A Unified View on the Decoding of Reed-Solomon Codes

陈　立
Li Chen

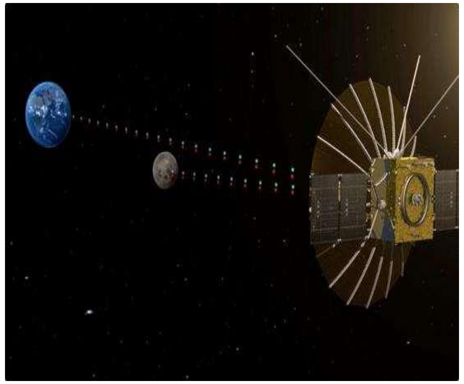
0 - 1 - 0

Outline

- Reed-Solomon Codes
- Gröbner Basis
- Key Equation Based Decoding
- Curve-fitting Based Decoding
- Foresee

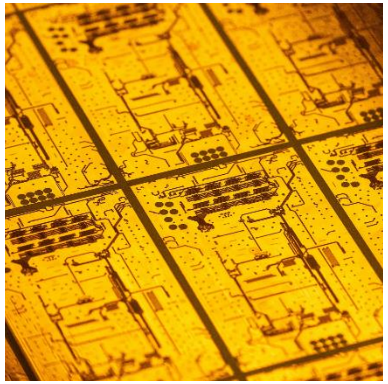
Reed-Solomon Codes

- Applications



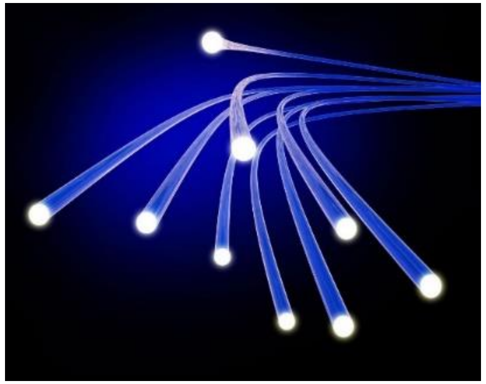
Deep space communications

Storage systems



Chip communications

Visible light communications



Optical communications

Reed-Solomon Codes

- Finite field: $\mathbf{F}_q = \{0, \sigma, \ldots, \sigma^{q-2}\}$, where σ is its primitive element // Poly. rings: $\mathbf{F}_q[x]$, $\mathbf{F}_q[x, y]$

- For an (n, k) RS code

  message $\boldsymbol{f} = (f_0, f_1, \ldots, f_{k-1})$

  or as in $f(x) = f_0 + f_1x + \cdot\cdot\cdot + f_{k-1}x^{k-1}$

  codeword $\boldsymbol{c} = (f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1}))$

  $c_0 \quad c_1 \quad \cdots \quad c_{n-1}$

  $(\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ belong to $\mathbf{F}_q\backslash\{0\}$ and called code locators

- ***Example 1*** For a (7, 3) RS code over $\mathbf{F}_8$

  Message $\boldsymbol{f}$ = (0, 2, 5), or $f(x) = 2x + 5x^2$

  Codeword $\boldsymbol{c}$ = $(f(1), f(2), f(4), f(3), f(6), f(7), f(5))$

  = (7, 6, 0, 1, 6, 1, 7)

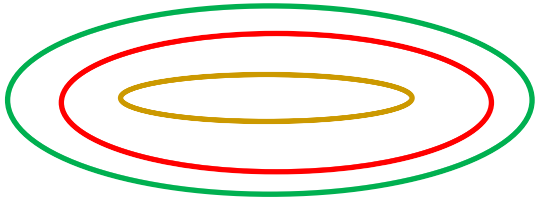
  - $\mathbf{F}_8$ is an extension field of $\mathbf{F}_2$, defined $p(x) = x^3 + x + 1$
  - In $\mathbf{F}_8$, $\{0, \sigma^0, \sigma^1, \sigma^2, \sigma^3, \sigma^4, \sigma^5, \sigma^6\} = \{0, 1, 2, 4, 3, 6, 7, 5\}$

Reed-Solomon Codes

- The channel
  - Transmit $\boldsymbol{c} = (c_0,\ c_1,\ \ldots\ \ldots,\ c_{n-1})$, in poly. $c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$
  - Channel $\boldsymbol{e} = (e_0,\ e_1,\ \ldots\ \ldots,\ e_{n-1})$, in poly. $e(x) = e_0 + e_1x + \cdots + e_{n-1}x^{n-1}$
  - Receive $\boldsymbol{\omega} = (\omega_0,\ \omega_1,\ \ldots\ \ldots,\ \omega_{n-1})$, in poly. $\omega(x) = \omega_0 + \omega_1x + \cdots + \omega_{n-1}x^{n-1}$
  - Hamming distance $d_H(\boldsymbol{c}, \boldsymbol{\omega}) = |\{j \mid e_j \neq 0\}|$ (the number of errors introduced by the channel)
- **Syndrome based decoding**
  - Error-correction capability: $t \leq \lfloor(n - k) / 2\rfloor$
- **Curve-fitting based decoding**
  - Error-correction capability: $t \geq \lfloor(n - k) / 2\rfloor$
- Probability based decoding
  - Process soft received information to improve decoding performance

Gröbner Basis

- Ideal / module (***I***) – submodule ($\boldsymbol{\Omega}_s$) – basis (<···>) – Gröbner basis (**G**)
- Given a point (α, β)



Gröbner Basis

- Submodule $\boldsymbol{\Omega}_s$:

$$\boldsymbol{\Omega}_s = \{Q(x, y) \mid Q \in \boldsymbol{I} \text{ and } \deg_y Q < s\}, \text{ and}$$

$$Q = \sum Q_{ab}x^a y^b = Q_{[0]} + Q_{[1]}y + \cdots + Q_{[s-1]}y^{s-1} \in \mathbf{F}_q[x, y], \text{ where } Q_{[.]} \in \mathbf{F}_q[x]$$

- Vector representation of Q:

$$Q = Q_{[0]} + Q_{[1]}y + \cdots + Q_{[s-1]}y^{s-1} \Leftrightarrow \underline{Q} = (Q_{[0]}, Q_{[1]}, \ldots, Q_{[s-1]}) \in \mathbf{F}_q[x]^s$$

- Ordering of Q:
  - $(1, w)$-weighted degree of $x^a y^b$: $\deg_{(1,w)} x^a y^b = a + b \cdot w$
  - $(1, w)$-revlex order: $\mathrm{ord}(x^{a_1}y^{b_1}) < \mathrm{ord}(x^{a_2}y^{b_2})$, if deg.. < deg.. or deg.. = deg.. & $b_1 < b_2$
  - $(1, w)$-weighted degree of Q: $\deg_{(1,w)} Q = \max\{\deg_{(1,w)} x^a y^b \mid Q_{ab} \neq 0\}$
  - Leading order of Q: $\mathrm{lod}(Q) = \max\{\mathrm{ord}(x^a y^b) \mid Q_{ab} \neq 0\}$
  - Leading position of $\underline{Q}$: $\mathrm{LP}(\underline{Q}) = \max\{i \mid \deg_{(1,w)} \underline{Q} = \deg_x Q_{[i]} + i \cdot w,\ Q_{[i]} \neq 0\}$

$$Q_1 < Q_2 \Leftrightarrow \mathrm{lod}(Q_1) < \mathrm{lod}(Q_2)$$
$$\underline{Q}_1 < \underline{Q}_2 \Leftrightarrow \deg_{(1,w)} \underline{Q}_1 < \deg_{(1,w)} \underline{Q}_2, \text{ or } \deg_{(1,w)} \underline{Q}_1 = \deg_{(1,w)} \underline{Q}_2 \text{ and } \mathrm{LP}(\underline{Q}_1) < \mathrm{LP}(\underline{Q}_2)$$

Gröbner Basis

- ***Example 2*** *Given $w = 2$. Let* $Q_1(x, y) = 1 + 3xy + 2x^2y^2$, $Q_2(x, y) = 3x + x^4y + 2y^2 \in \mathrm{F}_5[x, y]$, they can be presented in $\mathbf{F}_5[x]^3$ as

$$\underline{Q}_1 = (1, 3x, 2x^2),\ \underline{Q}_2 = (3x, x^4, 2)$$

For Q_1,

$$\deg_{(1,2)}Q_1 = \deg_{(1,2)}x^2y^2 = 6,\ \mathrm{lod}(Q_1) = \mathrm{ord}(x^2y^2) = 15,\ \mathrm{LP}(\underline{Q}_1) = 2$$

For Q_2,

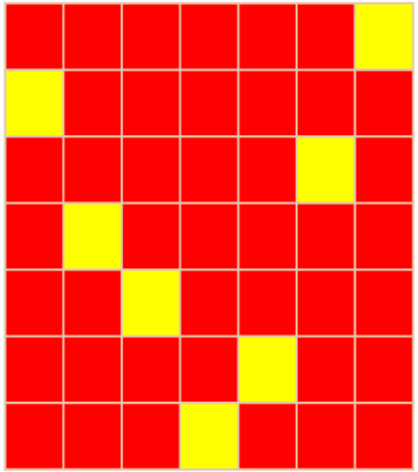
$$\deg_{(1,2)}Q_2 = \deg_{(1,\ 2)}x^4y = 6,\ \mathrm{lod}(Q_2) = \mathrm{ord}(x^4y) = 14,\ \mathrm{LP}(\underline{Q}_2) = 1$$

Therefore,

$$Q_2 < Q_1, \text{ or } \underline{Q}_2 < \underline{Q}_1$$

Gröbner Basis

- Given a basis of $\mathbf{\Omega}_s$: $\mathbf{G} = \{\underline{Q}_1, \underline{Q}_2, \ldots, \underline{Q}_s\} \subset \mathbf{\Omega}_s$, it can be represented in (light of $s = 7$)



$\underline{Q}_1$
$\underline{Q}_2$
$\underline{Q}_3$
$\underline{Q}_4$
$\underline{Q}_5$
$\underline{Q}_6$
$\underline{Q}_7$

LPs

$\mathrm{LP}(\underline{Q}_i) \neq \mathrm{LP}(\underline{Q}_j), \forall\, i \neq j$

$(\underline{Q}_1, \underline{Q}_2, \ldots, \underline{Q}_s)^T$ is in weak Popov form

- $\mathbf{G}$ is a Gröbner basis for $\mathbf{\Omega}_s$, and

$$\mathbf{\Omega}_s = \langle \underline{Q}_1, \underline{Q}_2, \ldots, \underline{Q}_s \rangle$$

Gröbner Basis

- ***Example 3*** Continue from ***Example 2***

Given $\underline{Q}_1 = (1, 3x, 2x^2)$, $\underline{Q}_2 = (3x, x^4, 2)$, $\underline{Q}_3 = (4x^6, 2x^3, 3x)$.

$\deg_{(1,2)}\underline{Q}_1 = 6$, $\mathrm{LP}(\underline{Q}_1) = 2$, $\deg_{(1,2)}\underline{Q}_2 = 6$, $\mathrm{LP}(\underline{Q}_2) = 1$, $\deg_{(1,2)}\underline{Q}_3 = 6$, $\mathrm{LP}(\underline{Q}_3) = 0$.

Therefore,

$$\begin{bmatrix} 1 & 3x & 2x^2 \\ 3x & x^4 & 2 \\ 4x^6 & 2x^3 & 3x \end{bmatrix}$$

i.e.,

$$\mathrm{LP}(\underline{Q}_1) \neq \mathrm{LP}(\underline{Q}_2) \neq \mathrm{LP}(\underline{Q}_3),$$

Therefore, $\{\underline{Q}_1, \underline{Q}_2, \underline{Q}_3\}$ is a **Gröbner basis** of submodule $< \underline{Q}_1, \underline{Q}_2, \underline{Q}_3 >$

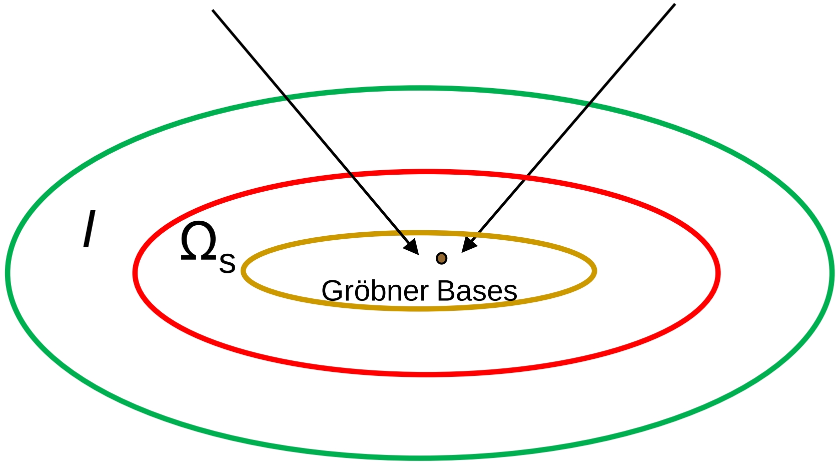
Gröbner Basis

- Decoding of Reed-Solomon codes can be seen as

  <u>Finding the minimum poly. in the ideal ***I***, or submodule $\mathbf{\Omega}_s$</u>

Syndrome based decoding:
minimum poly. $\Lambda(x)$

Curve-fitting based decoding:
minimum poly. $Q(x, y)$



Polynomial construction

$\Downarrow$

Gröbner basis

$\Uparrow$

Module basis reduction

Syndrome Based Decoding

- Syndrome: $S_1 = \omega(\sigma)$, $S_2 = \omega(\sigma^2)$, …, $S_{2t} = \omega(\sigma^{2t})$, $2t = n - k$

  Syndrome polynomial: $S(x) = S_1 + S_2x + \ldots + S_{2t}x^{2t-1}$
- Error locations: $X_1, X_2, \ldots, X_t$

  In light of a (7, 3) RS code, $(c_0, c_1, \boxed{c_2}, c_3, c_4, \boxed{c_5}, c_6)$, $X_1 = \sigma^2$, $X_2 = \sigma^5$. $\boxed{}$ erroneous sym.

- The error locator polynomial:

$$\Lambda(x) = (1 - xX_1)(1 - xX_2) \cdots (1 - xX_t) = 1 + \Lambda_1 x + \cdots + \Lambda_t x_t$$

$$\Lambda(X_1^{-1}) = \Lambda(X_2^{-1}) = \cdots = \Lambda(X_t^{-1}) = 0.$$

Peterson-Gorenstein-Zierler Alg. (1960), ***Berlekamp-Massey Alg. (1968), Euclidean Alg. (1975)***

- Determine the error locations: Chien search
- Determine the error magnitudes: Forney's algorithm
- Error-correction capability: $t \leq \lfloor (n - k) / 2 \rfloor$

From Gröbner Basis Perspective

- Error evaluator polynomial:

$$W(\mathrm{x}) = \sum_{j=1}^{t} e_{i_j} X_j \frac{\Lambda(x)}{(1-xX_j)}$$

Key equation: $S(x)\Lambda(x) = W(x) \pmod{x^{2t}}$

BM Alg.

- Known: $S(x)$
- Unknown: $\Lambda(x)$
- Module basis: $\{\ \Lambda(x)\ \}$

Euclidean Alg.

- Known: $S(x)$
- Unknown: $(\Lambda(x),\ W(x))$
- Solution module basis: $\{(1, S(x)), (0, x^{2t})\}$

Gröbner basis of the module

How to Determine the Gröbner Basis?

Polynomial Construction — Berlekamp-Massey Algorithm (1968), Kötter's Algorithm (1996)

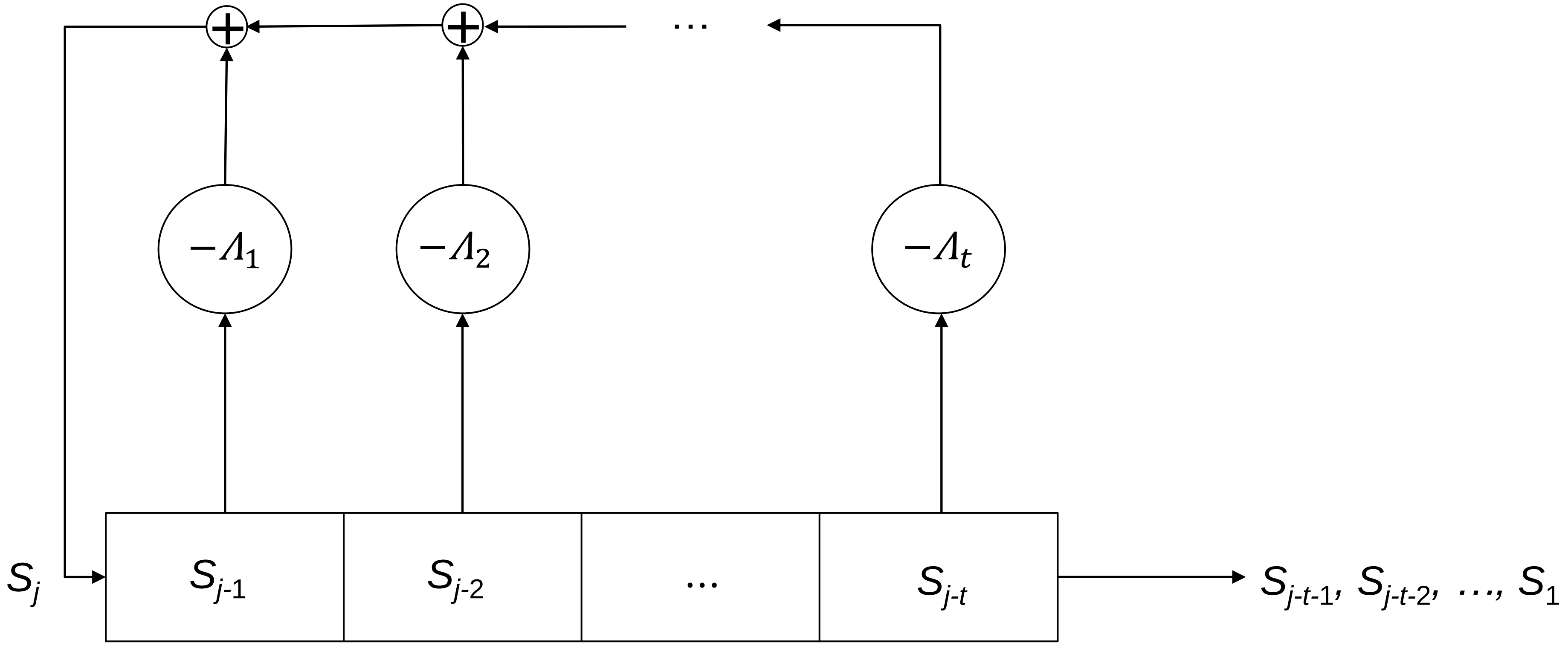
Gröbner Bases

- Syndrome Based Decoding: $\Lambda(x)$
- Curve-fitting Based Decoding: $Q(x, y)$

Module Basis Reduction — Euclidean Algorithm (1975), Lee–O'Sullivan Algorithm (2006)

Syndrome Based Decoding

- **Berlekamp-Massey (BM) algorithm**
- Error locator module: $< \Lambda(x) >$
- Goal: Find the Gröbner Basis of $< \Lambda(x) >$, i.e., the minimum polynomial $\Lambda(x)$.



Linear Feedback Shift Register

Syndrome Based Decoding

Example 4 Continue from ***Example 1***,

Given the (7, 3) RS codeword, the received word is

$$\boldsymbol{\omega} = (7, 6, \boxed{3,} 1, 6, \boxed{4,} 7)$$

With the received word, we can calculate syndromes as

$$S_1 = \omega(2) = 1, S_2 = \omega(4) = 5, S_3 = \omega(3) = 5, S_4 = \omega(6) = 1$$

Berlekamp-Massey algorithm produces

r	l	z	$\Lambda(x)$	$T(x)$	Δ
0	0	-1	1	x	1
1	1	0	$1 - x$	x	4
2	1	0	$1 - 5x$	x^2	2
3	2	1	$1 - 5x - 2x^2$	$5x - 7x^2$	7
4			$1 - 3x - x^2$	$5x^2 - 7x^3$	

The error locator polynomial is $\Lambda(x) = 1 - 3x - x^2$. In $\mathbf{F}_8$, 7 (σ^5) and 4 (σ^2) are its roots.
Therefore, ω_2 and ω_5 are erroneous.

How to Determine the Gröbner Basis?

Polynomial Construction — Berlekamp-Massey Algorithm (1968), Kötter's Algorithm (1996)

Gröbner Bases

- Key Equation Based Decoding: $\Lambda(x)$
- Interpolation Based Decoding: $Q(x, y)$

Module Basis Reduction — Euclidean Algorithm (1975), Lee–O'Sullivan Algorithm (2006)

Syndrome Based Decoding

- **The Euclidean algorithm**
- Key equation: $S(x)\Lambda(x) = W(x) (\bmod x^{2t})$
- Solution module basis: $\{(1, S(x)), (0, x^{2t})\}$
- Goal: Find the minimal $(\Lambda(x), W(x))$
- Perform row operations over $\mathbf{F}_q[x]$ on $\begin{bmatrix} 1 & S(x) \\ 0 & x^{2t} \end{bmatrix}$, until a weak Popov form is reached
- $\begin{bmatrix} 1 & S(x) \\ 0 & x^{2t} \end{bmatrix} \Longrightarrow \begin{bmatrix} \xi_{00}(x) & \xi_{01}(x) \\ \xi_{10}(x) & \xi_{11}(x) \end{bmatrix}$
- If $\deg \xi_{00}(x) + \deg \xi_{01}(x) > \deg \xi_{10}(x) + \deg \xi_{11}(x)$

$$\Lambda(x) = \xi_{10}(x), \text{ and } W(x) = \xi_{11}(x)$$

Otherwise,

$$\Lambda(x) = \xi_{00}(x), \text{ and } W(x) = \xi_{01}(x)$$

Syndrome Based Decoding

- ***Example 5*** Continue from ***Example 4***,

The syndrome polynomial is $S(x) = 1 + 5x + 5x^2 + 1x^3$

Solution module basis is

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

Reduce the above basis into a Gröbner basis as

$$\begin{bmatrix} 1 + 3x + x^2 & 1 + 6x \\ 7x + 7x^2 & 7x + x^2 \end{bmatrix}$$

and the error locator poly. is

$$\Lambda(x) = 1 + 3x + x^2$$

Syndrome Based Decoding - Chase Decoding

1924-2024 中山大學 世纪华诞 100th ANNIVERSARY SUN YAT-SEN UNIVERSITY

- **Test-Vectors Formulation**
  - Reliability matrix

$$\Pi = \begin{array}{c} \begin{array}{ccccccc} \omega_0 & \omega_1 & \cdots & \cdots & \cdots & \omega_{n-1} & \end{array} \\ \left[\begin{array}{cccccc} \pi_{0,0} & \pi_{0,1} & \cdots & \cdots & \cdots & \pi_{0,n-1} \\ \pi_{1,0} & \pi_{1,1} & \cdots & \cdots & \cdots & \pi_{1,n-1} \\ \vdots & \vdots & \ddots & & & \vdots \\ \vdots & \vdots & & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ \pi_{q-1,0} & \pi_{q-1,1} & \cdots & \cdots & \cdots & \pi_{q-1,n-1} \end{array}\right] \begin{array}{c} \sigma_0 \\ \sigma_1 \\ \vdots \\ \vdots \\ \vdots \\ \sigma_{q-1} \end{array} \end{array}$$

  - The two most likely decisions for c_j :

$$\begin{cases} \omega_j^{\mathrm{I}} = \sigma_{B_1} \\ \omega_j^{\mathrm{II}} = \sigma_{B_2} \end{cases} \Rightarrow \begin{cases} B_1 = \arg\max_i \{ \pi_{i,j} \} \\ B_2 = \arg\max_{i,\, i \neq i_j^{\mathrm{I}}} \{ \pi_{i,j} \} \end{cases}$$

  - The symbol-wise reliability metric :

$$\rho_j = \frac{\pi_{B_1,j}}{\pi_{B_2,j}} \quad \text{and } \rho_j \in [1, \infty)$$

  - Reliability sorting

$$\rho_{j_0} \geq \rho_{j_1} \geq \cdots \geq \rho_{j_{n-1}} \Rightarrow \omega_{j_0}, \omega_{j_1}, \ldots, \omega_{j_{n-1}}$$

A greater ρ_j indicates the received information of ω_j is more reliable

Syndrome Based Decoding - Chase Decoding

- **Test-Vectors Formulation**
  - Identifying η least reliable symbols, 2^η test-vectors are formed.

$$\underline{\omega}_u = (\omega_{j_0}^{(u)}, \omega_{j_1}^{(u)}, \cdots, \omega_{j_{n-\eta-1}}^{(u)}, \omega_{j_{n-\eta}}^{(u)}, \cdots, \omega_{j_{n-1}}^{(u)}),\ 1 \le u \le 2^\eta$$

$$\omega_{j_0}^{(u)} = \omega_{j_0}^{\mathrm{I}},\quad \omega_{j_1}^{(u)} = \omega_{j_1}^{\mathrm{I}},\quad \omega_{j_{n-\eta-1}}^{(u)} = \omega_{j_{n-\eta-1}}^{\mathrm{I}},\quad \omega_{j_{n-\eta}}^{(u)} = \omega_{j_{n-\eta}}^{\mathrm{I}} \text{ or } \omega_{j_{n-\eta}}^{\mathrm{II}},\quad \omega_{j_{n-1}}^{(u)} = \omega_{j_{n-1}}^{\mathrm{I}} \text{ or } \omega_{j_{n-1}}^{\mathrm{II}}$$

- **Decoding**

$$\underline{\omega}_1 \longrightarrow \boxed{\text{Syndrome based decoding}} \longrightarrow \underline{\hat{c}}_1$$

$$\underline{\omega}_2 \longrightarrow \boxed{\text{Syndrome based decoding}} \longrightarrow \underline{\hat{c}}_2$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\underline{\omega}_{2^\eta} \longrightarrow \boxed{\text{Syndrome based decoding}} \longrightarrow \underline{\hat{c}}_{2^\eta}$$

Curve-fitting Based Decoding

- **Guruswami-Sudan (GS) algorithm**
  - Code locators $(\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$
  - Transmit $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$
  - Receive $\boldsymbol{\omega} = (\omega_0, \omega_1, \ldots, \omega_{n-1})$
  - Interpolation points: $(\alpha_0, \omega_0), (\alpha_1, \omega_1), \ldots, (\alpha_{n-1}, \omega_{n-1})$
  - Interpolation multiplicity: m

- Interpolation ideal $\boldsymbol{I}_m$: a set of all $Q \in \mathbf{F}_q[x, y]$ which have a zero of multiplicity m at the above interpolation points
  - Designed decoding list size: l, $l = \deg_y Q$

- **Interpolation module $\boldsymbol{\Omega}_l$:**

$$\boldsymbol{\Omega}_l \in \boldsymbol{I}_m$$

- **Goal:** Find the Gröbner basis of $\boldsymbol{\Omega}_l$

Curve-fitting Based Decoding

- **Guruswami-Sudan (GS) algorithm**

- **Interpolation:** Generate a minimal polynomial $Q(x, y)$ in the interpolation module $\boldsymbol{\Omega}_l$

  If $\omega_j = c_j$, $(x - \alpha_j)^m \mid Q(x, f(x))$. Hence, $\prod_{j:\omega_j=c_j}(x - \alpha_j)^m \mid Q(x, f(x))$

- **Root-Finding:** Find the y-roots of $Q(x, y)$. If the decoding succeeds, the intended message $f(x)$ can be found in $Q(x, f(x)) = 0$

  If $m\,|\{j: \omega_j = c_j\}| > \deg Q(x, f(x))$, $Q(x, f(x)) = 0$

- The decoding corrects $n - |\{j: \omega_j = c_j\}| \geq \lfloor (n - k) / 2 \rfloor$ errors

Curve-fitting Based Decoding

- Why if $\omega_j = c_j$, $(x - \alpha_j)^m \mid Q(x, f(x))$?
- Point $(\alpha_j, \omega_j) = (\alpha_j, f(\alpha_j))$

How to Determine the Gröbner Basis?

Polynomial Construction — Berlekamp-Massey Algorithm (1968), Kötter's Algorithm (1996)

↓

Gröbner Bases —
- Key Equation Based Decoding: $\Lambda(x)$
- Interpolation Based Decoding: $Q(x, y)$

↑

Module Basis Reduction — Euclidean Algorithm (1975), Lee–O'Sullivan Algorithm (2006)

Curve-fitting Based Decoding

- **Kötter's interpolation**
- Interpolation points: (α_0, ω_0), (α_1, ω_1), … …, $(\alpha_{n-1}, \omega_{n-1})$
- Decoding parameter: m, l
- Determine $Q(x, y)$ in the Gröbner basis of interpolation module $\mathbf{\Omega}_l$

Initialize $\mathbf{G} = \{g^{(0)}, g^{(1)}, g^{(2)}, \ldots, g^{(l)}\} = \{1, y, y^2, \ldots, y^l\}$

Iterative polynomial construction

For each point (α_j, ω_j)

{

Test if $g^{(h)}(x, y) = \sum_{a+b\geq m} g_{ab}^{(h)} (x - \alpha_j)^a (y - \omega_j)^b \theta_{ab}(x, y)$, for all h

For those that are not

{

Modify them using bilinear modification, e.g. $g^{(h)}(x, y)\,(x - \alpha_j) \rightarrow g^{(h)}(x, y)$

}

}

$Q(x, y) = \min\{g^{(h)} \mid g^{(h)} \in \mathbf{G}\}$

Curve-fitting Based Decoding

1924-2024 中山大學 世纪华诞 100th ANNIVERSARY SUN YAT-SEN UNIVERSITY

Example 6 Given locators $\boldsymbol{\alpha} = (1, 2, 4, 3, 6, 7, 5)$, codeword $\boldsymbol{c} = (4, 6, 6, 0, 2, 4, 0)$,

Received word $\boldsymbol{\omega} = (\boxed{7}, 6, 6, 0, 2, \boxed{3}, \boxed{3})$

Interpolation points (1, 7), (2, 6), (4, 6), (3, 0), (6, 2), (7, 3), (5, 3)

Let m = 4, then l = 7 . Initialize **G** = $\{1, y, y^2, y^3, y^4, y^5, y^6, y^7\}$. Kötter's interpolation produces

j	(α, β)	Δ_0	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	i^*	**G**
0	(0, 0)	1	7	3	2	5	6	4	1	0	$\{x+1, y+7, y^2+3, y^3+2, y^4+5, y^5+6, y^6+4, y^7+1\}$
1	(0, 1)	0	1	0	3	0	5	0	4	1	$\{x+1, xy+y+7x+7, y^2+3, y^3+3y, y^4+5, y^5+5y, y^6+4, y^7+4y\}$
2	(0, 2)	0	0	1	7	0	0	5	6	2	$\{x+1, xy+y+\cdots, y^2+xy^2+\cdots, y^3+7y^2+\cdots, y^4+5, y^5+5y, y^6+5y^2+\cdots, y^7+6y^2+\cdots\}$
…	…	…	…	…	…	…	…	…	…	…	…
69	(4, 0)	5	5	1	1	1	0	0	4	5	$\{4y^7+4x^3y^6+\cdots, 5xy^7+7y^7+\cdots, 3x^3y^6+x^2y^6+\cdots, 3x^3y^6+6x^2y^6+\cdots, 3x^4y^6+6x^3y^6+\cdots, 7y^7+3x^2y^6+\cdots, 7x^2y^6+6xy^6+\cdots, 3xy^7+6x^3y^6+\cdots\}$

- $Q(x, y) = (5x + 6x^2 + \ldots + 7x^{15}) + (4 + 7x + \ldots + 6x^{13})y + \ldots + (6 + 6x + 3x^2)y^6 + 7y^7$
- Root-finding: $Q(x, y) \rightarrow \{1 + 7x + 6x^2, \underline{2 + 3x + 5x^2}, 2 + 6x + 3x^2, 3 + 5x + x^2, 7 + 2x + 6x^2\}$

How to Determine the Gröbner Basis?

Polynomial Construction — Berlekamp-Massey Algorithm (1968), Kötter's Algorithm (1996)

$\Downarrow$

Gröbner Bases

- Key Equation Based Decoding: $\Lambda(x)$
- Interpolation Based Decoding: $Q(x, y)$

$\Uparrow$

Module Basis Reduction — Euclidean Algorithm (1975), Lee–O'Sullivan Algorithm (2006)

Curve-fitting Based Decoding

- **Module basis reduction interpolation**
- Lagrange poly. w.r.t. α_j

$$L_j(x) = \prod_{j'=0, j'\neq j}^{n-1} \frac{x-\alpha_{j'}}{\alpha_j - \alpha_{j'}} \Longrightarrow \begin{array}{l} L_j\left(\alpha_{j'}\right) = 1, j' = j \\ L_j\left(\alpha_{j'}\right) = 0, j' \neq j \end{array}$$

- Seed polys.:

$$G(x) = \prod_{j=0}^{n-1}\left(x-\alpha_j\right) = x^n - 1, \quad R(x) = \sum_{j=0}^{n-1} \omega_j L_j(x)$$

Note that $R(\alpha_j) = \omega_j$ $\Longrightarrow$ $y - R(x)$ interpolates (α_j, ω_j)

- Generators of interpolation module $\mathbf{\Omega}_l$:

$$P_s(x, y) = G(x)^{m-t}(y - R(x))^t, \quad 0 \leq s \leq m$$
$$P_s(x, y) = y^{l-m}(y - R(x))^m \quad , \quad m < s \leq l$$

Note that $P_s(x, y)$ interpolate (α_0, ω_0), (α_1, ω_1) ,……, $(\alpha_{n-1}, \omega_{n-1})$ with a multiplicity of m

Curve-fitting Based Decoding

- ***Example 7*** Decoding the (7, 3) RS code using **module basis reduction interpolation** with $m = 4$ and $l = 7$

  Locators $\boldsymbol{\alpha} = (1,\ 2,\ 4,\ 3,\ 6,\ 7,\ 5)$

  Codeword $\boldsymbol{c} = (4,\ 6,\ 6,\ 0,\ 2,\ 4,\ 0)$

  Rex Word $\boldsymbol{\omega} = (\boxed{7},\ 6,\ 6,\ 0,\ 2,\ \boxed{3},\ \boxed{3})$

  Interpolation points (1, 7), (2, 6), (4, 6), (3, 0), (6, 2), (7, 3), (5, 3)

  Seed polys.

  $G(x) = (x-1)(x-2)(x-4)(x-3)(x-6)(x-7)(x-5) = x^7 - 1$

  $R(x) = 7L_0(x) + 6L_1(x) + 6L_2(x) + 0L_3(x) + 2L_4(x) + 3L_5(x) + 3L_6(x)$

  $= 5 + 7x + 5x^2 + 7x^4 + 3x^5 + 4x^6$

  Module generators

  $P_0(x, y) = G(x)^4$, $P_1(x, y) = G(x)^3(y - R(x))$, $P_2(x, y) = G(x)^2(y - R(x))^2$

  $P_3(x, y) = G(x)(y - R(x))^3$, $P_4(x, y) = (y - R(x))^4$, $P_5(x, y) = y(y - R(x))^4$

  $P_6(x, y) = y^2(y - R(x))^4$, $P_7(x, y) = y^3(y - R(x))^4$

Curve-fitting Based Decoding

- Module $\mathbf{\Omega}_7$ is

$$\begin{bmatrix} 1+x^{28} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5+7x+\ldots+4x^{27} & 1+x^7+x^{14}+x^{21} & 0 & 0 & 0 & 0 & 0 & 0 \\ 7+3x+\ldots+6x^{26} & 0 & 1+x^{14} & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & & & & \vdots \\ 0 & 0 & 0 & 3+5x^4+\ldots+2x^{24} & 0 & 0 & 0 & 1 \end{bmatrix}$$

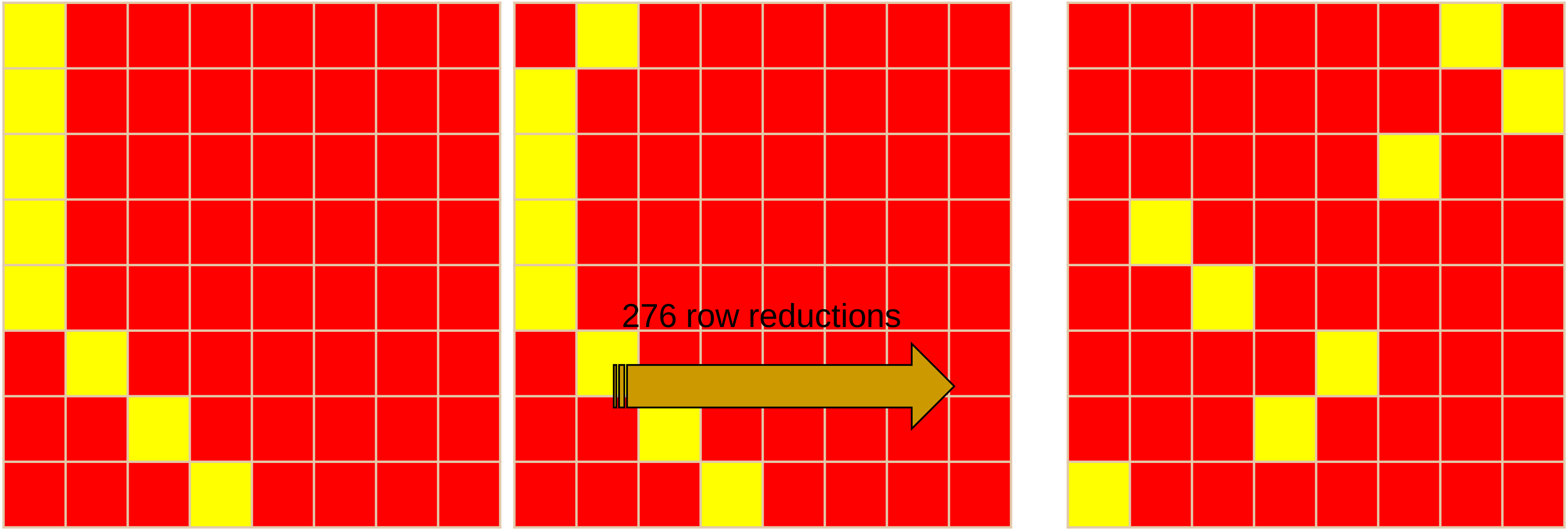
- Map $\mathbf{\Phi}_7 = \mathbf{\Omega}_7$ $\mathrm{diag}(1, x^2, \ldots, x^{14})$

$$\begin{bmatrix} (1+x^{28})1 & (0)x^2 & (0)x^4 & (0)x^6 & \ldots\ldots & (0)x^{12} & (0)x^{14} \\ (5+7x+\ldots+4x^{27})1 & (1+x^7+x^{14}+x^{21})x^2 & (0)x^4 & (0)x^6 & \ldots\ldots & (0)x^{12} & (0)x^{14} \\ (7+3x+\ldots+6x^{26})1 & (0)x^2 & (1+x^{14})x^4 & (0)x^6 & \ldots\ldots & (0)x^{12} & (0)x^{14} \\ \vdots & \vdots & \vdots & \ddots & & \vdots & \vdots \\ (0)1 & (0)x^2 & (0)x^4 & (3+5x^4+\ldots+2x^{24})x^6 & \ldots\ldots & (0)x^{12} & (1)x^{14} \end{bmatrix}$$

Curve-fitting Based Decoding

- Basis reduction: a row reduction process that brings $\mathbf{\Phi}_7$ into the weak Popov form

- A Mosaic visualization of basis reduction



- $Q(x, y) = (2x + 7x^2 + \ldots 3x^{13}) + (7 + x^2 + \ldots + 4x^{13})y + \ldots + (3 + 4x + 3x^3)y^6$
- Root-finding: $Q(x, y) \rightarrow \{1 + 7x + 6x^2, \underline{2 + 3x + 5x^2}, 2 + 6x + 3x^2, 3 + 5x + x^2\}$

Curve-fitting Based Decoding -- ReT

- **Re-encoding Transform (ReT)**
  - Transform the original interpolation problem to a reduced interpolation problem
  - Choose *k* symbols

    (For hard−decision decoding, the first *k* symbols can be chosen for simplicity)

$$h_j = \omega_j, \text{ if } j = 0,1,\ldots,k-1 \xrightarrow{\text{erasure decoding}} \underline{h} = (h_0, h_1, \ldots, h_{n-1})$$

  - Re-encoding transform

$$z_j = \begin{cases} 0, & j = 0,1,\ldots,k-1. \\ \dfrac{\omega_j - h_j}{V(\alpha_j)} = \dfrac{z_j}{V(\alpha_j)}, & j = k, k+1, \ldots, n-1. \end{cases}$$

$V(x) = \prod_{j=0,1,\ldots,k-1}(x-\alpha_j)$

  - The transformed vector

$$\underline{z} = (0, 0, \ldots, 0, z_k, \ldots, z_{n-1})$$

k points are set to zero

Their interpolation constraints are satisfied by $V(x)^m$

Only need to interpolate $n-k$ points: $(\alpha_k, z_k), (\alpha_{k+1}, z_{k+1}), \ldots, (\alpha_{n-1}, z_{n-1})$

Curve-fitting Based Decoding -- ReT

- **Module Basis Reduction with ReT**
  - Let us have a look at the module seeds
    - $G(x) = \boxed{(x-\alpha_0)\,(x-\alpha_1)\ldots(x-\alpha_{k-1})}(x-\alpha_k)\ldots(x-\alpha_{n-1})$
    - Since $z_0 = z_1 = \cdots = z_{k-1} = 0$

$$R(x) = z_k L_k + z_{k+1} L_{k+1} + \cdots + z_{n-1} L_{n-1}$$

$$L_k(x) = \frac{\boxed{(x-\alpha_0)(x-\alpha_1)\cdots(x-\alpha_{k-1})}(x-\alpha_{k+1})\cdots(x-\alpha_{n-1})}{(\alpha_k-\alpha_0)(\alpha_k-\alpha_1)\cdots(\alpha_k-\alpha_{k-1})(\alpha_k-\alpha_{k+1})\cdots(\alpha_k-\alpha_{n-1})}$$

$$L_{k+1}(x) = \frac{\boxed{(x-\alpha_0)(x-\alpha_1)\cdots(x-\alpha_{k-1})}(x-\alpha_k)\cdots(x-\alpha_{n-1})}{(\alpha_{k+1}-\alpha_0)(\alpha_{k+1}-\alpha_1)\cdots(\alpha_{k+1}-\alpha_{k-1})(\alpha_{k+1}-\alpha_k)\cdots(\alpha_{k+1}-\alpha_{n-1})}$$

$$\vdots$$

$$L_{n-1}(x) = \frac{\boxed{(x-\alpha_0)(x-\alpha_1)\cdots(x-\alpha_{k-1})}(x-\alpha_k)\cdots(x-\alpha_{n-2})}{(\alpha_{n-1}-\alpha_0)(\alpha_{n-1}-\alpha_1)\cdots(\alpha_{n-1}-\alpha_{k-1})(\alpha_{n-1}-\alpha_k)\cdots(\alpha_{n-1}-\alpha_{n-2})}$$

- $V(x) = \prod_{j=0,1,\ldots,k-1}(x-\alpha_j)$ becomes a GCD for both $G(x)$ and $R(x)$

Curve-fitting Based Decoding -- ReT

1924-2024
中山大學 世纪华诞
100th ANNIVERSARY
SUN YAT-SEN UNIVERSITY

- **Module Basis Reduction with ReT**
  - With $V(x) = \prod_{j=0,1,\ldots,k-1}(x-\alpha_j)$
  - Let

$$G^*(x) = \frac{G(x)}{V(x)} \qquad R^*(x) = \frac{R(x)}{V(x)}$$

The interpolation complexity is reduced by a factor of $\frac{k}{n}$.

  - An isomorphism of module can be created by

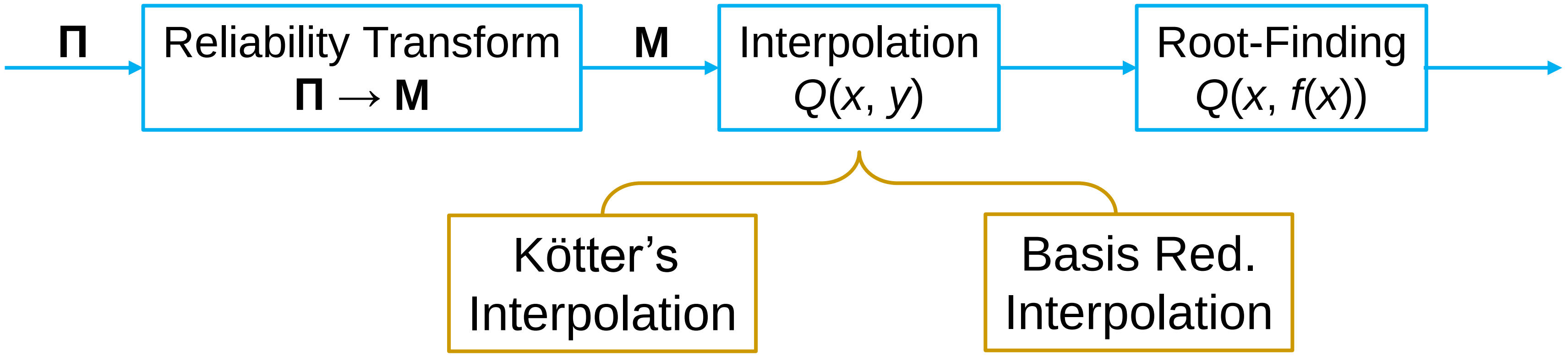
$P_s^*(x,y) = G^*(x)^{m-s}\left(y - R^*(x)\right)^s$, if $0 \le s \le m$.

$P_s^*(x,y) = (yV(x))^{s-m}\left(y - R^*(x)\right)^m$, if $m < s \le l$.

  - Basis reduction and restoration: $Q(x,y) = Q^*\left(x, \frac{y}{V(x)}\right) \cdot \psi(x) \longleftarrow \psi(x) = \prod_{j=0,1,\ldots,k-1}(x-\alpha_j)^m$

Curve-fitting Based Decoding -- ASD

- Algebraic Soft Decoding (ASD)



Curve-fitting Based Decoding -- ASD

1924-2024 中山大學世纪华诞 100th ANNIVERSARY SUN YAT-SEN UNIVERSITY

- Reliability matrix **Π** → multiplicity matrix **M** transform

Channel transition prob. $\Pr(\omega_j = \cdot \mid c_j)$

$$\mathbf{\Pi} = \begin{bmatrix} 0.02 & 0.00 & 0.01 & 0.68 & 0.03 & 0.00 & 0.01 \\ 0.02 & 0.10 & 0.02 & 0.11 & 0.05 & 0.02 & 0.08 \\ 0.01 & 0.05 & 0.01 & 0.05 & 0.71 & 0.01 & 0.31 \\ 0.00 & 0.03 & 0.11 & 0.01 & 0.09 & 0.56 & 0.42 \\ 0.12 & 0.02 & 0.02 & 0.00 & 0.00 & 0.38 & 0.13 \\ 0.03 & 0.01 & 0.00 & 0.10 & 0.10 & 0.00 & 0.01 \\ 0.00 & 0.74 & 0.83 & 0.03 & 0.00 & 0.02 & 0.01 \\ 0.80 & 0.05 & 0.00 & 0.02 & 0.02 & 0.01 & 0.03 \end{bmatrix}$$

Interpolation multiplicity for (3, 0) is 5.

Designed list size: l

Interpolation module: Ω_l

Interpolation points: Multiplicities

(1, 7): 6   (2, 6): 5   (4, 6): 6   (3, 0): 5

(6, 2): 5   (7, 3): 4   (7, 4): 2   (5, 2): 2

(5, 3): 3   (5, 4): 1

$$\mathbf{M} = \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 4 & 3 & 6 & 7 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 6 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}$$

Curve-fitting Based Decoding -- ASD

- **ASD algorithm using Kötter's interpolation (ASD-Kötter)**
- Interpolation points: $\{(\alpha_j, \sigma_i), m_{ij} \neq 0\}$
- Determine the minimum polynomial $Q(x, y)$ in $\boldsymbol{\Omega}_l$

Initialize $\mathbf{G} = \{g^{(0)}, g^{(1)}, g^{(2)}, \ldots, g^{(l)}\} = \{1, y, y^2, \ldots, y^l\}$

For each point (α_j, σ_i)

{

Test if $g^{(h)}(x, y) = \sum_{a+b\geq m_{ij}} g_{ab}^{(h)}(x-\alpha_j)^a(y-\sigma_i)^b\Theta_{ab}(x, y)$, for all h

For those that are not

{

Modify them using bilinear modification, e.g. $g^{(h)}(x, y)(x - \alpha_j) \rightarrow g^{(h)}(x, y)$

}

}

$Q = \min\{g^{(h)} \mid g^{(h)} \in \mathbf{G}\}$

Curve-fitting Based Decoding -- ASD

- **ASD with basis reduction interpolation (ASD-BR)**
- Point enumeration

$$
\mathbf{M} = \begin{array}{cc}
\begin{array}{ccccccc} 1 & 2 & 4 & 3 & 6 & 7 & 5 \end{array} & \\
\begin{bmatrix}
0 & 0 & 0 & 5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 5 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 4 & 3 \\
0 & 0 & 0 & 0 & 0 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 5 & 6 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}
\end{array}
$$

(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 2)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 4)	(5, 2)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)		(4, 6)			(7, 4)	(5, 4)

Curve-fitting Based Decoding -- ASD

- ASD-BR algorithm
- Module basis formulation

(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
$\boxed{5}$	$\boxed{4}$	$\boxed{5}$	$\boxed{4}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 2)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 4)	(5, 2)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)		(4, 6)			(7, 4)	(5, 4)

$$P_0(x, y) = (x - 1)^6 (x - 2)^5 (x - 4)^6 (x - 3)^5 (x - 6)^5 (x - 7)^4 (x - 5)^3$$

$$R_1(x) = 7L_0(x) + 6L_1(x) + 6L_2(x) + 0L_3(x) + 2L_4(x) + 3L_5(x) + 3L_6(x)$$

$$R_1(1) = 7,\ R_1(2) = 6,\ R_1(4) = 6,\ R_1(3) = 0,\ R_1(6) = 2,\ R_1(7) = 3,\ R_1(5) = 3$$

$y - R_1(x)$ interpolates the 7 points encircled by $\boxed{}$

$$P_1(x, y) = (y - R_1(x))\,(x - 1)^5 (x - 2)^4 (x - 4)^5 (x - 3)^4 (x - 6)^4 (x - 7)^3 (x - 5)^2$$

Curve-fitting Based Decoding -- ASD

- ASD-BR algorithm
- Module formulation

(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 2)

$$\boxed{4}\begin{cases}(1,7)\\(1,7)\\(1,7)\\(1,7)\end{cases}\ \boxed{3}\begin{cases}(2,6)\\(2,6)\\(2,6)\end{cases}\ \boxed{4}\begin{cases}(4,6)\\(4,6)\\(4,6)\\(4,6)\end{cases}\ \boxed{3}\begin{cases}(3,0)\\(3,0)\\(3,0)\end{cases}\ \boxed{3}\begin{cases}(6,2)\\(6,2)\\(6,2)\end{cases}\ \boxed{2}\begin{cases}(7,3)\\(7,4)\\(7,3)\\(7,4)\end{cases}\ \boxed{2}\begin{cases}(5,3)\\(5,2)\\(5,3)\\(5,4)\end{cases}$$

$$R_2(x) = 7L_0(x) + 6L_1(x) + 6L_2(x) + 0L_3(x) + 2L_4(x) + 3L_5(x) + 2L_6(x)$$

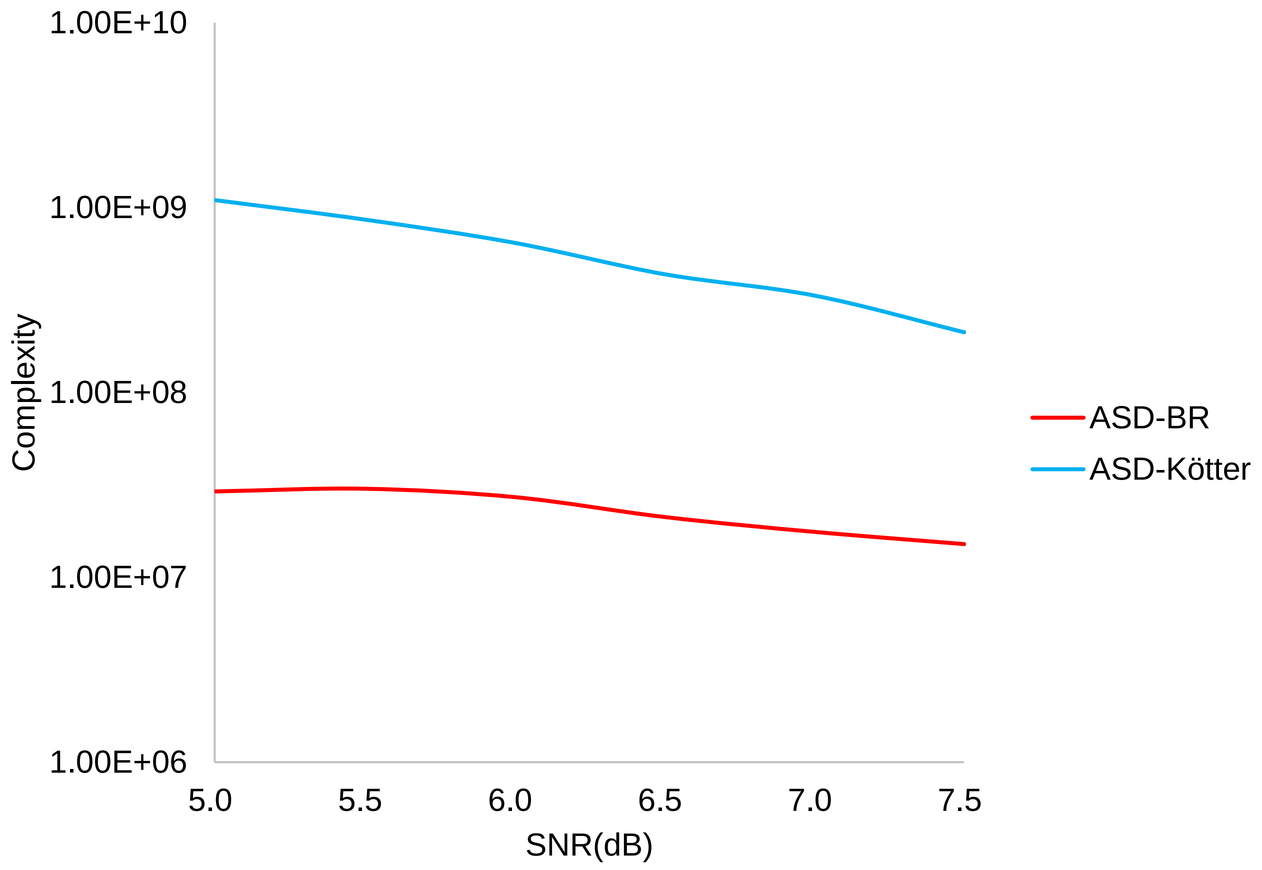
$y - R_2(x)$ interpolates the 7 points encircled by

$$P_2(x, y) = (y - R_1(x))\,(y - R_2(x))\,(x - 1)^4\,(x - 2)^3\,(x - 4)^4\,(x - 3)^3\,(x - 6)^3\,(x - 7)^2\,(x - 5)^2$$

- Generators of module $\mathbf{\Omega}_l$: $P_s(x, y) = \prod_{\varepsilon=0}^{s-1}(y - R_\varepsilon(x))\prod_{j=0}^{n-1}\left(x-\alpha_j\right)^{m_j(s)}$, $0 \le s \le l$
- Reduce the basis of $\mathbf{\Omega}_l$ into a Gröbner basis

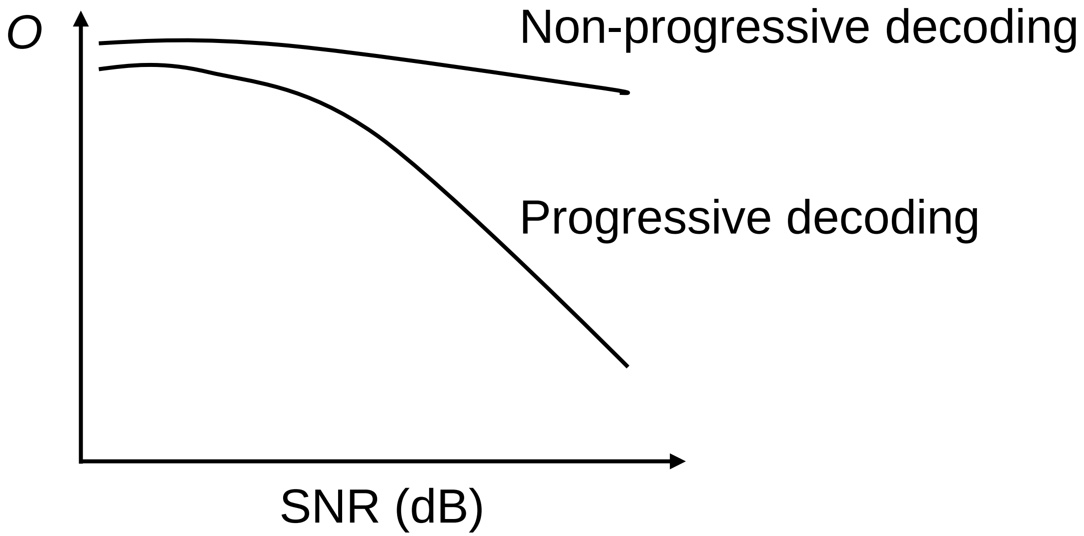
Curve-fitting Based Decoding -- ASD

- Complexity of ASD-Kötter and ASD-BR in decoding the (255, 239) RS code
- AWGN, BPSK



Curve-fitting Based Decoding -- ASD

- **Progressive Algebraic Soft Decoding (PASD)**
- Motivation: $O = \Gamma(\mathbf{\Pi})$, i.e., decoding complexity becomes a function of reliability

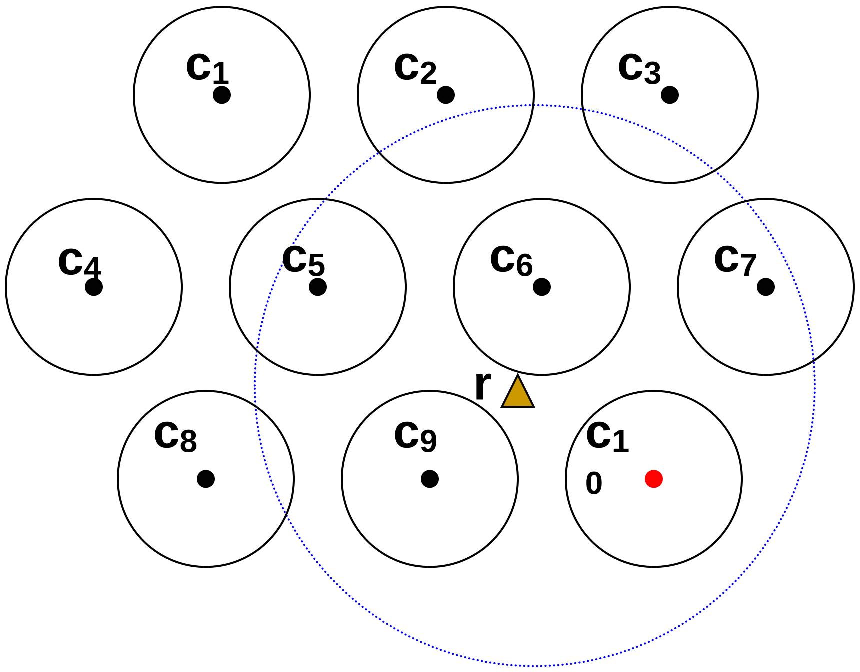


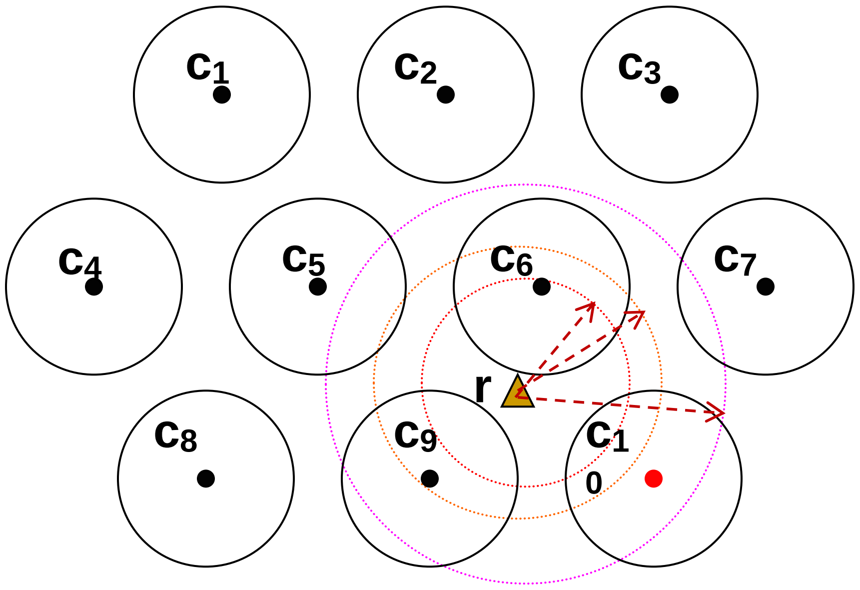
Curve-fitting Based Decoding -- PASD

- L - decoding output list

v - current output list size

ASD (l = 5) ⟶ PASD ($v = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$)





$L = \{c_5, c_6, c_7, c_9, c_{10}\}$

$v = 1 \rightarrow L = \{c_6\}$

$v = 2 \rightarrow L = \{c_6, c_9\}$

$v = 3 \rightarrow L = \{c_6, c_9, c_{10}\}$

- Enlarge v progressively and terminate once the codeword is found.

Curve-fitting Based Decoding -- PASD

- **PASD algorithm using Kötter's interpolation (PASD-Kötter)**: Progressively enlarge the polynomial group **G** → progressively enlarge $\deg_y Q_v(x, y)$
- With reliability matrix **Π** and $v = 1 \rightarrow \mathbf{M}_1$: ($\mathbf{M}_1$) defines interpolation constraints
- Initialize $\mathbf{G}_1 = \{g_0, g_1\} = \{1, y\}$

The Gröbner basis of $\mathbf{\Omega}_v$

$$\mathbf{G}_1 = \{g_0, g_1\} \xrightarrow{(\mathbf{M}_1)} \mathbf{G}_1' = \{g_0', g_1'\} \longrightarrow Q_1(x, y) \xrightarrow{\text{fac.}} f(x) \times$$

$$g_2 = y^2 \xrightarrow{(\mathbf{M}_1)} \mathbf{G}_2 = \mathbf{G}_1' \cup g_2' \xrightarrow{(\mathbf{M}_2) - (\mathbf{M}_1)} \mathbf{G}_2' = \{g_0', g_1', g_2'\} \longrightarrow Q_2(x, y) \xrightarrow{\text{fac.}} f(x) \times$$

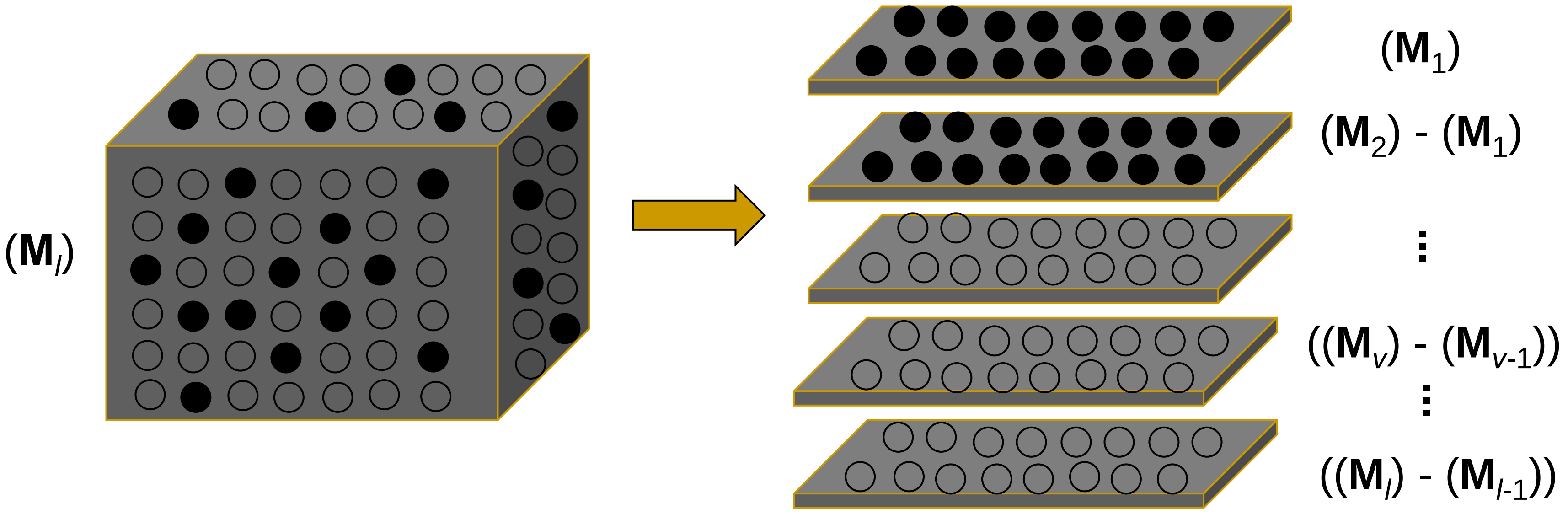
$$\vdots$$

$$g_v = y^v \xrightarrow{(\mathbf{M}_{v-1})} \mathbf{G}_v = \mathbf{G}_{v-1}' \cup g_v' \xrightarrow{(\mathbf{M}_v) - (\mathbf{M}_{v-1})} \mathbf{G}_v' = \{g_0', g_1', \ldots, g_v'\} \longrightarrow Q_v(x, y) \xrightarrow{\text{fac.}} f(x) \surd$$

The newly introduced polynomials need to be updated using the intermediate decoding information that is stored in system.

Curve-fitting Based Decoding -- PASD

- Interpretation of PASD
- $(\mathbf{M}_l) = (\mathbf{M}_1) \cup ((\mathbf{M}_2) - (\mathbf{M}_1)) \cup \cdots \cup ((\mathbf{M}_v) - (\mathbf{M}_{v-1})) \cup \cdots \cup ((\mathbf{M}_l) - (\mathbf{M}_{l-1}))$

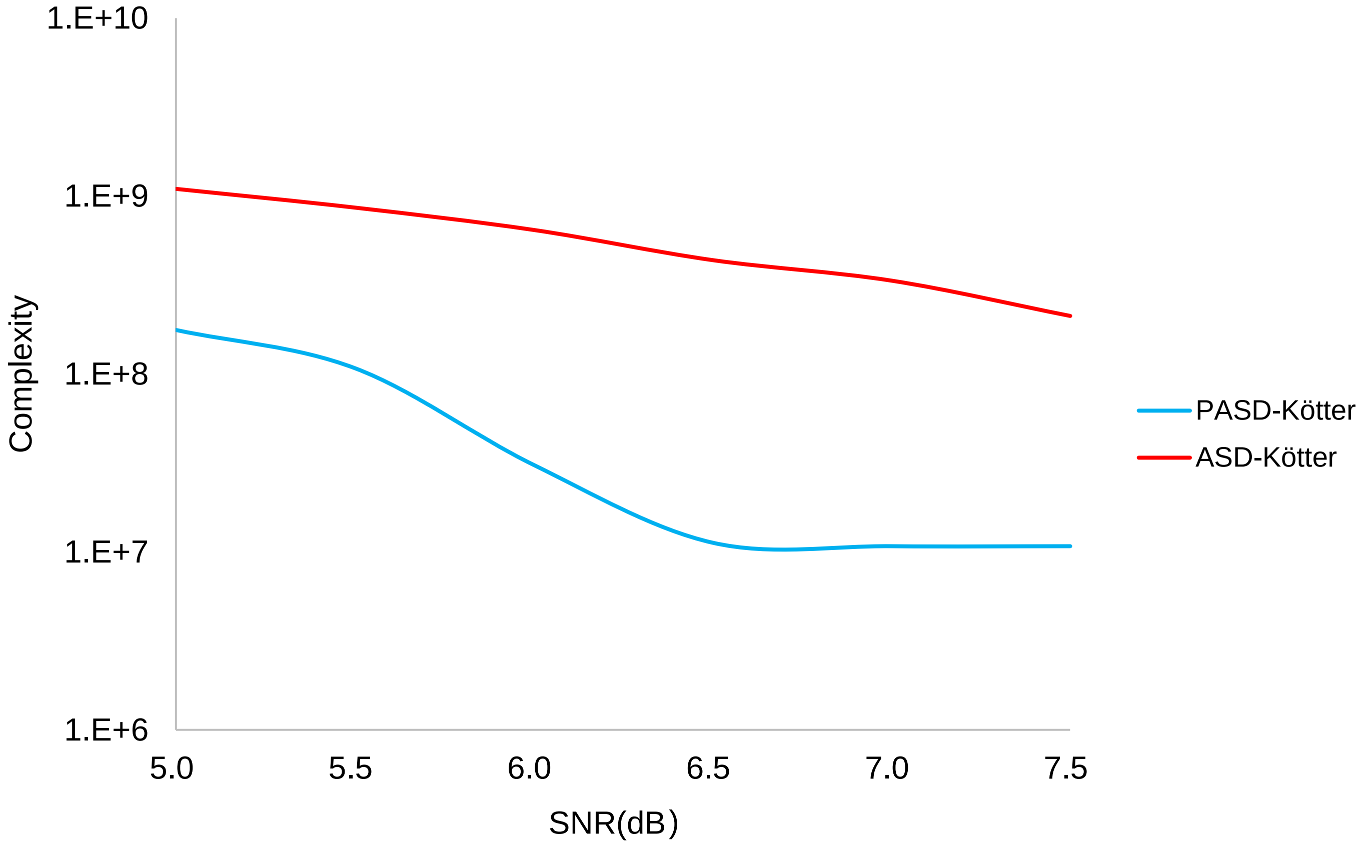


○ The constraints to be satisfied.

● The satisfied constraints.

Curve-fitting Based Decoding -- PASD

- Complexity of ASD-Kötter and PASD-Kötter in decoding the (255, 239) RS code
- AWGN, BPSK



Low decoding complexity is exchanged by system memory

Curve-fitting Based Decoding -- PASD

1924-2024 中山大學 世纪华诞 100th ANNIVERSARY SUN YAT-SEN UNIVERSITY

- **PASD algorithm using BR interpolation (PASD-BR)**

(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 2)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 4)	(5, 2)
(1, 7)	(2, 6)	(4, 6)	(3, 0)	(6, 2)	(7, 3)	(5, 3)
(1, 7)		(4, 6)			(7, 4)	(5, 4)

- Start with a decoding OLS of 1 ($\deg_y Q = 1$)

$\boxed{G_0(x) = G_1(x)\cdot T_0(x)}$

$$P_0(x, y) = \boxed{(x - 1)^6 (x - 2)^5 (x - 4)^6 (x - 3)^5 (x - 6)^5 (x - 7)^4 (x - 5)^3}$$

$$P_1(x, y) = (y - R_1(x)) \boxed{(x - 1)^5 (x - 2)^4 (x - 4)^5 (x - 3)^4 (x - 6)^4 (x - 7)^3 (x - 5)^2}$$

$\boxed{G_1(x) \text{ is the GCD.}}$

$$P_0(x, y) = G_0(x) = G_1(x)\cdot T_0(x)$$

$$P_1(x, y) = G_1(x)\cdot H_1(x, y)$$

$$B_1 = \begin{bmatrix} P^{(0)}{}_0(x) & 0 \\ P^{(0)}{}_1(x) & P^{(1)}{}_1(x) \end{bmatrix} \Longrightarrow G_1(x)\boxed{\begin{bmatrix} T_0(x) & 0 \\ H^{(0)}{}_1(x) & H^{(1)}{}_1(x) \end{bmatrix}} \rightarrow \boxed{\Xi_1}$$

- Reducing image of B_1 into the weak Popov form is equivalent to reducing image of Ξ_1 into the weak Popov form.

Curve-fitting Based Decoding -- PASD

- Increasing the decoding OLS from 1 to 2: $\deg_y Q = 2$

$$P_0(x, y) = \boxed{(x - 1)^6 (x - 2)^5 (x - 4)^6 (x - 3)^5 (x - 6)^5 (x - 7)^4 (x - 5)^3} \rightarrow \boxed{G_0(x) = T_1(x) \cdot T_0(x) \cdot G_2(x)}$$

$$P_1(x, y) = (y - R_1(x)) \boxed{(x - 1)^5 (x - 2)^4 (x - 4)^5 (x - 3)^4 (x - 6)^4 (x - 7)^3 (x - 5)^2} \rightarrow \boxed{G_1(x) = T_1(x) \cdot G_2(x)}$$

$$P_2(x, y) = (y - R_1(x)) (y - R_2(x)) \boxed{(x - 1)^4 (x - 2)^3 (x - 4)^4 (x - 3)^3 (x - 6)^3 (x - 7)^2 (x - 5)^2}$$

$\boxed{G_2(x) \text{ is the GCD.}}$

$$P_0(x, y) = G_0(x) = T_1(x) \cdot T_0(x) \cdot G_2(x)$$

$$P_1(x, y) = H_1(x, y) \cdot T_1(x) \cdot G_2(x)$$

$$P_2(x, y) = H_2(x, y) \cdot G_2(x)$$

$$B_2 = \begin{bmatrix} P^{(0)}{}_0(x) & 0 & 0 \\ P^{(0)}{}_1(x) & P^{(1)}{}_1(x) & 0 \\ P^{(0)}{}_2(x) & P^{(1)}{}_2(x) & P^{(2)}{}_2(x) \end{bmatrix} \Longrightarrow G_2(x) \boxed{\begin{bmatrix} T_1(x) \cdot T_0(x) & 0 & 0 \\ T_1(x) \cdot H^{(0)}{}_1(x) & T_1(x) \cdot H^{(0)}{}_1(x) & 0 \\ H^{(0)}{}_2(x) & H^{(1)}{}_2(x) & H^{(2)}{}_2(x) \end{bmatrix}}$$

$$\downarrow$$

$$\Xi_2 = \begin{bmatrix} T_1(x) \cdot \Xi_1 & & 0_{2\times 1} \\ H^{(0)}{}_2(x) & H^{(1)}{}_2(x) & H^{(2)}{}_2(x) \end{bmatrix}$$

- Ξ_2 is generated based on Ξ_1. $H^{(0)}{}_2(x)$, $H^{(1)}{}_2(x)$, $H^{(2)}{}_2(x)$ are generated based on enumeration lists.

Curve-fitting Based Decoding -- PASD

1924-2024 中山大學 世纪华诞 100th ANNIVERSARY SUN YAT-SEN UNIVERSITY

- Based on the above deduction

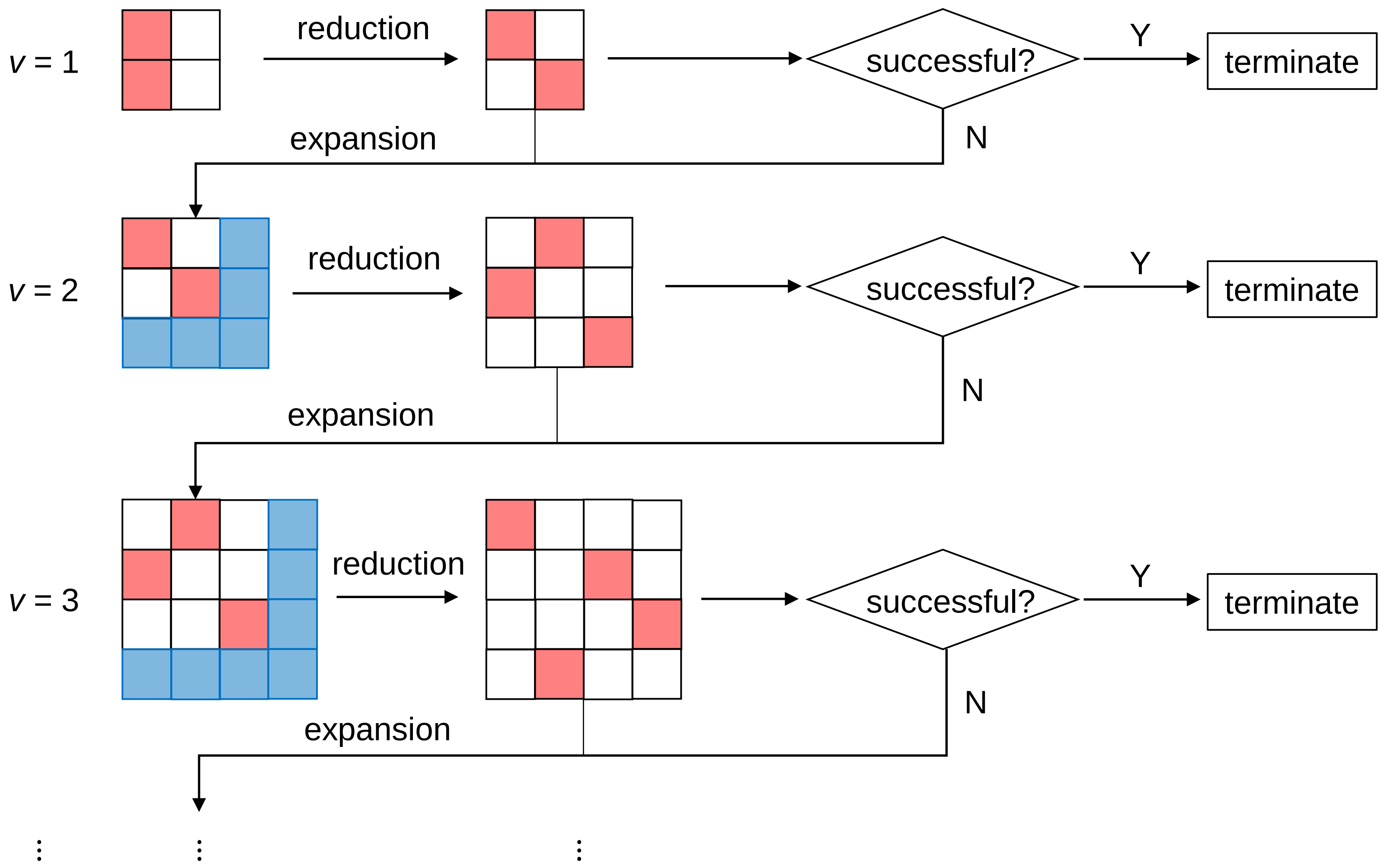
$$B_v = G_v(x)\cdot\, \Xi_v \quad\Longrightarrow\quad \Xi_v = \begin{bmatrix} T_{v\text{-}1}(x)\cdot\Xi_{v\text{-}1} & & 0_{v\times 1} \\ H^{(0)}{}_v(x) & \cdots H^{(1)}{}_v(x) & H^{(v)}{}_v(x) \end{bmatrix}$$

- Progressively enlarge Ξ_v → progressively enlarge y-degree of interpolated polynomial $Q_v(x, y)$
- The PASD-BR algorithm

The Gröbner basis of $\mathbf{\Omega}_v$

$$\begin{tikzcd}
\Xi_1 \arrow[r, "\text{Reduction}"] & \Xi_1' \arrow[r] \arrow[dl, "\text{Expansion}"] & Q_1(x, y) \arrow[r, "\text{Root-finding}"] & f(x) & \times \\
\Xi_2 \arrow[r, "\text{Reduction}"] & \Xi_2' \arrow[r] \arrow[dl, "\text{Expansion}"] & Q_2(x, y) \arrow[r, "\text{Root-finding}"] & f(x) & \times \\
\vdots & \vdots & \vdots & \vdots & \\
\Xi_v \arrow[r, "\text{Reduction}"] & \Xi_v' \arrow[r] & Q_v(x, y) \arrow[r, "\text{Root-finding}"] & f(x) & \surd
\end{tikzcd}$$

Curve-fitting Based Decoding -- PASD

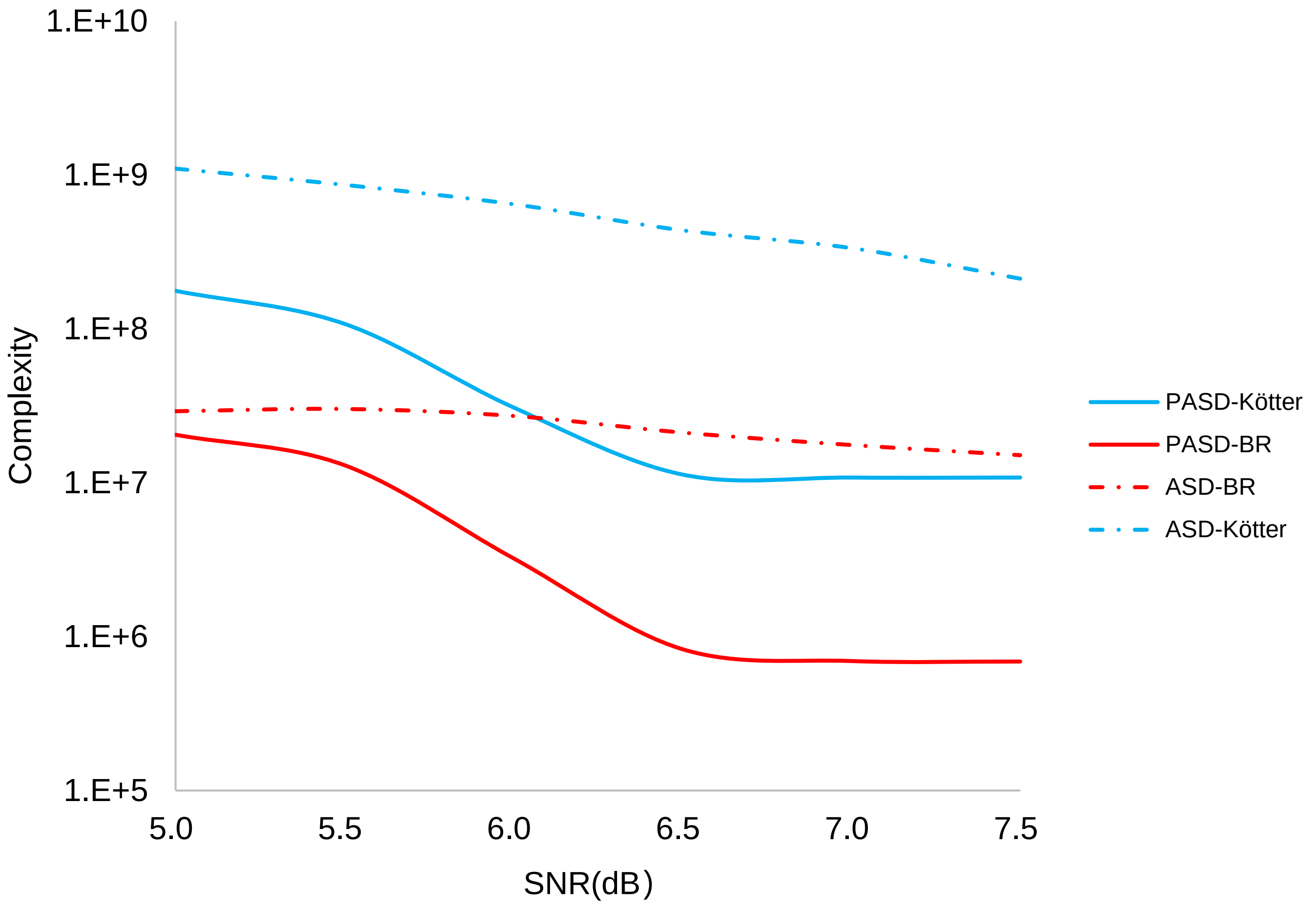


- It can remove the memory requirement of the original PASD algorithm and achieve a lower decoding complexity.

Curve-fitting Based Decoding -- PASD

1924-2024 中山大學 世纪华诞 100th ANNIVERSARY SUN YAT-SEN UNIVERSITY

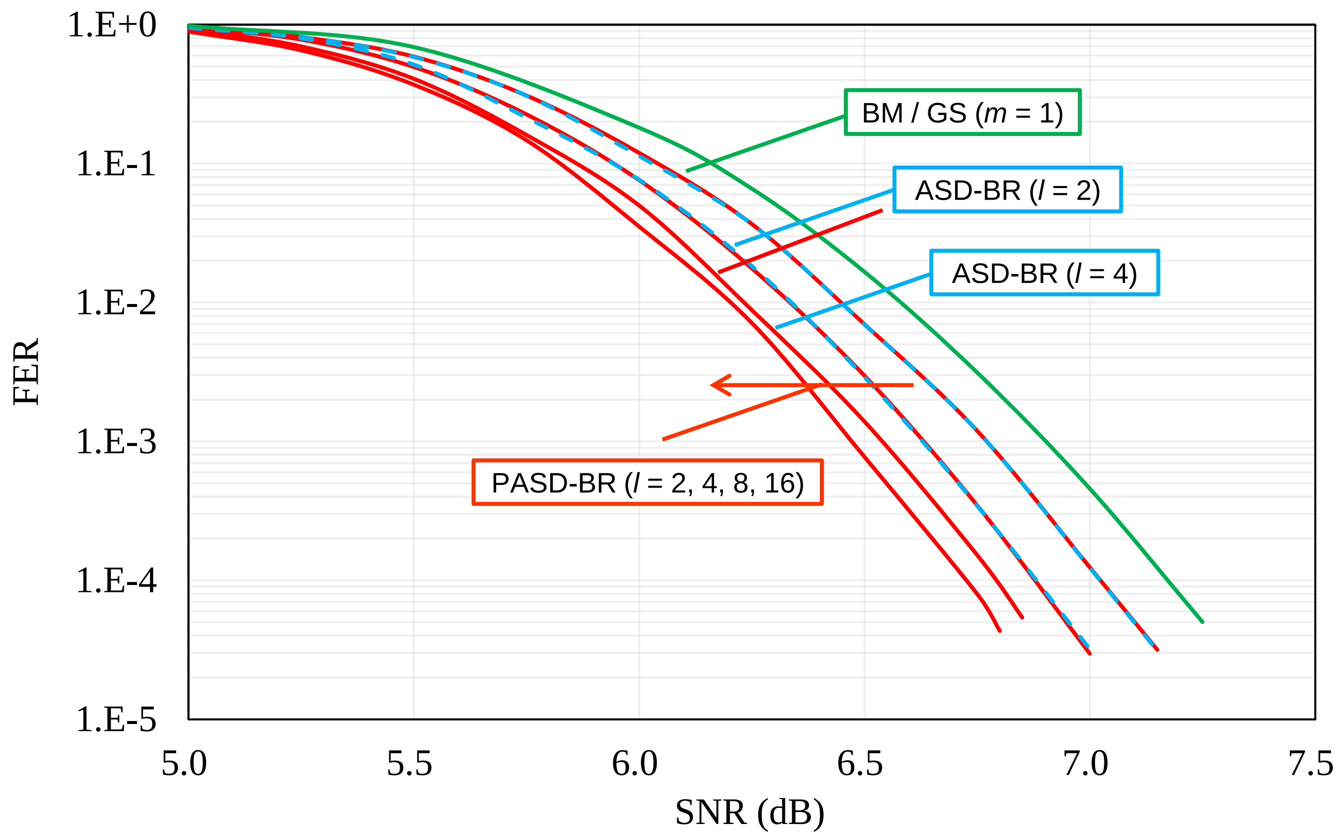
- Complexity of ASD-BR and PASD-BR in decoding the (255, 239) RS code
- AWGN, BPSK



Lower decoding complexity, memory cost is removed

Curve-fitting Based Decoding-PASD

- Performance of ASD and PASD in decoding the (255, 239) RS code
- AWGN, BPSK



Foresee

	Syndrome Based Decoding	Curve-fitting Based Decoding
Error-Correction Capability	$t = \left\lfloor \frac{n-k}{2} \right\rfloor$	$t = n - 1 - \left\lfloor \sqrt{(k-1)n} \right\rfloor$
Complexity Poly. Cons. Complexity Basis Red.	$O(t^2)$ $O(t^2)$	$\tilde{O}(n^2)$ $\delta \cdot \tilde{O}(n^2) \rightarrow \delta \cdot \tilde{O}(n)$, $\delta < 1$
Enhanced Variants	GMD, Chase decoding, Power decoding, etc.	ASD, Chase decoding, etc.
Low-complexity Variants	?	Re-encoding transform, PASD

$\tilde{O}$ denotes O without l, m and log-factors.