by the exposition below.

In Sudan's approach, the step from classical decoding to list decoding is accomplished by allowing $Q(\xi, \eta)$ to be *nonlinear* in $\eta$. Thus, the basic strategy is to find a nonzero bivariate polynomial $Q(\xi, \eta) = \Sigma_{i,j}\, q_{ij} \xi^i \eta^j$ for

which

$$Q(\xi_i, \eta_i) = 0 \qquad \text{for } i = 1, \ldots, n. \tag{6}$$

This can be solved if we allow for $> n$ monomials in $Q$. If we constrain the "degree" of $Q$ to a certain upper bound, we then need to make sure that this bound is high enough. For our notion of "degree" we use the following definition (in e.g. [24]):

**Definition 2.1** *For positive integers $w_x$ and $w_y$, the* $(w_x, w_y)$-weighted degree *of a bivariate polynomial* $Q(\xi, \eta) = \Sigma_{i,j}\ q_{ij}\xi^i\eta^j$ *is defined to be max* $\{iw_x + jw_y \mid q_{ij} \neq 0\}$.

Let $\delta^*$ be the smallest integer such that the $(1, \kappa - 1)$-weighted degree of $\delta^*$ allows for $> n$ monomials. Then we can trivially find $Q(\xi, \eta)$ of $(1, \kappa-1)$-weighted degree $\leq \delta^*$ such that (6) holds. It is less trivial to find $Q(\xi, \eta)$ of *minimal* $(1, \kappa - 1)$-weighted degree, say $\ell$, see [20] for an algorithm. In the case that $\tau < n - \ell$ errors occurred, there exists a polynomial $m$ of degree $< \kappa$ such that $m(\xi_i) = \eta_i$ for $> \ell$ locations. As a result, the polynomial $Q(\xi, m(\xi)$ has $> \ell$ zeros and is thus of degree $> \ell$. On the other hand, because of the fact that the $(1, \kappa - 1)$-weighted degree of $Q(\xi, \eta)$ equals $\ell$, we have that deg $Q(\xi, m(\xi) \leq \ell$. We conclude that $Q(\xi, m(\xi) \equiv 0$, i.e. $\eta - m(\xi)$ divides $Q(\xi, \eta)$.

Because of the above, the second step in Sudan's strategy is to factorize $Q$ into linear factors $\eta - m(\xi)$. Note that this approach thus aims directly at the message polynomial and does not involve an intermediate step (crucial in classical decoding) in which error locations and values are derived. It is easy to verify that for $\tau < (n - \kappa + 1)/2$, the polynomial $Q(\xi, \eta)$ of minimal $(1, \kappa - 1)$-weighted degree necessarily has the form $d(\xi)(\eta - m(\xi))$—in this case minimization and factorization are equivalent.

Algorithms that achieve the above factorization can be found in e.g. [20] and references therein, see also references in [10].

Sudan's approach is extended in [7] by requiring the interpolation condition $Q(\xi_i, \eta_i) = 0$ to hold with prespecified multiplicity $s$, meaning that the smallest degree monomial in $Q(\xi + \xi_i, \eta + \eta_i)$ has degree $s$. Obviously this imposes $s(s + 1)/2$ constraints per interpolation point and leads to list decoding of $\tau < n - \ell/s$ errors. An upper bound on the probability that the list contains more than one codeword is derived in [20] and is found to be very small so that Sudan's approach usually leads to unique decoding beyond $(n - \kappa + 1)/2$ errors, a result which is a major breakthrough in this area. For increasing multiplicity $s$ and codelength $n$ it can be proven that any fraction $\tau/n \leq 1 - \sqrt{\kappa/n}$ of errors can be corrected. By comparison, classical decoding gives an error rate of $\tau/n = \frac{1-\kappa/n}{2}$.

All of the above is concerned with hard-decision decoding. In the recent papers [9, 10] a soft-decision decoding procedure is presented that builds on the above interpolation approach. More specifically, soft decision reliability information is converted in a probabilistically optimal way into a set of interpolation points with varying multiplicities.

As outlined above, the step from classical decoding to list decoding is accomplished by allowing $Q(\xi, \eta)$ to be nonlinear in $\eta$. From a behavioral point of view this can be seen as a conversion from a scalar minimal interpolation problem to a multivariable interpolation problem. Indeed, the aim is no longer to derive just two polynomials $d_0(\xi)$ and $d_1(\xi)$, leading to $Q(\xi, \eta) = d_1(\xi)\eta + d_0(\xi)$, but to derive $m + 1$ polynomials that make up $Q(\xi, \eta) = d_m(\xi)\eta^m + d_{m-1}(\xi)\eta^{m-1} + \cdots + d_0(\xi)$, see also [6]. It is a topic of current investigation to convert this problem into a multivariable minimal behavioral interpolation problem in the most general case of soft-decision decoding with varying multiplicities. Such a reformulation would then allow for an efficient and insightful algorithm in the form of the multivariable extension of Algorithm 1.1. In particular it would be of interest to establish a connection with the algorithms of [6, 20].

## 3 Conclusions

The aim of this line of research is to create a concise framework for soft- and hard-decision decoding on the basis of minimal interpolation. For this, we seek to extend our formulation of classical decoding as scalar minimal behavioral interpolation to a formulation of list decoding as multivariable minimal behavioral interpolation. This opens up the possibility of discovering more connections between coding theory and system/control theory—in particular, it would be interesting to relate to applications in control that allow for a similar usage of bivariate polynomials.

## References


[1] Antoulas, A.C., J.A. Ball, J. Kang and J.C. Willems, *On the solution of the minimal rational interpolation problem*, Linear Alg. Appl., **137**, 1990, 511-573.

[2] Berlekamp, E.R., *Algebraic Coding Theory*, New York, McGraw-Hill, 1968.

[3] Blahut, R.E., *Theory and Practice of Error Control Codes*, Addison-Wesley, 1983.

[4] Blackburn, S.R., *A generalized rational interpolation problem and the solution of the Welch-Berlekamp algorithm*, Designs, Codes and Cryptography, **11**, 1997, 223-234.

[5] Chambers, W.G., R.E. Peile, K.Y. Tsie and N. Zein, *Algorithm for solving the Welch-Berlekamp key-equation, with a simplified proof*, Electronics Letters, **29**, 1993, 1620-1621.

[6] G-L. Feng, G-L., *A generalization of the Welch-Berlekamp algorithm for weigthed curve fitting with application to the Sudan decoding procedure*, in Proceedings of the 13th Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes (AAECC), Hawaii, USA, 1999, 88-89.

[7] Guruswami, V. and M. Sudan, *Improved decoding of Reed-Solomon and algebraic-geometric codes*, IEEE Trans. Info. Theory, **45**, no. 6, 1999, 1757-1768.

[8] Kimura, H., *Robust stabilizability for a class of transfer functions*, IEEE Trans. Aut. Control, **29**, 1984, 788-793.

[9] Kötter, R. and A. Vardy, *Algebraic soft-decision decoding of Reed-Solomon codes*, in *Proceedings 2000 IEEE International Symposium on Information Theory* (ISIT'00, Sorrento, Italy, 2000), 61.

[10] Kötter, R. and A. Vardy, *Algebraic soft-decision decoding of Reed-Solomon codes*, Draft manuscript, 2000.

[11] Kuijper, M., *First-Order Representations of Linear Systems*, Series on "Systems and Control: Foundations and Applications", Birkhäuser, Boston, 1994.

[12] Kuijper, M., *An algorithm for constructing a minimal partial realization in the multivariable case*, Systems & Control Letters, **31**, 1997, 225-233.

[13] Kuijper, M., *The Berlekamp-Massey algorithm, error-correction, keystreams and modeling*, in *Dynamical Systems, Control, Coding, Computer Vision: New trends, Interfaces, and Interplay*, G. Picci, D. S. Gilliam eds., Birkhäuser's series "Progress in Systems and Control Theory", (1999) 321-342.

[14] Kuijper, M., *Parametrizations and finite options*, in "The Mathematics of Systems and Control: from Intelligent Control to Behavioral Systems" (Festschrift on the occasion of the 60th birthday of Jan C. Willems), H.L. Trentelman, J. W. Polderman eds., ISBN 90-367-1112-6, 1999, 59-72.

[15] Kuijper, M., *Further results on the use of a generalized B-M algorithm for BCH decoding beyond the designed error-correcting capability*, in "Proceedings of the 13th Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes (AAECC)", Hawaii, USA, 1999, 98-99.

[16] Kuijper, M., *Algorithms for Decoding and Interpolation*, in *Codes, Systems, and Graphical Models* , Institute for Mathematics and its Applications (IMA) Series, G.D. Forney Jr., J. Rosenthal, B. Marcus and A. Vardy (eds.), Springer-Verlag, 265-283, 2000.

[17] Kuijper, M., *A system-theoretic derivation of the Welch-Berlekamp algorithm*, in *Proceedings 2000 IEEE International Symposium on Information Theory* (ISIT'00, Sorrento, Italy, 2000), 418.

[18] Kuijper, M. and J.C. Willems, *On constructing a shortest linear recurrence relation*, IEEE Trans. Aut. Control, **42**, 1997, 1554-1558.

[19] Massey, J.L. (1969). *Shift-register synthesis and BCH decoding*, IEEE Trans. Info. Theory, **15**, 1969, 122-127.

[20] Nielsen, R.R. and T. Hoeholdt, *Decoding Reed-Solomon codes beyond half the minimum distance*, Draft manuscript, 1999, `http://www.student.dtu.dk/p938546`

[21] Polderman, J.W. and J.C. Willems, *Introduction to Mathematical Systems Theory—a behavioral approach*, Springer Verlag, New York, 1998.

[22] Reed, I.S. and G. Solomon. *Polynomial codes over certain finite fields*, SIAM Journal on Applied Mathematics, **8**, 1960, 300-304.

[23] Sudan, M., *Decoding of Reed-Solomon codes beyond the error correction bound*, Journal of Complexity, **13**, 1997, 180-193.

[24] Sudan, M., (1997). *Decoding of Reed-Solomon codes beyond the error correction diameter*, in "Proceedings of the 35th Allerton Conference on Communication, Control and Computing", 1997, `http://theory.lcs.mit.edu/ madhu/papers.html`.

[25] Welch, L.R. and E.R. Berlekamp, *Error correction for algebraic block codes*, U.S. Patent 4 633 470, issued Dec. 30 1986.

[26] Willems, J.C., *From time series to linear system. Part I: Finite-dimensional linear time invariant systems.*, Automatica, **22**, 1986, 561-580.

[27] Willems, J.C., *From time series to linear system. Part II: Exact modeling*, Automatica, **22**, 1986, 675-694.

[28] Willems, J.C., *Paradigms and puzzles in the theory of dynamical systems*, IEEE Trans. Aut. Control, **36**, 1991, 259-294.