

Algorithms for Decoding and Interpolation

Margreet Kuijper

Dept. EE Engineering, University of Melbourne

margreet@ee.mu.oz.au

IMA Workshop '99

Overview

- Introduction:
 - interpolation in a systems & control context
 - \updownarrow
 - interpolation in a coding context
- RS/BCH codes
 - A The original approach
 - B The textbook approach
- Various “key equations”
- Connections
 - (\rightarrow Welch-Berlekamp interpolation)
- Conclusions
- Further research

Why interpolation?

In systems & control:

↔ feedback stabilization

e.g. Dorato et al. '89

↔ robust stabilization

e.g. Kimura '84

In coding theory:

↔ decoding of generalized RS/BCH codes

What is interpolation?

In systems & control:

Minimal rational interpolation problem:

Given n distinct points x_i and N values $y_{i,j}$, find a rational function $y(s)$ of minimal McMillan degree, such that

$$y^{(j)}(x_i) = y_{i,j}.$$

(Here: writing $y(s) = \frac{n(s)}{d(s)}$, the **McMillan degree** of $y(s)$ is defined as $\max\{\deg d(s), \deg n(s)\}$).

Then requirements on interpolant: e.g. properness, stability, norm bound...

Note that the **rational** interpolation problem:

Find a rational function $y(s) = \frac{n(s)}{d(s)}$ of minimal McMillan degree, such that

$$\frac{n(x_i)}{d(x_i)} = y_i \quad \text{for } i = 1, \dots, N$$

differs from the **polynomial** interpolation problem:

Find two polynomials $d(s)$ and $n(s)$ with $\max\{\deg d(s), \deg n(s)\}$ minimal such that

$$n(x_i) = y_i d(x_i) \quad \text{for } i = 1, \dots, N$$

Example: Interpolation data:

$$\begin{aligned}x_1 = 0 & \rightarrow y_1 = 1 \\x_2 = 1 & \rightarrow y_2 = 0 \\x_3 = -1 & \rightarrow y_3 = 1\end{aligned}$$

then unique minimal *polynomial* solution of degree 1, namely $(d(s), n(s)) = (s - 1, s - 1)$, but *rational* interpolants are of degree ≥ 2 , e.g.

$$y(s) = \frac{s^2 - 2s + 1}{-3s + 1}$$

Special case:

Given N values y_1, \dots, y_N , find a rational function $y(s)$ of minimal McMillan degree, such that

$$y^{(j-1)}(0) = y_j \quad \text{for } j = 1, \dots, N$$

=

Minimal partial realization problem

Given N values a_1, \dots, a_N , find a rational function $t(s)$ of minimal McMillan degree, such that

$$t(s) = a_1 + a_2 s^{-1} + \dots + a_N s^{-N+1} + s^{-N}(\dots)$$

Literature in systems theory:

Minimal partial realization:

- e.g.*
- Ho & Kalman '65
 - Dickinson, Morf & Kailath '74
 - Gragg & Lindquist '83
 - Antoulas '86 '94
 - Kuijper & Willems '97
 - Kuijper '97, '99

Minimal rational interpolation:

- e.g.*
- Antoulas, Ball, Kang, Willems '90
 - Antoulas & Anderson '86 '90
 - Antoulas & Willems '90

Literature in coding theory:

Various “key equations” and interpolation algorithms in

- e.g.*
- Berlekamp, Massey '68
 - Sugiyama et al. '75
 - Welch & Berlekamp '83
 - Morii & Kasahara '92
 - Sorger '93
 - Chambers et al. '93
 - Fitzpatrick '95
 - Berlekamp '96

Connections coding theory \leftrightarrow system theory:

- e.g.*
- Sain '75
 - Kuijper & Willems '97
 - Blackburn '97
 - Kuijper '99

Defining RS/BCH codes

A The original approach (Reed & Solomon '60):

codeword = set of q points on a curve

decoding = curve fitting

More specifically: write $GF(q) = \{0, \alpha, \dots, \alpha^{q-1} = 1\}$.

A codeword in a (q, k) RS code is defined as

$$c = (M(0), M(\alpha), \dots, M(\alpha^{q-1}))$$

where $M(s)$ is some message polynomial of degree

$< k$.

Decoding: given received word r , find $M(s)$ of degree $< k$ such that

$$\begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{q-1} \end{bmatrix} = \begin{bmatrix} M(0) \\ M(\alpha) \\ \vdots \\ M(\alpha^{q-1}) \end{bmatrix}$$

for $q - e$ entries, with e minimal.

Reed & Solomon '60: solve by repeated Lagrange **interpolation** = majority logic decoding

Then solution unique if $2e < d_{min} = q - (k - 1)$.

But not efficient.

Reformulation:

Given received word r , find polynomials $N(s)$ and $D(s)$ such that

$$N(x_i) = r_i D(x_i)$$

for $x_i = \alpha^i$ ($i = 0, \dots, q - 1$) with

- deg D minimal

($D(s)$ = error locator polynomial)

- $N(s) = M(s)D(s)$ with deg $M < k$

Note: Common factor $M(s)$ is crucial

Intermezzo I

Bounded distance decoding: Given a-priori specified e , and received word r , find *all* polynomials $M(s)$ of degree $< k$ such that

$$\begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{q-1} \end{bmatrix} = \begin{bmatrix} M(0) \\ M(\alpha) \\ \vdots \\ M(\alpha^{q-1}) \end{bmatrix}$$

for $q - e$ entries

Shokrollahi & Wasserman '99: A code is called (e, b) -decodable if there are $\leq b$ such polynomials $M(s)$ for any received word r

Intermezzo II: generalization of RS code

(→ shortened RS code/Goppa code/alternant code)

Let $A = (\alpha_1, \alpha_2, \dots, \alpha_n)$ with α_i 's distinct elements in $GF(q)$ and let $V = (v_1, v_2, \dots, v_n)$ with v_i 's nonzero elements in $GF(q)$

Then a (n, k) **generalized RS code**, denoted by $GRS_k(A, V)$, consists of codewords

$$c = (v_1 M(\alpha_1), v_2 M(\alpha_2), \dots, v_n M(\alpha_n))$$

where $M(s)$ is some message polynomial of degree
 $< k$

So $GRS_k(A, (1, 1, \dots, 1))$ consists of codewords

$$c = (M(1), M(\alpha), \dots, M(\alpha^{n-1}))$$

Theorem Let $n = q - 1$ and $A = (1, \alpha, \dots, \alpha^{n-1})$. Then the dual of $GRS_k(A, (1, 1, \dots, 1))$ is $GRS_{n-k}(A, (1, \alpha, \dots, \alpha^{n-1}))$

Corollary A parity-check matrix H for $GRS_k(A, (1, 1, \dots, 1))$ is given by

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \dots & (\alpha^2)^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{n-k} & (\alpha^{n-k})^2 & \dots & (\alpha^{n-k})^{n-1} \end{bmatrix}$$

Then codeword $c \leftrightarrow$ polynomial $c(s)$ with zeros at $\alpha, \alpha^2, \dots, \alpha^{n-k}$

Defining RS/BCH codes

B The textbook approach (Gorenstein & Zierler '61):

codeword = polynomial with prescribed zeros

decoding = finding shortest LFSR for
syndrome

More specifically: A codeword in a $(n, n - N)$ RS code is defined by $c(s)$ of degree $< n$ with zeros at $\alpha, \alpha^2, \dots, \alpha^N$

Decoding: given received word $r(s)$, compute syndromes

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} r(\alpha) \\ r(\alpha^2) \\ \vdots \\ r(\alpha^N) \end{bmatrix}$$

and solve **Key equation 1**

$$(a_1s + a_2s^2 + \cdots + a_Ns^N)C(s) = P(s) \bmod s^{N+1}$$

such that $C(0) = 1$ and $\max \{\deg C, \deg P\}$ is minimal

= **minimal rational interpolation at $s = 0$**

$$C(s) = \prod_{\text{error locations}} (\alpha^i s - 1)$$

Behavioral approach for solving Key equation 1:

I Basic idea

Consider $R(\sigma)\mathbf{w} = 0$ with $\mathbf{w} : \mathbb{Z}_- \mapsto \mathbb{R}^2$

σ : forward shift, and

$$R(s) = \begin{bmatrix} 1 & a_1s + a_2s^2 + \cdots + a_Ns^N \\ 0 & s^{N+1} \end{bmatrix}$$

This represents “smallest model” that contains

$$\left(\cdots, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} a_1 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} a_N \\ 0 \end{bmatrix} \right)$$

II Behavioral theory of exact modeling

Behavioral approach (Jan C. Willems, 1983...)

Basic set-up:

“system” \neq transfer function

but

“system” = behavior \mathcal{B} = set of trajectories
in time

Here:

$$\mathcal{B} = \{w : \mathbb{Z}_- \mapsto \mathbb{R}^q \mid R(\sigma)w = 0\}$$

Basic Lemma: Let

$$\mathcal{B} = \{\mathbf{w} \mid R(\sigma)\mathbf{w} = 0\} \quad \text{and}$$

$$\tilde{\mathcal{B}} = \{\mathbf{w} \mid \tilde{R}(\sigma)\mathbf{w} = 0\}$$

Then $\mathcal{B} \subset \tilde{\mathcal{B}}$ if and only if there exists a polynomial matrix $F(s)$ such that

$$\tilde{R}(s) = F(s)R(s)$$

Corollary: Parametrization of representations of \mathcal{B} :

$$\mathcal{B} = \{\mathbf{w} \mid U(\sigma)R(\sigma)\mathbf{w} = 0\},$$

where $U(s)$ unimodular polynomial matrix

Definition:

$R(\sigma)\mathbf{w} = 0$ is called a minimal representation if the row degrees of $R(s)$ are minimal



$R(s)$ is **row proper**, i.e. sum of its row degrees equals its determinant

III General iterative modeling procedure

Exact modeling of data from a behavioral perspective:

Willems '86 '91

Antoulas & Willems '93

$\mathbf{w}_1, \dots, \mathbf{w}_N$ trajectories $\mathbb{Z}_- \mapsto \mathbb{F}^q$

How to construct $R(s)$ such that

$$R(\sigma)\mathbf{w} = 0$$

models $\{\mathbf{w}_1, \dots, \mathbf{w}_N\}$?

iterative procedure: (Willems '91)

Initialize $R_0(s) := I$

For $i = 1, 2, \dots, N$

- compute the **error** trajectory

$$e_i := R_{i-1}(\sigma)w_i$$

- compute a representation

$$E_i(\sigma)w = 0$$

that models e_i

- update

$$R_i(s) := E_i(s)R_{i-1}(s)$$

Then $R_i(\sigma)w = 0$ models $\{w_1, \dots, w_i\}$

Definition (Willems '86)

A model \mathcal{B} for a data set D is called the **most powerful unfalsified model (MPUM)** for D if for any other model $\tilde{\mathcal{B}}$ for D we have that

$$\mathcal{B} \subset \tilde{\mathcal{B}}$$

In the iterative procedure:

- $R_N(\sigma)\mathbf{w} = 0$ most powerful if at each step i :

$$E_i(\sigma)\mathbf{w} = 0 \text{ most powerful}$$

- $R_N(\sigma)\mathbf{w} = 0$ minimal representation if at each step i :

$E_i(s)$ chosen such that $E_i(s)R_{i-1}(s)$ still row proper

Now solving Key equation 1 =

Find minimal representation

$$C(\sigma)\mathbf{y} = P(\sigma)\mathbf{u}$$

such that corresponding \mathcal{B} contains trajectory

$$\left(\cdots, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} a_1 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} a_N \\ 0 \end{bmatrix} \right)$$

Extra requirement: $C(0) = 1$ **(ER 1)**

Let \mathcal{B}_i^* be MPUM of

$$\left(\cdots, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} a_1 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} a_i \\ 0 \end{bmatrix} \right)$$

Iteratively construct *minimal* representations for

$$\mathcal{B}_0^* \subset \mathcal{B}_1^* \subset \cdots \subset \mathcal{B}_N^* :$$

$$R_0(\sigma)\mathbf{w} = 0, R_1(\sigma)\mathbf{w} = 0, \dots, R_N(\sigma)\mathbf{w} = 0$$

via update formula

$$R_i(s) = E_i(s)R_{i-1}(s)$$

Choosing E_i 's such that **(ER 1)** holds and

$$R_N(s) = \begin{bmatrix} C(s) & P(s) \\ \star & \star \end{bmatrix}$$

= **Berlekamp-Massey algorithm**

Multivariable algorithm: Kuijper '97

Can be used for improved BCH decoding

(Kuijper '99)

Recall **bounded-distance decoding**: given a-priori specified e and syndromes a_1, \dots, a_N , find all solutions of degree e to Key equation 1

To solve this: use **parametrization**

$$\begin{aligned}
 & \begin{bmatrix} q_1(s) & q_2(s) \end{bmatrix} R_N(s) \\
 = & \begin{bmatrix} q_1(s) & q_2(s) \end{bmatrix} \begin{bmatrix} C(s) & P(s) \\ \star & \star \end{bmatrix}
 \end{aligned}$$

with

- q_1 monic of degree $e - L_1$,
- q_2 of degree $e - L_2$,

where L_i is i 'th row degree of R_N ($i = 1, 2$)

Slightly different approach:

Decoding: given received word $r(s)$, compute

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} r(\alpha) \\ r(\alpha^2) \\ \vdots \\ r(\alpha^N) \end{bmatrix}$$

and solve **Key equation 2**

$$(a_N + a_{N-1}s + \cdots + a_1s^{N-1})D(s) = H(s) \bmod s^N$$

such that $\deg H < \deg D$ and $\deg D$ is minimal

= **min. polynomial interpolation at $s = 0$**

$$\begin{aligned} D(s) &= \prod_{\text{error locations}} (s - \alpha^i) \\ &= \text{error locator polynomial} \end{aligned}$$

Now solving Key equation 2 =

Find minimal representation

$$D(\sigma)\mathbf{y} = H(\sigma)\mathbf{u}$$

such that corresponding \mathcal{B} contains trajectory

$$\left(\cdots, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} a_N \\ 1 \end{bmatrix}, \begin{bmatrix} a_{N-1} \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} a_1 \\ 0 \end{bmatrix} \right)$$

Extra requirement: $\deg H < \deg D$ (**ER 2**)

Employ again iterative modeling procedure, now choose E_i 's such that

$$R_N(s) = \begin{bmatrix} \tilde{D}(s) & \tilde{H}(s) \\ \tilde{K}(s) & \tilde{Q}(s) \end{bmatrix}$$

with

$$L_1 + L_2 = N + 1 \text{ and } L_1 \leq L_2$$

where

$$L_1 = \max \{ \deg \tilde{D}, 1 + \deg \tilde{H} \},$$

$$L_2 = \max \{ \deg \tilde{K}, 1 + \deg \tilde{Q} \}$$

Then, if errors within error-correcting capability, \tilde{D} is error-locator polynomial

= **Blackburn's algorithm** (=generalized Welch-Berlekamp algorithm, Blackburn '97)

Non-iterative solution method for Key equation 2:

Euclidean algorithm (Sugiyama et al. '75)

Recall: $R(\sigma)\mathbf{w} = 0$ with

$$R(s) = \begin{bmatrix} 1 & a_N + a_{N-1}s + \cdots + a_1s^{N-1} \\ 0 & s^N \end{bmatrix}$$

represents MPUM that contains

$$\left(\cdots, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} a_N \\ 1 \end{bmatrix}, \begin{bmatrix} a_{N-1} \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} a_1 \\ 0 \end{bmatrix} \right)$$

Euclidean algorithm achieves

$$\begin{bmatrix} q_{k-1}(s) & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} q_1(s) & 1 \\ 1 & 0 \end{bmatrix} R(s) = \begin{bmatrix} t_k(s) & r_k(s) \\ t_{k-1}(s) & r_{k-1}(s) \end{bmatrix}$$

Now stop as soon as $\deg r_k < \deg t_k$

Then $D(s) := t_k(s)$ and $H(s) := r_k(s)$ solve Key equation 2 with requirement **(ER 2)**

Recall decoding = solving **Key equation 2**

$$(a_N + a_{N-1}s + \cdots + a_1s^{N-1})D(s) = H(s) \bmod s^N$$

such that $\deg H < \deg D$ and $\deg D$ is minimal

Let $F(s) = F_0 + F_1s + \cdots + F_Ns^N$ with $F_N \neq 0$
and

$$\begin{bmatrix} \tilde{a}_1 \\ \vdots \\ \tilde{a}_N \end{bmatrix} = \begin{bmatrix} F_N & 0 & 0 \\ \vdots & \ddots & \vdots \\ F_1 & \cdots & F_N \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix}$$

Then solving Key equation 2 is equivalent to
solving **Key equation 3**

$$(\tilde{a}_N + \tilde{a}_{N-1}s + \cdots + \tilde{a}_1s^{N-1})D(s) = N(s) \bmod F(s)$$

such that $\deg N < \deg D$ and $\deg D$ is minimal

=**Goppa decoding** if $F(s)$ has no zeros in
 $1, \alpha, \dots, \alpha^{n-1}$

Then error value calculation:

$$e_j = \frac{N(X_j)}{X_j F(X_j) D'(X_j)}$$

Solution method: **Euclidean algorithm**
(Sugiyama et al. '75)

Let $F(s) = (s + 1)(s + \alpha) \cdots (s + \alpha^{N-1})$

Then Key equation 3 \leftrightarrow interpolation problem

Key equation 4:

$$N(x_i) = y_i D(x_i) \quad \text{for } x_i = \alpha^i \quad (i = 0, \dots, N-1)$$

Let $\bar{r}(s) = \bar{r}_0 + \bar{r}_1 s + \cdots + \bar{r}_{N-1} s^{N-1}$ (**remainder polynomial**) be such that syndrome

$$a_i = \bar{r}(\alpha^i) \quad i = 1, \dots, N$$

Theorem Decoding = solving Key equation 4 with $y_i = \alpha^i F'(\alpha^i) \bar{r}_i$, such that $\deg N < \deg D$ and $\deg D$ is minimal

= **minimal polynomial interpolation**

(Welch & Berlekamp '83)

Conclusion W-B decoding can be derived as a special case of Goppa decoding

Note no syndrome computation needed

Let \mathcal{B}_i^* be MPUM of $\{\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_i\}$ where

$$\mathbf{w}_i = \left(\dots, \begin{bmatrix} y_i x_i^2 \\ x_i^2 \end{bmatrix}, \begin{bmatrix} y_i x_i \\ x_i \end{bmatrix}, \begin{bmatrix} y_i \\ 1 \end{bmatrix} \right)$$

Iteratively construct *minimal* representations for

$$\mathcal{B}_0^* \subset \mathcal{B}_1^* \subset \dots \subset \mathcal{B}_{N-1}^* :$$

$$R_0(\sigma)\mathbf{w} = 0, R_1(\sigma)\mathbf{w} = 0, \dots, R_{N-1}(\sigma)\mathbf{w} = 0$$

via update formula

$$R_i(s) = E_i(s)R_{i-1}(s)$$

Choosing E_i 's such that

$$R_{N-1}(s) = \begin{bmatrix} \tilde{D}(s) & \tilde{N}(s) \\ \tilde{K}(s) & \tilde{Q}(s) \end{bmatrix}$$

with

$$L_1 + L_2 = N \text{ and } L_1 \leq L_2$$

where

$$L_1 = \max \{ \deg \tilde{D}, 1 + \deg \tilde{N} \},$$

$$L_2 = \max \{ \deg \tilde{K}, 1 + \deg \tilde{Q} \}$$

Then, if errors within error-correcting capability,
 \tilde{D} is error-locator polynomial

= Welch-Berlekamp algorithm '83

Note: order of y_i 's not important

In more detail: **Algorithm:**

$$\text{Initialize } R_0 = \begin{bmatrix} D_0 & N_0 \\ K_0 & Q_0 \end{bmatrix} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; J_0 := 0$$

Compute error trajectory

$$e_i := R_{i-1}(\sigma)w_i = \left(\dots, \begin{bmatrix} \Delta_i x_i^2 \\ \gamma_i x_i^2 \end{bmatrix}, \begin{bmatrix} \Delta_i x_i \\ \gamma_i x_i \end{bmatrix}, \begin{bmatrix} \Delta_i \\ \gamma_i \end{bmatrix} \right)$$

Update $R_i(s) := E_i(s)R_{i-1}(s)$ with

$$\begin{aligned} E_i(s) &= \begin{bmatrix} 1 & 0 \\ 0 & s - x_i \end{bmatrix} \text{ if } \Delta_i = 0; J_i := J_{i-1} + 1 \\ &= \begin{bmatrix} \gamma_i/\Delta_i & -1 \\ s - x_i & 0 \end{bmatrix} \text{ if } \Delta_i \neq 0 \text{ and } J_{i-1} = 0 \\ &= \begin{bmatrix} s - x_i & 0 \\ \gamma_i/\Delta_i & -1 \end{bmatrix} \text{ otherwise; } J_i := J_{i-1} - 1 \end{aligned}$$

Here $J = 0 \Leftrightarrow$

$$L_1 = L_2 \text{ or } (L_2 = L_1 + 1 \text{ and } \deg Q < \deg K)$$

Another way of looking at Key equation 4:

Since $r(s)$ and $\bar{r}(s)$ differ by a codeword we can just as well decode $\bar{r}(s)$ (= **re-encoding**): find $\tilde{M}(s)$ of degree $< k$ such that

$$\begin{bmatrix} \bar{r}_0 \\ \vdots \\ \bar{r}_{N-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{M}(1) \\ \vdots \\ \tilde{M}(\alpha^{N-1}) \\ \tilde{M}(\alpha^N) \\ \vdots \\ \tilde{M}(\alpha^{n-1}) \end{bmatrix}$$

for $n - e$ entries, with e minimal

This means: find $D(s)$ and $\tilde{N}(s) = \tilde{M}(s)D(s)$ with $\deg D$ minimal, such that

$$\begin{aligned} \tilde{N}(\alpha^i) &= D(\alpha^i)\bar{r}_i \quad \text{for } i = 0, \dots, N - 1, \\ \tilde{N}(\alpha^i) &= 0 \quad \text{for } i = N, \dots, n - 1 \end{aligned}$$

Let $h(s) = (s + \alpha^N) \cdots (s + \alpha^{n-1})$ and write $\tilde{N}(s) = N(s)h(s)$. Then equivalent: solving

$$h(\alpha^i)N(\alpha^i) = \bar{r}_i D(\alpha^i) \quad \text{for } i = 0, \dots, N - 1.$$

such that $\deg N < \deg D$ and $\deg D$ is minimal

= Key equation 4

The decoded message $M(s)$ is retrieved as

$$M(s) = \bar{M}(s) + \tilde{M}(s)$$

where $\tilde{M}(s)$ is the Lagrange interpolating polynomial $\alpha^i \mapsto r_i$ for $i = N, \dots, n - 1$

Note:

- D and N can have common factors
- no need to compute error locations from $D(s)$
- any k entries of r can be reencoded—e.g. most reliable ones \leftrightarrow successive erasure decoding, bounded distance soft decoding (Berlekamp '96)

Conclusions

- Analogies between W-B decoding
and Goppa decoding
= connection original \leftrightarrow textbook approach
- Common factors crucial in coding context
- Solutions with common factors appear
naturally in behavioral modeling approach
- Shown how W-B and B-M algorithm are both
special instances of general iterative
behavioral modeling procedure
- Leads to parametrization \rightarrow bounded
distance decoding

Further research

Note: Welch-Berlekamp decoding=

decoding up to $\frac{d_{min}}{2}$

Recent literature: decoding $\geq \frac{d_{min}}{2}$

(list decoding)

(e.g. Sudan '97, Guruswami & Sudan '98,
Shokrollahi '98)

Further research:

- development of efficient algorithms for list decoding a la W-B (e.g. Nielsen & Hoholdt '98)
- applications to BCH decoding