On the structure of the syndrome variety

Emmanuela Orsini

Department of Mathematics
University of Pisa
(orsini@posso.dm.unipi.it)

S^3CM, Soria 1-11 July

Outline

1. Introduction
   - Notation and preliminaries
   - Syndrome variety
   - A decoding algorithm
2. General error locator polynomial
   - General error locator polynomial
   - Properties of stratified ideals
   - A new syndrome variety
   - A new decoding algorithm
3. Conclusions
   - General error locator polynomial for linear codes
   - Correcting erasures via the syndrome variety
   - Multidimensional general error locator polynomials
   - Efficiency of the proposed algorithm

Outline

1. Introduction
   - Notation and preliminaries
   - Syndrome variety
   - A decoding algorithm
2. General error locator polynomial
   - General error locator polynomial
   - Properties of stratified ideals
   - A new syndrome variety
   - A new decoding algorithm
3. Conclusions
   - General error locator polynomial for linear codes
   - Correcting erasures via the syndrome variety
   - Multidimensional general error locator polynomials
   - Efficiency of the proposed algorithm

Definitions

Let C be an $[n, k, d]_q$ cyclic code, with $d = 2t + 1$ and defining set

$$S_C = \{i_1, \ldots, i_{n-k}\}.$$

Let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$.

Definitions

Let C be an $[n, k, d]_q$ cyclic code, with $d = 2t + 1$ and defining set

$$S_C = \{i_1, \ldots, i_{n-k}\}.$$

Let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$.

$$c(x) = c_0 + \cdots + c_{n-1}x^{n-1}$$ **transmitted polynomial**

$$v(x) = v_0 + \cdots + v_{n-1}x^{n-1}$$ **received polynomial**

$$e(x) = v(x) - c(x)$$ **error polynomial**

Definitions

Let C be an $[n,k,d]_q$ cyclic code, with $d = 2t + 1$ and defining set

$$S_C = \{i_1, \ldots, i_{n-k}\}.$$

Let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$.

$$c(x) = c_0 + \cdots + c_{n-1}x^{n-1} \quad \textbf{transmitted polynomial}$$

$$v(x) = v_0 + \cdots + v_{n-1}x^{n-1} \quad \textbf{received polynomial}$$

$$e(x) = v(x) - c(x) \quad \textbf{error polynomial}$$

If the weight of $\mathbf{e}$ is $\mu \le t$, then

$$\mathbf{e} = (\underbrace{0,\ldots,0}_{l_1-1}, \underset{\substack{\uparrow\\ l_1}}{e_{l_1}}, 0,\ldots,0, \underset{\substack{\uparrow\\ l_i}}{e_{l_i}}, 0,\ldots,0, \underset{\substack{\uparrow\\ l_\mu}}{e_{l_\mu}}, \underbrace{0,\ldots,0}_{n-1-l_\mu}),$$

Definitions

Let C be an $[n,k,d]_q$ cyclic code, with $d = 2t+1$ and defining set

$$S_C = \{i_1, \ldots, i_{n-k}\}.$$

Let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$.

$$c(x) = c_0 + \cdots + c_{n-1}x^{n-1} \quad \textbf{transmitted polynomial}$$

$$v(x) = v_0 + \cdots + v_{n-1}x^{n-1} \quad \textbf{received polynomial}$$

$$e(x) = v(x) - c(x) \quad \textbf{error polynomial}$$

If the weight of $\mathbf{e}$ is $\mu \leq t$, then

$$\mathbf{e} = (\underbrace{0,\ldots,0}_{l_1-1}, \underset{\substack{\uparrow\\ l_1}}{e_{l_1}}, 0, \ldots, 0, \underset{\substack{\uparrow\\ l_i}}{e_{l_i}}, 0, \ldots, 0, \underset{\substack{\uparrow\\ l_\mu}}{e_{l_\mu}}, \underbrace{0,\ldots,0}_{n-1-l_\mu}),$$

- $\mathrm{L} = \{l \mid e_l \neq 0, 0 \leq l \leq n-1\} = \{l_1, \ldots, l_\mu\}$ **set of error positions**
- $\{e_l \mid l \in \mathrm{L}\}$ **set of the error magnitudes**

Definitions

$$Hv^T = H(c^T + e^T) = Hc^T + He^T = 0 + He^T = s^T$$

$$He^T = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \cdots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \cdots & \alpha^{(n-1)i_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{i_{n-k}} & \alpha^{2i_{n-k}} & \cdots & \alpha^{(n-1)i_{n-k}}, \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-1} \end{pmatrix} = \begin{pmatrix} e(\alpha^{i_1}) \\ e(\alpha^{i_2}) \\ \vdots \\ e(\alpha^{i_{n-k}}) \end{pmatrix}$$

- $e = 0$ then $s = 0$,
- otherwise $s_j = e(\alpha^{i_j}) = \sum_{l \in \mathrm{L}} e_l \alpha^{i_j l} = \sum_{l \in \mathrm{L}} e_l (\alpha^l)^{i_j}, \quad j = 1, \ldots, n-k.$

where $\{\alpha^l \mid l \in \mathrm{L}\}$ **set of the error locations**

Definitions

$$Hv^T = H(c^T + e^T) = Hc^T + He^T = 0 + He^T = s^T$$

$$He^T = \begin{pmatrix} 1 & \alpha^{i_1} & \alpha^{2i_1} & \dots & \alpha^{(n-1)i_1} \\ 1 & \alpha^{i_2} & \alpha^{2i_2} & \dots & \alpha^{(n-1)i_2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{i_{n-k}} & \alpha^{2i_{n-k}} & \dots & \alpha^{(n-1)i_{n-k}}, \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-1} \end{pmatrix} = \begin{pmatrix} e(\alpha^{i_1}) \\ e(\alpha^{i_2}) \\ \vdots \\ e(\alpha^{i_{n-k}}) \end{pmatrix}$$

- $e = 0$ then $s = 0$,
- otherwise $s_j = e(\alpha^{i_j}) = \sum_{l \in L} e_l \alpha^{i_j l} = \sum_{l \in L} e_l (\alpha^l)^{i_j}, \quad j = 1, \dots, n-k.$

where $\{\alpha^l \mid l \in L\}$ **set of the error locations**

$$\sigma(z) = \prod_{l \in L} (1 - z\alpha^l)$$ **classical error locator polynomial**

$$L_e(z) = \prod_{l \in L} (z - \alpha^l)$$ **plain error locator polynomial**

Return

Decoding cyclic codes: the Cooper philosophy

The problem of decoding (generic) cyclic codes using Gröbner basis methods has been investigated by many authors. We recall:

- Brinton-Cooper (1990).
- Chen, Reed, Helleseth, Truong (1994).
- Loustaunau, York, (1997).
- Caboara, Mora (2002).
- Augot, Bardet, Faugere, (2003).

They work on variations of an ideal (the **syndrome ideal**) whose variety contains the error locations corresponding to any error.

Decoding cyclic codes: the Cooper philosophy

Let C be a binary BCH code with

$$S = \{2i + 1, 0 \leq i < t\}$$

and let $\overline{s} = (s_1, \ldots, s_{2t-1}) \in (\mathbb{F}_{2^m})^{2t}$ be a syndrome vector.

Decoding cyclic codes: the Cooper philosophy

Let C be a binary BCH code with

$$S = \{2i+1, 0 \leq i < t\}$$

and let $\overline{s} = (s_1, \ldots, s_{2t-1}) \in (\mathbb{F}_{2^m})^{2t}$ be a syndrome vector.

$$\mathcal{F}_C : \quad \{f_i = \sum_{j=1}^{t} z_j^{2i-1} - s_{2i-1}, \quad 1 \leq i \leq t\}$$

Decoding cyclic codes: the Cooper philosophy

Let C be a binary BCH code with

$$S = \{2i+1, 0 \leq i < t\}$$

and let $\overline{s} = (s_1, \ldots, s_{2t-1}) \in (\mathbb{F}_{2^m})^{2t}$ be a syndrome vector.

$$\mathcal{F}_C : \quad \left\{ f_i = \sum_{j=1}^{t} z_j^{2i-1} - s_{2i-1}, \quad 1 \leq i \leq t \right\}$$

The plain error locator polynomial is the monic generator $g(z_1)$ of the ideal:

$$\left\{ \sum_{i=1}^{t} g_i f_i,\ g_i \in \mathbb{F}_2(s_1, \ldots, s_{2t-1})[z_1, \ldots, z_t] \right\} \cap \mathbb{F}_2(s_1, \ldots, s_{2t-1})[z_1]$$

Decoding cyclic codes: the Cooper philosophy

The problem of decoding (generic) cyclic codes using Gröbner basis methods has been investigated by many authors. We recall:

- Brinton-Cooper (1990).
- Chen, Reed, Helleseth, Truong (1994).
- Loustaunau, York, (1997).
- Caboara, Mora (2002).
- Augot, Bardet, Faugere, (2003).

Return

They work on variations of an ideal (the **syndrome ideal**) whose variety contains the error locations corresponding to any error.

Defining the syndrome variety

Let C be an $[n,k,d]_q$ cyclic code with defining set $\{i_1,\ldots,i_{n-k}\}$.
We compute the syndrome and we obtain a system of equation

$$s_j = v(\alpha^{i_j}) = \sum_{l\in \mathrm{L}} e_l \alpha^{i_j l} = \sum_{l\in \mathrm{L}} e_l (\alpha^l)^{i_j}, \quad j = 1,\ldots,n-k$$

Return

Defining the syndrome variety

Let C be an $[n,k,d]_q$ cyclic code with defining set $\{i_1,\dots,i_{n-k}\}$.
We compute the syndrome and we obtain a system of equation

$$s_j = v(\alpha^{i_j}) = \sum_{l\in L} e_l \alpha^{i_j l} = \sum_{l\in L} e_l (\alpha^l)^{i_j}, \quad j=1,\dots,n-k$$

Return

variables	representant
$x_1,\dots,x_r$	correctable syndromes
$z_1,\dots,z_t$	error locations
$y_1,\dots,y_t$	error values

Defining the syndrome variety

Let C be an $[n, k, d]_q$ cyclic code with defining set $\{i_1, \ldots, i_{n-k}\}$.
We compute the syndrome and we obtain a system of equation

$$s_j = v(\alpha^{i_j}) = \sum_{l \in \mathrm{L}} e_l \alpha^{i_j l} = \sum_{l \in \mathrm{L}} e_l (\alpha^l)^{i_j}, \quad j = 1, \ldots, n-k$$

Return

variables	representant
$x_1, \ldots, x_r$	correctable syndromes
$z_1, \ldots, z_t$	error locations
$y_1, \ldots, y_t$	error values

Defining the syndrome variety

Let C be an $[n, k, d]_q$ cyclic code with defining set $\{i_1, \ldots, i_{n-k}\}$.
We compute the syndrome and we obtain a system of equation

$$s_j = v(\alpha^{i_j}) = \sum_{l \in L} e_l \alpha^{i_j l} = \sum_{l \in L} e_l (\alpha^l)^{i_j}, \quad j = 1, \ldots, n-k$$

Return

variables	representant
$x_1, \ldots, x_r$	correctable syndromes
$z_1, \ldots, z_t$	error locations
$y_1, \ldots, y_t$	error values

Defining the syndrome variety

Let C be an $[n,k,d]_q$ cyclic code with defining set $\{i_1,\dots,i_{n-k}\}$.
We compute the syndrome and we obtain a system of equation

$$s_j = v(\alpha^{i_j}) = \sum_{l \in L} e_l \alpha^{i_j l} = \sum_{l \in L} e_l (\alpha^l)^{i_j}, \quad j = 1,\dots,n-k$$

Return

variables	representant
$x_1,\dots,x_r$	correctable syndromes
$z_1,\dots,z_t$	error locations
$y_1,\dots,y_t$	error values

$$\sum_{l=1}^{t} y_l z_l^j - x_j, \quad j \in S_C$$

Syndrome variety

We denote by I the ideal

$$I = \mathcal{I}(\mathcal{F}) \subset \mathbb{F}_q[x_1, \ldots, x_{n-k}, z_1, \ldots, z_t, y_1, \ldots, y_t],$$

where

$$\mathcal{F} = \{f_i, h_j, \chi_i, \lambda_j, \ i \in S_C, 1 \leq j \leq t\},$$

with

$$\begin{cases} f_i := \sum_{j=1}^{t} y_j z_j^i - x_i, & i \in S_C, 1 \leq j \leq t \\ \chi_i := x_i^{q^m} - x_i, & i \in S_C \\ h_j := z_j^{n+1} - z_j, & 1 \leq j \leq t \\ \lambda_j := y_j^{q-1} - 1, & 1 \leq j \leq t \end{cases}$$

The variety $V(I)$ is the *syndrome variety*.

Gröbner basis structure

Let $\mathcal{Q} := \mathbb{F}_q[x_1, \ldots, x_{n-k}]$.
Let G be the reduced Gröbner basis of I w.r.t. the lex ordering with

$$x_1 < \cdots < x_{n-k} < z_t < \cdots < z_1 < y_1 < \cdots < y_t$$

Gröbner basis structure

Let $\mathcal{Q} := \mathbb{F}_q[x_1, \dots, x_{n-k}]$.
Let G be the reduced Gröbner basis of I w.r.t. the lex ordering with

$$x_1 < \cdots < x_{n-k} < z_t < \cdots < z_1 < y_1 < \cdots < y_t$$

Let $G = \{g_1, \dots, g_s\}$, s.t. $\mathbf{T}(g_1) < \cdots < \mathbf{T}(g_s)$.
For any $\iota \leq t$, let G_ι be $G \cap (\mathcal{Q}[z_t, \dots, z_\iota] \setminus \mathcal{Q}[z_t, \dots, z_{\iota+1}])$ and

$$\forall \ell \in \mathbb{N}, \;\; G_{\iota\ell} := \{g \in G_\iota \mid \deg_{z_\iota}(g) = \ell\},$$

Gröbner basis structure

Let $\mathcal{Q} := \mathbb{F}_q[x_1, \ldots, x_{n-k}]$.
Let G be the reduced Gröbner basis of I w.r.t. the lex ordering with

$$x_1 < \cdots < x_{n-k} < z_t < \cdots < z_1 < y_1 < \cdots < y_t$$

Let $G = \{g_1, \ldots, g_s\}$, s.t. $\mathbf{T}(g_1) < \cdots < \mathbf{T}(g_s)$.
For any $\iota \leq t$, let G_ι be $G \cap (\mathcal{Q}[z_t, \ldots, z_\iota] \setminus \mathcal{Q}[z_t, \ldots, z_{\iota+1}])$ and

$$\forall \ell \in \mathbb{N},\ \ G_{\iota\ell} := \{g \in G_\iota \mid \deg_{z_\iota}(g) = \ell\},$$

so that each G_ι can be decomposed into blocks of polynomials according to their degree with respect to the variable z_ι: $G_\iota = \sqcup_\ell G_{\iota\ell}$. If $g \in G_{\iota\ell}$:

- $g \in \mathcal{Q}[z_t, \ldots, z_{\iota+1}][z_\iota] \setminus \mathcal{Q}[z_t, \ldots, z_{\iota+1}]$;
- $\deg_{z_\iota}(g) = \ell$, i.e. $g = Lp(g) z_\iota^\ell + \ldots + Tp(g)$.

Gröbner basis structure

Let $\mathcal{Q} := \mathbb{F}_q[x_1, \ldots, x_{n-k}]$.
Let G be the reduced Gröbner basis of I w.r.t. the lex ordering with

$$x_1 < \cdots < x_{n-k} < z_t < \cdots < z_1 < y_1 < \cdots < y_t$$

Let $G = \{g_1, \ldots, g_s\}$, s.t. $\mathbf{T}(g_1) < \cdots < \mathbf{T}(g_s)$.
For any $\iota \leq t$, let G_ι be $G \cap (\mathcal{Q}[z_t, \ldots, z_\iota] \setminus \mathcal{Q}[z_t, \ldots, z_{\iota+1}])$ and

$$\forall \ell \in \mathbb{N}, \ \ G_{\iota\ell} := \{g \in G_\iota \mid \deg_{z_\iota}(g) = \ell\},$$

so that each G_ι can be decomposed into blocks of polynomials according to their degree with respect to the variable z_ι: $G_\iota = \sqcup_\ell G_{\iota\ell}$. If $g \in G_{\iota\ell}$:

- $g \in \mathcal{Q}[z_t, \ldots, z_{\iota+1}][z_\iota] \setminus \mathcal{Q}[z_t, \ldots, z_{\iota+1}]$;
- $\deg_{z_\iota}(g) = \ell$, i.e. $g = Lp(g)z_\iota^\ell + \ldots + Tp(g)$.

Moreover, we enumerate each $G_{\iota\ell}$ as

$$G_{\iota\ell} := \{g_{\iota\ell 1}, \ldots, g_{\iota\ell j_{\iota\ell}}\}, \mathbf{T}(g_{\iota\ell 1}) < \cdots < \mathbf{T}(g_{\iota\ell j_{\iota\ell}}).$$

Gröbner basis structure. (THEOREM)

With the above notation, we have:

Return

- if $\ell < \iota$ then $G_{\iota\ell} = \emptyset$;
- if $\ell > \iota$ then $\ell = n+1$, $G_{\iota\ell} = \{z_\iota^{n+1} - z_\iota\}$

For each $g \in G_{\iota\iota}$,

$$Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0 \iff g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) \neq 0.$$

Gröbner basis structure. (THEOREM)

With the above notation, we have:

Return

- if $\ell < \iota$ then $G_{\iota\ell} = \emptyset$;
- if $\ell > \iota$ then $\ell = n+1$, $G_{\iota\ell} = \{z_\iota^{n+1} - z_\iota\}$

For each $g \in G_{\iota\iota}$,

$$Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0 \iff g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) \neq 0.$$

If the error has weight μ, then, for each $g \in G_{\iota\iota}$,

1. if $\iota < \mu$ then $g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = 0$;
2. if $\iota = \mu$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$0 \neq g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) = z_\mu^\mu L_e(z_\mu);$$
3. if $\iota = \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = z_\iota \cdot (z_\iota^\mu L_e(z_\iota));$$
4. if $\iota > \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$z_\iota \cdot (z_\iota^\mu L_e(z_\iota)) \mid g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota).$$

Gröbner basis structure. (THEOREM)

Return

With the above notation, we have:

- if $\ell < \iota$ then $G_{\iota\ell} = \emptyset$;
- if $\ell > \iota$ then $\ell = n+1$, $G_{\iota\ell} = \{z_\iota^{n+1} - z_\iota\}$

For each $g \in G_{\iota\iota}$,

$$Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0 \iff g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) \neq 0.$$

If the error has weight μ, then, for each $g \in G_{\iota\iota}$,

1. if $\iota < \mu$ then $g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = 0$;
2. if $\iota = \mu$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$0 \neq g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) = z_\mu^\mu L_e(z_\mu);$$
3. if $\iota = \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = z_\iota \cdot (z_\iota^\mu L_e(z_\iota));$$
4. if $\iota > \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$z_\iota \cdot (z_\iota^\mu L_e(z_\iota)) \mid g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota).$$

Gröbner basis structure. (THEOREM)

Return

With the above notation, we have:

- if $\ell < \iota$ then $G_{\iota\ell} = \emptyset$;
- if $\ell > \iota$ then $\ell = n+1$, $G_{\iota\ell} = \{z_\iota^{n+1} - z_\iota\}$

For each $g \in G_{\iota\iota}$,

$$Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0 \iff g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) \neq 0.$$

If the error has weight μ, then, for each $g \in G_{\iota\iota}$,

1. if $\iota < \mu$ then $g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = 0$;
2. if $\iota = \mu$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$0 \neq g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) = z_\mu^\mu L_e(z_\mu);$$
3. if $\iota = \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = z_\iota \cdot (z_\iota^\mu L_e(z_\iota));$$
4. if $\iota > \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$z_\iota \cdot (z_\iota^\mu L_e(z_\iota)) \mid g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota).$$

Gröbner basis structure. (THEOREM)

Return

With the above notation, we have:

- if $\ell < \iota$ then $G_{\iota\ell} = \emptyset$;
- if $\ell > \iota$ then $\ell = n+1$, $G_{\iota\ell} = \{z_\iota^{n+1} - z_\iota\}$

For each $g \in G_{\iota\iota}$,

$$Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0 \iff g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) \neq 0.$$

If the error has weight μ, then, for each $g \in G_{\iota\iota}$,

1. if $\iota < \mu$ then $g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = 0$;
2. if $\iota = \mu$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$0 \neq g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) = z_\mu^\mu L_e(z_\mu);$$
3. if $\iota = \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = z_\iota \cdot (z_\iota^\mu L_e(z_\iota));$$
4. if $\iota > \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$z_\iota \cdot (z_\iota^\mu L_e(z_\iota)) \mid g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota).$$

Gröbner basis structure. (THEOREM)

Return

With the above notation, we have:

- if $\ell < \iota$ then $G_{\iota\ell} = \emptyset$;
- if $\ell > \iota$ then $\ell = n+1$, $G_{\iota\ell} = \{z_\iota^{n+1} - z_\iota\}$

For each $g \in G_{\iota\iota}$,

$$Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0 \iff g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) \neq 0.$$

If the error has weight μ, then, for each $g \in G_{\iota\iota}$,

1. if $\iota < \mu$ then $g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = 0$;
2. if $\iota = \mu$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$0 \neq g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\mu) = z_\mu^\mu L_e(z_\mu);$$
3. if $\iota = \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota) = z_\iota \cdot (z_\iota^\mu L_e(z_\iota));$$
4. if $\iota > \mu + 1$ and $Lp(g)(s_1, \dots, s_{n-k}, 0, \dots, 0) \neq 0$ then
$$z_\iota \cdot (z_\iota^\mu L_e(z_\iota)) \mid g(s_1, \dots, s_{n-k}, 0, \dots, 0, z_\iota).$$

Example

```
 A Computer Algebra System for Polynomial Computations   /   version 3-0-4
                                                          0<
    by: G.-M. Greuel, G. Pfister, H. Schoenemann          \   Nov 2007
FB Mathematik der Universitaet, D-67653 Kaiserslautern      \

> ring R= (2),(z_1,z_2,z_3,x_5,x_3,x_1),lp;
> ideal I=z_1+z_2+z_3+x_1, z_1^3+z_2^3+z_3^3+x_3,z_1^5+z_2^5+ z_3^5+x_5,
          z_1^16+z_1,  z_2^16+z_2,  z_3^16+z_3,  x_1^16+x_1,  x_3^16+x_3, x_5^16+x_5;
> option(redSB);
>  timer=1;
> ideal J=groebner(I);
  //used time: 0.70 sec
```

Example

```
> J;
J[1]=x_1^16+x_1
J[2]=x_3^16+x_3
J[3]=x_5*x_3^10+x_5*x_3^8*x_1^6+x_5*x_3^5+x_5*x_3^4*x_1^3+x_5*x_3^2*x_1^9+x_5*x_3*x_1^12+x_5+x_
3^10*x_1^5+x_3^8*x_1^11+x_3^5*x_1^5+x_3^4*x_1^8+x_3^2*x_1^14+x_3*x_1^2+x_1^5
J[4]=x_5^3+x_5^2*x_1^5+x_5*x_1^10+x_3^10+x_3^8*x_1^6+x_3^5+x_3^4*x_1^3+x_3^2*x_1^9+x_3*x_1^12+x
_1^15
J[5]=z_3^3*x_3+z_3^3*x_1^3+z_3^2*x_3*x_1+z_3^2*x_1^4+z_3*x_5+z_3*x_3*x_1^2+x_5*x_1+x_3^2+x_3*x_
1^3+x_1^6
J[6]=z_3^3*x_5+z_3^3*x_1^5+z_3^2*x_5*x_1+z_3^2*x_1^6+z_3*x_5^2*x_3^9+z_3*x_5^2*x_3^8*x_1^3+z_3*
x_5^2*x_3^4+z_3*x_5^2*x_3*x_1^9+z_3*x_5*x_1^2+z_3*x_3^9*x_1^10+z_3*x_3^8*x_1^13+z_3*x_3^4*x_1^1
0+z_3*x_3*x_1^4+z_3*x_1^7+x_5^2*x_3^9*x_1+x_5^2*x_3^8*x_1^4+x_5^2*x_3^4*x_1+x_5^2*x_3*x_1^10+x
_5*x_3+x_3^9*x_1^11+x_3^8*x_1^14+x_3^4*x_1^11
J[7]=z_3^16+z_3
J[8]=z_2^2*x_3+z_2^2*x_1^3+z_2*z_3*x_3+z_2*z_3*x_1^3+z_2*x_3*x_1+z_2*x_1^4+z_3^2*x_3+z_3^2*x_1^
3+z_3*x_3*x_1+z_3*x_1^4+x_5+x_3*x_1^2
J[9]=z_2^2*x_5+z_2^2*x_1^5+z_2*z_3*x_5+z_2*z_3*x_1^5+z_2*x_5*x_1+z_2*x_1^6+z_3^2*x_5+z_3^2*x_1^
5+z_3*x_5*x_1+z_3*x_1^6+x_5^2*x_3^9+x_5^2*x_3^8*x_1^3+x_5^2*x_3^4+x_5^2*x_3*x_1^9+x_5*x_1^2+x_
3^9*x_1^10+x_3^8*x_1^13+x_3^4*x_1^10+x_3*x_1^4+x_1^7
J[10]=z_2^2*z_3+z_2^2*x_1+z_2*z_3^2+z_2*x_1^2+z_3^2*x_1+z_3*x_1^2+x_3+x_1^3
J[11]=z_2^16+z_2
J[12]=z_1+z_2+z_3+x_1
```

Example

J[1]=x_1^16+x_1

J[2]=x_3^16+x_3

J[3]=x_5*x_3^10+x_5*x_3^8*x_1^6+x_5*x_3^5+x_5*x_3^4*x_1^3+x_5*x_3^2*x_1^9+x_5*x_3*x_1^12+x_5+x_3^10*x_1^5+x_3^8*x_1^11+x_3^5*x_1^5+x_3^4*x_1^8+x_3^2*x_1^14+x_3*x_1^2+x_1^5

J[4]=x_5^3+x_5^2*x_1^5+x_5*x_1^10+x_3^10+x_3^8*x_1^6+x_3^5+x_3^4*x_1^3+x_3^2*x_1^9+x_3*x_1^12+x_1^15

J[5]=z_3^3*x_3+z_3^3*x_1^3+z_3^2*x_3*x_1+z_3^2*x_1^4+z_3*x_5+z_3*x_3*x_1^2+x_5*x_1+x_3^2+x_3*x_1^3+x_1^6

J[6]=z_3^3*x_5+z_3^3*x_1^5+z_3^2*x_5*x_1+z_3^2*x_1^6+z_3*x_5^2*x_3^9+z_3*x_5^2*x_3^8*x_1^3+z_3*x_5^2*x_3^4+z_3*x_5^2*x_3*x_1^9+z_3*x_5*x_1^2+z_3*x_3^9*x_1^10+z_3*x_3^8*x_1^13+z_3*x_3^4*x_1^10+z_3*x_3*x_1^4+z_3*x_1^7+x_5^2*x_3^9*x_1+x_5^2*x_3^8*x_1^4+x_5^2*x_3^4*x_1+x_5^2*x_3*x_1^10+x_5*x_3+x_3^9*x_1^11+x_3^8*x_1^14+x_3^4*x_1^11

J[7]=z_3^16+z_3

J[8]=z_2^2*x_3+z_2^2*x_1^3+z_2*z_3*x_3+z_2*z_3*x_1^3+z_2*x_3*x_1+z_2*x_1^4+z_3^2*x_3+z_3^2*x_1^3+z_3*x_3*x_1+z_3*x_1^4+x_5+x_3*x_1^2

J[9]=z_2^2*x_5+z_2^2*x_1^5+z_2*z_3*x_5+z_2*z_3*x_1^5+z_2*x_5*x_1+z_2*x_1^6+z_3^2*x_5+z_3^2*x_1^5+z_3*x_5*x_1+z_3*x_1^6+x_5^2*x_3^9+x_5^2*x_3^8*x_1^3+x_5^2*x_3^4+x_5^2*x_3*x_1^9+x_5*x_1^2+x_3^9*x_1^10+x_3^8*x_1^13+x_3^4*x_1^10+x_3*x_1^4+x_1^7

J[10]=z_2^2*z_3+z_2^2*x_1+z_2*z_3^2+z_2*x_1^2+z_3^2*x_1+z_3*x_1^2+x_3+x_1^3

J[11]=z_2^16+z_2

J[12]=z_1+z_2+z_3+x_1

Example

$$
\begin{aligned}
g_{3,3,1} &= \mathbf{z_3^3}(x_3 + x_1^3) + z_3^2 x_3 x_1 + z_3^2 x_1^4 + z_3 x_5 + z_3 x_3 x_1^2 + x_5 x_1 + x_3^2 + x_3 x_1^3 + x_1^6 \\
g_{3,3,2} &= \mathbf{z_3^3}(x_5 + x_1^5) + z_3^2 x_5 x_1 + z_3^2 x_1^6 + z_3 x_5^2 x_3^9 + z_3 x_5^2 x_3^8 x_1^3 + z_3 x_5^2 x_3^4 + z_3 x_5^2 x_3 x_1^9 + z_3 x_5 x_1^2 + z_3 x_3^9 x_1^{10} + z_3 x_3^8 x_1^{13} + \\
&\quad z_3 x_3^4 x_1^{10} + z_3 x_3 x_1^4 + z_3 x_1^7 + x_5^2 x_3^9 x_1 + x_5^2 x_3^8 x_1^4 + x_5^2 x_3^4 x_1 + x_5^2 x_3 x_1^{10} + x_5 x_3 + x_3^9 x_1^{11} + x_3^8 x_1^{14} + x_3^4 x_1^{11} \\
g_{3,16,1} &= \mathbf{z_3^{16}} + z_3 \\
g_{2,2,1} &= \mathbf{z_2^2}(x_3 + x_1^3) + z_2 z_3 x_3 + z_2 z_3 x_1^3 + z_2 x_3 x_1 + z_2 x_1^4 + z_3^2 x_3 + z_3^2 x_1^3 + z_3 x_3 x_1 + z_3 x_1^4 + x_5 + x_3 x_1^2 \\
g_{2,2,2} &= \mathbf{z_2^2}(x_5 + x_1^5) + z_2 z_3 x_5 + z_2 z_3 x_1^5 + z_2 x_5 x_1 + z_2 x_1^6 + z_3^2 x_5 + z_3^2 x_1^5 + z_3 x_5 x_1 + z_3 x_1^6 + x_5^2 x_3^9 + x_5^2 x_3^8 x_1^3 + x_5^2 x_3^4 + \\
&\quad x_5^2 x_3 x_1^9 + x_5 x_1^2 + x_3^9 x_1^1 0 + x_3^8 x_1^1 3 + x_3^4 x_1^1 0 + x_3 x_1^4 + x_1^7 \\
g_{2,2,3} &= \mathbf{z_2^2}(z_3 + x_1) + z_2 z_3^2 + z_2 x_1^2 + z_3^2 x_1 + z_3 x_1^2 + x_3 + x_1^3 \\
g_{2,16,1} &= \mathbf{z_2^{16}} + z_2 \\
g_{1,1,1} &= \mathbf{z_1} + z_2 + z_3 + x_1
\end{aligned}
$$

$$
\begin{aligned}
G_3 &= \{G_{3,3}, G_{3,16}\} & G_{3,3} &= \{g_{3,3,1}, g_{3,3,2}\},\ G_{3,16} = \{g_{3,16,1}\} \\
G_2 &= \{G_{2,2}, G_{2,16}\} & G_{2,2} &= \{g_{2,2,1}, g_{2,2,2}, g_{2,2,3}\},\ G_{2,16} = \{g_{2,16,1}\} \\
G_1 &= \{G_{1,1}\} & G_{1,1} &= \{g_{1,1,1}\}
\end{aligned}
$$

Decoding algorithm

Input $\mu := t$, $g := 1$,
Repeat
$j := 0$
Repeat $j := j + 1$
Until $Lp(g_{\mu\mu j})(s, 0) \neq 0$ **or** $j > j_{\mu\mu}$
if $j > j_{\mu\mu}$ **then** $\mu := \mu - 1$ **else**
if $Tp(g_{\mu\mu j})(s, 0) = 0$ **do** $\mu := \mu - 1$
else $g(z) := g_{\mu\mu j}(s, 0, z)$;
Until $g \neq 1$ **or** $\mu = 0$
Output μ, $x^{\mu} g(x^{-1})$

Table: Decoding algorithm

Remark

For any correctable syndrome **s**, there are some points in $\mathcal{V}(I)$ that determine the error locations and the error values

$$(z_1, \ldots, z_\mu, \underbrace{0, \ldots, 0}_{t-\mu}, y_1, \ldots, y_\mu, \mathrm{y}_1, \ldots, \mathrm{y}_{t-\mu}),$$

where $\overline{y}_j$ is an arbitrary element in $\mathbb{F}_q$ for any j.

But in $\mathcal{V}(I)$ there are also other points that do not correspond directly to error vectors. For example, if $\mu \leq t-2$

$$(z_1, \ldots, z_\mu, z, z, \underbrace{0, \ldots, 0}_{t-(\mu+2)}, y_1, \ldots, y_\mu, \overline{y}_1, \ldots, \overline{y}_{t-\mu}),$$

with z any n-th root of unity and the other components as above.

Outline

1. Introduction
   - Notation and preliminaries
   - Syndrome variety
   - A decoding algorithm
2. General error locator polynomial
   - General error locator polynomial
   - Properties of stratified ideals
   - A new syndrome variety
   - A new decoding algorithm
3. Conclusions
   - General error locator polynomial for linear codes
   - Correcting erasures via the syndrome variety
   - Multidimensional general error locator polynomials
   - Efficiency of the proposed algorithm

Definition

Let C be an $[n,k,d]_q$ linear code and t its correction capability. Let $d \geq 3$ and $(n,q)=1$. Let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$.

Let $\mathcal{L}$ be a polynomial in $\mathbb{F}_q[S,z]$, where $S=(s_1,\dots,s_{n-k})$. Then $\mathcal{L}$ is a **general error locator polynomial** of C if

1. $\mathcal{L}(S,z)=z^t+a_{t-1}z^{t-1}+\cdots+a_0$, with $a_j \in \mathbb{F}_q[S]$, $0 \leq j \leq t-1$, that is, $\mathcal{L}$ is a monic polynomial with degree t with respect to the variable z and its coefficients are in $\mathbb{F}_q[S]$;
2. given a correctable syndrome $\mathbf{s}=(\bar{s}_1,\dots\bar{s}_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$, corresponding to a vector error of weight $\mu \leq t$ and error positions $\{l_1,\dots,l_\mu\}$, if we evaluate the S variables in $\mathbf{s}$, then the roots of $\mathcal{L}(\mathbf{s},z)$ are exactly $\{\alpha^{l_1},\dots,\alpha^{l_\mu},\underbrace{0,\dots,0}_{t-\mu}\}$.

Definition

Let C be an $[n,k,d]_q$ linear code and t its correction capability. Let $d \geq 3$ and $(n,q)=1$. Let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$.

Let $\mathcal{L}$ be a polynomial in $\mathbb{F}_q[S,z]$, where $S=(s_1,\dots,s_{n-k})$. Then $\mathcal{L}$ is a **general error locator polynomial** of C if

1. $\mathcal{L}(S,z)=z^t+a_{t-1}z^{t-1}+\cdots+a_0$, with $a_j \in \mathbb{F}_q[S]$, $0 \leq j \leq t-1$, that is, $\mathcal{L}$ is a monic polynomial with degree t with respect to the variable z and its coefficients are in $\mathbb{F}_q[S]$;
2. given a correctable syndrome $\mathbf{s}=(\bar{s}_1,\dots\bar{s}_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$, corresponding to a vector error of weight $\mu \leq t$ and error positions $\{l_1,\dots,l_\mu\}$, if we evaluate the S variables in $\mathbf{s}$, then the roots of $\mathcal{L}(\mathbf{s},z)$ are exactly $\{\alpha^{l_1},\dots,\alpha^{l_\mu},\underbrace{0,\dots,0}_{t-\mu}\}$.

Definition

Let C be an $[n, k, d]_q$ linear code and t its correction capability. Let $d \geq 3$ and $(n, q) = 1$. Let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$.

Let $\mathcal{L}$ be a polynomial in $\mathbb{F}_q[S, z]$, where $S = (s_1, \ldots, s_{n-k})$. Then $\mathcal{L}$ is a **general error locator polynomial** of C if

1. $\mathcal{L}(S, z) = z^t + a_{t-1}z^{t-1} + \cdots + a_0$, with $a_j \in \mathbb{F}_q[S]$, $0 \leq j \leq t-1$, that is, $\mathcal{L}$ is a monic polynomial with degree t with respect to the variable z and its coefficients are in $\mathbb{F}_q[S]$;
2. given a correctable syndrome $\mathbf{s} = (\bar{s}_1, \ldots \bar{s}_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$, corresponding to a vector error of weight $\mu \leq t$ and error positions $\{l_1, \ldots, l_\mu\}$, if we evaluate the S variables in $\mathbf{s}$, then the roots of $\mathcal{L}(\mathbf{s}, z)$ are exactly $\{\alpha^{l_1}, \ldots, \alpha^{l_\mu}, \underbrace{0, \ldots, 0}_{t-\mu}\}$.

Definition

Let C be an $[n,k,d]_q$ linear code and t its correction capability. Let $d \geq 3$ and $(n,q)=1$. Let α be a primitive n-th root of unity in $\mathbb{F}_{q^m}$.

Let $\mathcal{L}$ be a polynomial in $\mathbb{F}_q[S,z]$, where $S=(s_1,\ldots,s_{n-k})$. Then $\mathcal{L}$ is a **general error locator polynomial** of C if

1. $\mathcal{L}(S,z) = z^t + a_{t-1}z^{t-1} + \cdots + a_0$, with $a_j \in \mathbb{F}_q[S]$, $0 \leq j \leq t-1$, that is, $\mathcal{L}$ is a monic polynomial with degree t with respect to the variable z and its coefficients are in $\mathbb{F}_q[S]$;
2. given a correctable syndrome $\mathbf{s} = (\bar{s}_1, \ldots \bar{s}_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$, corresponding to a vector error of weight $\mu \leq t$ and error positions $\{l_1,\ldots,l_\mu\}$, if we evaluate the S variables in $\mathbf{s}$, then the roots of $\mathcal{L}(\mathbf{s},z)$ are exactly $\{\alpha^{l_1},\ldots,\alpha^{l_\mu},\underbrace{0,\ldots,0}_{t-\mu}\}$.

Stratified ideals

Let $\mathbb{K}$ be a field and $J \subset \mathbb{K}[\mathcal{S}, \mathcal{A}, \mathcal{T}]$ be a zero-dimensional radical ideal with

$$\mathcal{S} = (s_1, \ldots, s_H), \quad \mathcal{A} = (a_1, \ldots, a_L), \quad \mathcal{T} = (t_1, \ldots, t_K).$$

Stratified ideals

Let $\mathbb{K}$ be a field and $J \subset \mathbb{K}[\mathcal{S}, \mathcal{A}, \mathcal{T}]$ be a zero-dimensional radical ideal with

$$\mathcal{S} = (s_1, \ldots, s_H), \quad \mathcal{A} = (a_1, \ldots, a_L), \quad \mathcal{T} = (t_1, \ldots, t_K).$$

We fix a term ordering $>$ on $\mathbb{K}[\mathcal{S}, \mathcal{A}, \mathcal{T}]$, with $\mathcal{S} < \mathcal{A} < \mathcal{T}$, such that

$$a_1 > a_2 > \cdots > a_L$$

Stratified ideals

Let $\mathbb{K}$ be a field and $J \subset \mathbb{K}[\mathcal{S},\mathcal{A},\mathcal{T}]$ be a zero-dimensional radical ideal with

$$\mathcal{S} = (s_1,\dots,s_H), \quad \mathcal{A} = (a_1,\dots,a_L), \quad \mathcal{T} = (t_1,\dots,t_K).$$

We fix a term ordering $>$ on $\mathbb{K}[\mathcal{S},\mathcal{A},\mathcal{T}]$, with $\mathcal{S} < \mathcal{A} < \mathcal{T}$, such that

$$a_1 > a_2 > \cdots > a_L$$

We use the usual notation for the elimination ideals:

$$J_{\mathcal{S}} = J \cap \mathbb{K}[\mathcal{S}]$$

$$J_{\mathcal{S},a_L} = J \cap \mathbb{K}[\mathcal{S},a_L]$$

$$\vdots$$

$$J_{\mathcal{S},\mathcal{A}} = J_{\mathcal{S},a_L,\dots,a_1} = J \cap \mathbb{K}[\mathcal{S},a_L,\dots,a_1] = J \cap \mathbb{K}[\mathcal{S},\mathcal{A}]$$

Stratified ideals

$$\Sigma_j^L = \{(\bar{s}_1,\ldots,\bar{s}_N) \in \mathcal{V}(J_{\mathcal{S}}) \mid \exists \text{ exactly } j \text{ distinct values } \{\bar{a}_L^{(1)},\ldots,\bar{a}_L^{(j)}\}, \text{ s.t. } (\bar{s}_1,\ldots,\bar{s}_N,\bar{a}_L^{(i)}) \in \mathcal{V}(J_{\mathcal{S},a_L}), 1 \leq i \leq j\};$$

$$\Sigma_j^{h-1} = \{(\bar{s}_1,\ldots,\bar{s}_N,\bar{a}_L,\ldots,\bar{a}_h) \in \mathcal{V}(J_{\mathcal{S},a_L,\ldots,a_h}) \mid \exists \text{ exactly } j \text{ distinct values } \{\bar{a}_{h-1}^{(1)},\ldots,\bar{a}_{h-1}^{(j)}\}, \text{s.t. } (\bar{s}_1,\ldots,\bar{s}_N,\bar{a}_L,\ldots,\bar{a}_h,\ \bar{a}_{h-1}^{(i)}) \in \mathcal{V}(J_{\mathcal{S},a_L,\ldots,a_{h-1}}), 1 \leq i \leq j\}.$$

Stratified ideals

$$\Sigma_j^L = \{(\bar{s}_1,\dots,\bar{s}_N) \in \mathcal{V}(J_{\mathcal{S}}) \mid \exists \text{ exactly } j \text{ distinct values } \{\bar{a}_L^{(1)},\dots,\bar{a}_L^{(j)}\}, \\ \text{s.t. } (\bar{s}_1,\dots,\bar{s}_N,\bar{a}_L^{(i)}) \in \mathcal{V}(J_{\mathcal{S},a_L}), 1 \le i \le j\};$$

$$\Sigma_j^{h-1} = \{(\bar{s}_1,\dots,\bar{s}_N,\bar{a}_L,\dots,\bar{a}_h) \in \mathcal{V}(J_{\mathcal{S},a_L,\dots,a_h}) \mid \exists \text{ exactly } j \text{ distinct values} \\ \{\bar{a}_{h-1}^{(1)},\dots,\bar{a}_{h-1}^{(j)}\}, \text{s.t. } (\bar{s}_1,\dots,\bar{s}_N,\bar{a}_L,\dots,\bar{a}_h,\ \bar{a}_{h-1}^{(i)}) \in \mathcal{V}(J_{\mathcal{S},a_L,\dots,a_{h-1}}), \\ 1 \le i \le j\}.$$

Then it holds:

- $\mathcal{V}(J_{\mathcal{S}}) = \sqcup_{j=1}^{\lambda(L)} \Sigma_j^L$
- $\mathcal{V}(J_{\mathcal{S},a_L,\dots,a_h}) = \sqcup_{j=1}^{\lambda(h-1)} \Sigma_j^{h-1}, \quad 2 \le h \le L.$

For any arbitrary zero-dimensional ideal J nothing can be said about $\lambda(h)$, except that $\lambda(h) \ge 1$ for any $2 \le h \le L$.

Stratified ideals

We say that J is **stratified** w.r.t. the $\mathcal{A}$ variable if:

1. $\lambda(h) = h$, $1 \leq h \leq L$, (the number of distinct extensions is at most h for any point in $\mathcal{V}(J_{\mathcal{S},\mathsf{a}_L,\dots,\mathsf{a}_h})$) and
2. $\sum_j^h \neq \emptyset$, $1 \leq h \leq L$, $1 \leq j \leq h$ (there is at least a point with one extensions, . . . , up to $\lambda(h) = h$).

The definition of stratified ideals depends on the choice of the $\mathcal{A}$ variables.

Stratified ideals. Example

Let $\mathcal{S} = \{s_1\}$, $\mathcal{A} = \{a_1, a_2, a_3\}$ $(L = 3)$ and $\mathcal{T} = \{t_1\}$ s.t. $a_1 > a_2 > a_3$.

Stratified ideals. Example

Let $\mathcal{S} = \{s_1\}$, $\mathcal{A} = \{a_1, a_2, a_3\}$ ($L = 3$) and $\mathcal{T} = \{t_1\}$ s.t. $a_1 > a_2 > a_3$.
Let $J = \mathcal{I}(Z) \subset \mathbb{C}[s_1, a_3, a_2, a_1, t_1]$ with $Z = \{(1,2,1,0,0), (1,2,2,0,0),$ $(1,4,0,0,0), (1,6,0,0,0), (2,5,0,0,0), (3,1,0,0,0), (3,3,0,0,0), (5,2,0,0,0)\}$.
Then:

$$\begin{aligned}
\mathcal{V}(J_{\mathcal{S}}) &= \{1, 2, 3, 5\} \\
\mathcal{V}(J_{\mathcal{S},a_3}) &= \{(1,2), (1,4), (1,6), (2,5), (3,1), (3,3), (5,2)\} \\
\mathcal{V}(J_{\mathcal{S},a_3,a_2}) &= \{(1,2,1), (1,2,2)(1,4,0), (1,6,0), (2,5,0), (3,1,0), (3,3,0), (5,2,0)\} \\
\mathcal{V}(J_{\mathcal{S},a_3,a_2,a_1}) &= \{(1,2,1,0), (1,2,2,0)(1,4,0,0), (1,6,0,0), (2,5,0,0), (3,1,0,0), (3,3,0,0), (5,2,0,0)\}
\end{aligned}$$

Stratified ideals. Example

Let $\mathcal{S} = \{s_1\}$, $\mathcal{A} = \{a_1, a_2, a_3\}$ $(L = 3)$ and $\mathcal{T} = \{t_1\}$ s.t. $a_1 > a_2 > a_3$.
Let $J = \mathcal{I}(Z) \subset \mathbb{C}[s_1, a_3, a_2, a_1, t_1]$ with $Z = \{(1,2,1,0,0), (1,2,2,0,0), (1,4,0,0,0), (1,6,0,0,0), (2,5,0,0,0), (3,1,0,0,0), (3,3,0,0,0), (5,2,0,0,0)\}$.
Then:

$$\begin{aligned}
\mathcal{V}(J_{\mathcal{S}}) &= \{1, 2, 3, 5\} \\
\mathcal{V}(J_{\mathcal{S},a_3}) &= \{(1,2), (1,4), (1,6), (2,5), (3,1), (3,3), (5,2)\} \\
\mathcal{V}(J_{\mathcal{S},a_3,a_2}) &= \{(1,2,1), (1,2,2)(1,4,0), (1,6,0), (2,5,0), (3,1,0), (3,3,0), (5,2,0)\} \\
\mathcal{V}(J_{\mathcal{S},a_3,a_2,a_1}) &= \{(1,2,1,0), (1,2,2,0)(1,4,0,0), (1,6,0,0), (2,5,0,0), (3,1,0,0), (3,3,0,0), (5,2,0,0)\}
\end{aligned}$$

Let us consider the projection $\pi : \mathcal{V}(J_{\mathcal{S},a_3}) \to \mathcal{V}(J_{\mathcal{S}})$. Then:

$$|\pi^{-1}(\{5\})| = 1,\ |\pi^{-1}(\{2\})| = 1,\ |\pi^{-1}(\{3\})| = 2,\ |\pi^{-1}(\{1\})| = 3$$

so $\quad \sum_1^3 = \{2, 5\}$, $\sum_2^3 = \{3\}$, $\sum_3^3 = \{1\}$ and $\sum_i^3 = \emptyset$, $i > 3$.

Stratified ideals. Example

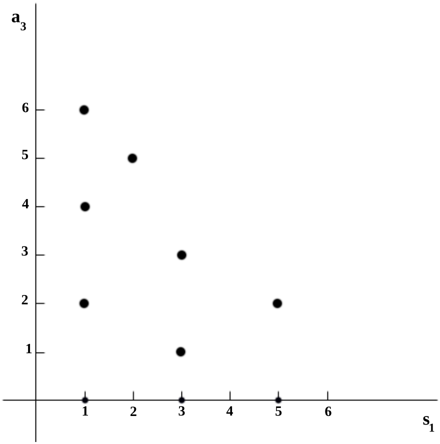

Figure: $\mathcal{V}(J_{\mathcal{S},a_l})$ in a stratified case

Structure theorem

Let G be a reduced Gröbner basis of J w.r.t. $>$. The elements of $G \cap (\mathbb{K}[\mathcal{S}, \mathsf{a}_L, \ldots, \mathsf{a}_1] \setminus \mathbb{K}[\mathcal{S}])$ can be collected into non-empty blocks $\{G_i\}_{1 \leq \iota \leq L}$ and each $\{G_i\}$ can be decomposed into blocks of polynomials according to their degree with respect to the variable a_i:

$$G_i = \sqcup_\ell G_{i\ell}.$$

Structure theorem

Let G be a reduced Gröbner basis of J w.r.t. $>$. The elements of $G \cap (\mathbb{K}[\mathcal{S}, a_L, \dots, a_1] \setminus \mathbb{K}[\mathcal{S}])$ can be collected into non-empty blocks $\{G_i\}_{1 \le \iota \le L}$ and each $\{G_i\}$ can be decomposed into blocks of polynomials according to their degree with respect to the variable a_i:

$$G_i = \sqcup_\ell G_{i\ell}.$$

Proposition

Let J be a stratified ideal w.r.t. the $\mathcal{A}$ variable. Let G be a reduced Gröbner basis of J w.r.t. $>$. Then

- $G_i = \sqcup_{\delta=1}^{i} G_{i\delta}$ *and* $G_{i\delta} \neq \emptyset$, $1 \le i \le t$ *and* $1 \le \delta \le i$;
- $G_{ii} = \{g_{ii1}\}$, $1 \le i \le L$, *i.e. exactly one polynomial exists with degree* i *w.r.t. the variable* a_i *in* G_i;
- $T(g_{ii1}) = a_i^i$.

Structure theorem

Let G be a reduced Gröbner basis of J w.r.t. $>$. The elements of $G \cap (\mathbb{K}[\mathcal{S}, \mathrm{a}_L, \ldots, \mathrm{a}_1] \setminus \mathbb{K}[\mathcal{S}])$ can be collected into non-empty blocks $\{G_i\}_{1 \leq \iota \leq L}$ and each $\{G_i\}$ can be decomposed into blocks of polynomials according to their degree with respect to the variable a_i:

$$G_i = \sqcup_\ell G_{i\ell}.$$

Proposition

Let J be a stratified ideal w.r.t. the $\mathcal{A}$ variable. Let G be a reduced Gröbner basis of J w.r.t. $>$. Then

- $G_i = \sqcup_{\delta=1}^{i} G_{i\delta}$ *and* $G_{i\delta} \neq \emptyset$, $1 \leq i \leq t$ *and* $1 \leq \delta \leq i$;
- $G_{ii} = \{g_{ii1}\}$, $1 \leq i \leq L$, *i.e. exactly one polynomial exists with degree* i *w.r.t. the variable* a_i *in* G_i;
- $T(g_{ii1}) = \mathrm{a}_i^i$.

Return

Structure theorem

Let G be a reduced Gröbner basis of J w.r.t. $>$. The elements of $G \cap (\mathbb{K}[\mathcal{S}, \mathrm{a}_L, \ldots, \mathrm{a}_1] \setminus \mathbb{K}[\mathcal{S}])$ can be collected into non-empty blocks $\{G_i\}_{1\leq\iota\leq L}$ and each $\{G_i\}$ can be decomposed into blocks of polynomials according to their degree with respect to the variable a_i:

$$G_i = \sqcup_{\ell} G_{i\ell}.$$

Proposition

Let J be a stratified ideal w.r.t. the $\mathcal{A}$ variable. Let G be a reduced Gröbner basis of J w.r.t. $>$. Then

- $G_i = \sqcup_{\delta=1}^{i} G_{i\delta}$ *and* $G_{i\delta} \neq \emptyset$, $1 \leq i \leq t$ *and* $1 \leq \delta \leq i$;
- $G_{ii} = \{g_{ii1}\}$, $1 \leq i \leq L$, *i.e. exactly one polynomial exists with degree* i *w.r.t. the variable* a_i *in* G_i;
- $T(g_{ii1}) = \mathrm{a}_i^i$.

Structure theorem

Let G be a reduced Gröbner basis of J w.r.t. $>$. The elements of $G \cap (\mathbb{K}[\mathcal{S}, \mathrm{a}_L, \ldots, \mathrm{a}_1] \setminus \mathbb{K}[\mathcal{S}])$ can be collected into non-empty blocks $\{G_i\}_{1 \le \iota \le L}$ and each $\{G_i\}$ can be decomposed into blocks of polynomials according to their degree with respect to the variable a_i:

$$G_i = \sqcup_\ell G_{i\ell}.$$

Proposition

Let J be a stratified ideal w.r.t. the $\mathcal{A}$ variable. Let G be a reduced Gröbner basis of J w.r.t. $>$. Then

- $G_i = \sqcup_{\delta=1}^{i} G_{i\delta}$ *and* $G_{i\delta} \neq \emptyset$, $1 \le i \le t$ *and* $1 \le \delta \le i$;
- $G_{ii} = \{g_{ii1}\}$, $1 \le i \le L$, *i.e. exactly one polynomial exists with degree* i *w.r.t. the variable* a_i *in* G_i;
- $T(g_{ii1}) = \mathrm{a}_i^i$.

Return

Defining a new syndrome variety

We use the variables $(x_1, \ldots, x_{n-k})$, $(z_1, \ldots, z_t)$ and $(y_1, \ldots, y_t)$ as before.

Defining a new syndrome variety

We use the variables $(x_1, \ldots, x_{n-k})$, $(z_1, \ldots, z_t)$ and $(y_1, \ldots, y_t)$ as before.

Definition

Let $n \in \mathbb{N}$ be an integer. We denote by $\mathrm{p}(n, z_l, z_{\tilde{l}}) \in \mathbb{F}_q[z_1, \ldots, z_t]$ the polynomial:

$$\mathrm{p}(n, z_l, z_{\tilde{l}}) = \frac{z_l^n - z_{\tilde{l}}^n}{z_l - z_{\tilde{l}}}, \quad 1 \le l < \tilde{l} \le t.$$

Defining a new syndrome variety

We use the variables $(x_1, \dots, x_{n-k})$, $(z_1, \dots, z_t)$ and $(y_1, \dots, y_t)$ as before.

Definition

Let $n \in \mathbb{N}$ be an integer. We denote by $\mathrm{p}(n, z_l, z_{\tilde{l}}) \in \mathbb{F}_q[z_1, \dots, z_t]$ the polynomial:

$$\mathrm{p}(n, z_l, z_{\tilde{l}}) = \frac{z_l^n - z_{\tilde{l}}^n}{z_l - z_{\tilde{l}}}, \quad 1 \le l < \tilde{l} \le t.$$

We denote by I' the ideal $\mathcal{I}(\mathcal{F}') \subset \mathbb{F}_q[x_1, \dots, x_{n-k}, z_1, \dots, z_t, y_1, \dots, y_t]$, where $\mathcal{F}' = \{f_i, \chi_i, h_j, \lambda_j, \eta_{\tilde{l},l} \mid 1 \le j \le t,\ i \in S_C,\ 1 \le \tilde{l} < l \le t\}$, with

$$\mathcal{F}' = \begin{cases} f_i := \sum_{j=1}^{t} y_j z_j^i - x_i, \\ \chi_i := x_i^{q^m} - x_i \\ h_j := z_j^{n+1} - z_j, \\ \lambda_j := y_j^{q-1} - 1 \\ \eta_{\tilde{l},l} := z_{\tilde{l}} \cdot z_l \cdot \mathrm{p}(n, z_{\tilde{l}}, z_l) \end{cases}$$

$V(I')$ is a *new syndrome variety*.

General error locator polynomial for cyclic codes

These polynomials remove all the spuriuos solutions

Let G be the reduced Gröbner basis of I' w.r.t. the lex ordering with $x_1 < \cdots < x_{n-k} < z_t < \cdots < z_1 < y_1 < \cdots < y_t$.

Theorem

Let C be an $[n,k,d]_q$ cyclic code. Let I' and G be defined as above. Then:

- *ideal I' is a stratified ideal*
- *in G there exists a unique polynomial of type*

$$g = z_t^t + a_{t-1} z^{t-1} + \cdots + a_0, \quad a_i \in \mathbb{F}_q[X].$$

General error locator polynomial for cyclic codes

Let g be the unique polynomial with degree t w.r.t. variable z_t in G_t:

$$g = z_t^t + \sum_{l=1}^{t} a_{t-l} z_t^{t-l}$$

General error locator polynomial for cyclic codes

Let g be the unique polynomial with degree t w.r.t. variable z_t in G_t:

$$g = z_t^t + \sum_{l=1}^{t} a_{t-l} z_t^{t-l}$$

- there are exactly μ errors;
- $a_{t-l}(s) = 0$ for $l > \mu$ and $a_{t-\mu}(s) \neq 0$;
- $g(s, z_t) = z^{t-\mu}(L_e(z))$;

General error locator polynomial for cyclic codes

Let g be the unique polynomial with degree t w.r.t. variable z_t in G_t:

$$g = z_t^t + \sum_{l=1}^{t} a_{t-l} z_t^{t-l}$$

- there are exactly μ errors;
- $a_{t-l}(s) = 0$ for $l > \mu$ and $a_{t-\mu}(s) \neq 0$;
- $g(s, z_t) = z^{t-\mu}(L_e(z))$;

General error locator polynomial for cyclic codes

Let g be the unique polynomial with degree t w.r.t. variable z_t in G_t:

$$g = z_t^t + \sum_{l=1}^{t} a_{t-l} z_t^{t-l}$$

- there are exactly μ errors;
- $a_{t-l}(s) = 0$ for $l > \mu$ and $a_{t-\mu}(s) \neq 0$;
- $g(s, z_t) = z^{t-\mu} \left(L_e(z)\right)$;

General error locator polynomial for cyclic codes

Let g be the unique polynomial with degree t w.r.t. variable z_t in G_t:

$$g = z_t^t + \sum_{l=1}^{t} a_{t-l} z_t^{t-l}$$

- there are exactly μ errors;
- $a_{t-l}(s) = 0$ for $l > \mu$ and $a_{t-\mu}(s) \neq 0$;
- $g(s, z_t) = z^{t-\mu}\left(L_e(z)\right)$;

and imply that $\sigma(z) = z^{\mu} g(s, z^{-1})$.

General error locator polynomial for cyclic codes

Let g be the unique polynomial with degree t w.r.t. variable z_t in G_t:

$$g = z_t^t + \sum_{l=1}^{t} a_{t-l} z_t^{t-l}$$

- there are exactly μ errors;
- $a_{t-l}(s) = 0$ for $l > \mu$ and $a_{t-\mu}(s) \neq 0$;
- $g(s, z_t) = z^{t-\mu}(L_e(z))$;

and imply that $\sigma(z) = z^{\mu} g(s, z^{-1})$. This means that g is a monic polynomial in $\mathcal{Q}[z]$ which satisfies the following property:

> given a syndrome vector $s = (s_1, \ldots, s_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$ corresponding to an error with weight $\mu \leq t$, then its t roots are the μ error locations plus zero counted with multiplicity $t - \mu$,

and is a **general error locator polynomial** of C.

Decoding algorithm

Once we have computed a general error locator polynomial for the code C, the decoding algorithm is straightforward:

Input $\mathbf{s} = (s_1, \dots, s_{n-k})$
$\mu = t$
While $a_{t-\mu}(s_1, \dots, s_{n-k}) = 0$ **do**
$\mu := \mu - 1;$
Output μ, $L_e(z)$

Table: Decoding algorithm

Decoding algorithm

The classical approach has the following problem:

one should choose a polynomial in the Gröbner basis, specialize it at the received syndrome and then find its roots. The point is that it is not possible to know in advance which polynomial has to be chosen and, as soon as the code parameters are not trivial, there might be many candidate.

An improved was proposed by Caboara and Mora.

We enlarged the syndrome variety and we have removed exactly the "spurious solutions". The new ideal turns out to be stratified and hence to contain the gelp, which is the only polynomial that needs to be specialized.

Example

```
                SINGULAR                                   /
 A Computer Algebra System for Polynomial Computations    /   version 3-0-4
                                                        0<
   by: G.-M. Greuel, G. Pfister, H. Schoenemann           \   Nov 2007
FB Mathematik der Universitaet, D-67653 Kaiserslautern     \

> ring R= (2),(z_1,z_2,z_3,x_5,x_3,x_1),lp;
> option(redSB);
> proc p (n,b,c) {
. poly tmp; tmp=0; int i;
. for (i=0;i<n;i++) {
. tmp=tmp+b^i*c^(n-1-i);  };
. return(tmp);  };
> ideal I=z_1+z_2+z_3+x_1, z_1^3+z_2^3+z_3^3+x_3, z_1^5+z_2^5+z_3^5+x_5,
          z_1^16+z_1, z_2^16+z_2, z_3^16+z_3, x_1^16+x_1, x_3^16+x_3, x_5^16+x_5,
          z_1*z_2*p(15,z_1,z_2), z_1*z_3*p(15,z_1,z_3), z_2*z_3*p(15,z_2,z_3);
> timer=1;
> ideal J=groebner(I);
//used time: 1.21 sec
```

Example

J[1]=x_1^16+x_1

J[2]=x_3^16+x_3

J[3]=x_5*x_3^10+x_5*x_3^8*x_1^6+x_5*x_3^5+x_5*x_3^4*x_1^3+x_5*x_3^2*x_1^9+x_5*x_3*x_1^12+x_5+
x_3^10*x_1^5+x_3^8*x_1^11+x_3^5*x_1^5+x_3^4*x_1^8+x_3^2*x_1^14+x_3*x_1^2+x_1^5

J[4]=x_5^3+x_5^2*x_1^5+x_5*x_1^10+x_3^10+x_3^8*x_1^6+x_3^5+x_3^4*x_1^3+x_3^2*x_1^9+x_3*x_1^12+x_1^15

J[5]=**z_3**(x_3^15*x_1^15+x_3^15+x_1^15+1)

J[6]=**z_3^2**(x_3^15+x_3^14*x_1^3+x_3^13*x_1^6+x_3^12*x_1^9+x_3^11*x_1^12+x_3^10*x_1^15+x_3^9*x_1^3+
x_3^8*x_1^6+x_3^7*x_1^9+x_3^6*x_1^12+x_3^5*x_1^15+x_3^4*x_1^3+x_3^3*x_1^6+x_3^2*x_1^9+x_3*x_1^12+
x_1^15+1) +**z_3**(x_3^15*x_1+x_3^14*x_1^4+x_3^13*x_1^7+x_3^12*x_1^10+x_3^11*x_1^13+x_3^10*x_1+
x_3^9*x_1^4+x_3^8*x_1^7+x_3^7*x_1^10+x_3^6*x_1^13+x_3^5*x_1+x_3^4*x_1^4+x_3^3*x_1^7+x_3^2*x_1^10+
x_3*x_1^13

J[7]=**z_3^3**+**z_3^2***x_1+**z_3**(x_5*x_3^9+x_5*x_3^8*x_1^3+x_5*x_3^4+x_5*x_3*x_1^9+x_3^15*x_1^2+x_3^14*x_1^5+
x_3^13*x_1^8+x_3^12*x_1^11+x_3^11*x_1^14+x_3^10*x_1^2+x_3^7*x_1^11+x_3^6*x_1^14+x_3^5*x_1^2+
x_3^3*x_1^8+x_3^2*x_1^11+x_1^2)+x_5*x_3^9*x_1+x_5*x_3^8*x_1^4+x_5*x_3^4*x_1+x_5*x_3*x_1^10+
x_3^15*x_1^3+x_3^14*x_1^6+x_3^13*x_1^9+x_3^12*x_1^12+x_3^11*x_1^15+x_3^10*x_1^3+x_3^7*x_1^12+
x_3^6*x_1^15+x_3^5*x_1^3+x_3^3*x_1^9+x_3^2*x_1^12+x_3

J[8]=**z_2**(x_3^15*x_1^15+x_3^15+x_1^15+1)

J[9]=**z_2**(z_3*x_3^15+z_3*x_3^14*x_1^3+z_3*x_3^13*x_1^6+z_3*x_3^12*x_1^9+z_3*x_3^11*x_1^12+z_3*x_3^10*
x_1^15+z_2*z_3*x_3^9*x_1^3+z_3*x_3^8*x_1^6+z_3*x_3^7*x_1^9+z_3*x_3^6*x_1^12+z_3*x_3^5*x_1^15+
z_3*x_3^4*x_1^3+*z_3*x_3^3*x_1^6+z_2*z_3*x_3^2*x_1^9+z_3*x_3*x_1^12+z_3*x_1^15+z_3)

J[10]=**z_2^2**+**z_2**(z_3+x_1)+z_3^2+z_3*x_1+x_5*x_3^9+x_5*x_3^8*x_1^3+x_5*x_3^4+x_5*x_3*x_1^9+x_3^15*x_1^2+
x_3^14*x_1^5+x_3^13*x_1^8+x_3^12*x_1^11+x_3^11*x_1^14+x_3^10*x_1^2+x_3^7*x_1^11+x_3^6*x_1^14+
x_3^5*x_1^2+x_3^3*x_1^8+x_3^2*x_1^11+x_1^2

J[11]=**z_1**+z_2+z_3+x_1

Example

J[1]=x_1^16+x_1

J[2]=x_3^16+x_3

J[3]=x_5*x_3^10+x_5*x_3^8*x_1^6+x_5*x_3^5+x_5*x_3^4*x_1^3+x_5*x_3^2*x_1^9+x_5*x_3*x_1^12+x_5+
x_3^10*x_1^5+x_3^8*x_1^11+x_3^5*x_1^5+x_3^4*x_1^8+x_3^2*x_1^14+x_3*x_1^2+x_1^5

J[4]=x_5^3+x_5^2*x_1^5+x_5*x_1^10+x_3^10+x_3^8*x_1^6+x_3^5+x_3^4*x_1^3+x_3^2*x_1^9+x_3*x_1^12+x_1^15

J[5]=**z_3**(x_3^15*x_1^15+x_3^15+x_1^15+1)

J[6]=**z_3^2**(x_3^15+x_3^14*x_1^3+x_3^13*x_1^6+x_3^12*x_1^9+x_3^11*x_1^12+x_3^10*x_1^15+x_3^9*x_1^3+
x_3^8*x_1^6+x_3^7*x_1^9+x_3^6*x_1^12+x_3^5*x_1^15+x_3^4*x_1^3+x_3^3*x_1^6+x_3^2*x_1^9+x_3*x_1^12+
x_1^15+1) +**z_3**(x_3^15*x_1+x_3^14*x_1^4+x_3^13*x_1^7+x_3^12*x_1^10+x_3^11*x_1^13+x_3^10*x_1+
x_3^9*x_1^4+x_3^8*x_1^7+x_3^7*x_1^10+x_3^6*x_1^13+x_3^5*x_1+x_3^4*x_1^4+x_3^3*x_1^7+x_3^2*x_1^10+
x_3*x_1^13

J[7]=**z_3^3**+**z_3^2***x_1+**z_3**(x_5*x_3^9+x_5*x_3^8*x_1^3+x_5*x_3^4+x_5*x_3*x_1^9+x_3^15*x_1^2+x_3^14*x_1^5+
x_3^13*x_1^8+x_3^12*x_1^11+x_3^11*x_1^14+x_3^10*x_1^2+x_3^7*x_1^11+x_3^6*x_1^14+x_3^5*x_1^2+
x_3^3*x_1^8+x_3^2*x_1^11+x_1^2)+x_5*x_3^9*x_1+x_5*x_3^8*x_1^4+x_5*x_3^4*x_1+x_5*x_3*x_1^10+
x_3^15*x_1^3+x_3^14*x_1^6+x_3^13*x_1^9+x_3^12*x_1^12+x_3^11*x_1^15+x_3^10*x_1^3+x_3^7*x_1^12+
x_3^6*x_1^15+x_3^5*x_1^3+x_3^3*x_1^9+x_3^2*x_1^12+x_3

J[8]=**z_2**(x_3^15*x_1^15+x_3^15+x_1^15+1)

J[9]=**z_2**(z_3*x_3^15+z_3*x_3^14*x_1^3+z_3*x_3^13*x_1^6+z_3*x_3^12*x_1^9+z_3*x_3^11*x_1^12+z_3*x_3^10*
x_1^15+z_2*z_3*x_3^9*x_1^3+z_3*x_3^8*x_1^6+z_3*x_3^7*x_1^9+z_3*x_3^6*x_1^12+z_3*x_3^5*x_1^15+
z_3*x_3^4*x_1^3+*z_3*x_3^3*x_1^6+z_2*z_3*x_3^2*x_1^9+z_3*x_3*x_1^12+z_3*x_1^15+z_3)

J[10]=**z_2^2**+**z_2**(z_3+x_1)+z_3^2+z_3*x_1+x_5*x_3^9+x_5*x_3^8*x_1^3+x_5*x_3^4+x_5*x_3*x_1^9+x_3^15*x_1^2+
x_3^14*x_1^5+x_3^13*x_1^8+x_3^12*x_1^11+x_3^11*x_1^14+x_3^10*x_1^2+x_3^7*x_1^11+x_3^6*x_1^14+
x_3^5*x_1^2+x_3^3*x_1^8+x_3^2*x_1^11+x_1^2

J[11]=**z_1**+z_2+z_3+x_1

Example

$$
\begin{aligned}
g_{3,1,1} =& \mathbf{z_3}(x_3^{15}x_1^{15} + x_3^{15} + x_1^{15} + 1)\\
g_{3,2,1} =& \mathbf{z_3^2}(x_3^{15} + x_3^{14}x_1^3 + x_3^{13}x_1^6 + x_3^{12}x_1^9 + x_3^{11}x_1^{12} + x_3^{10}x_1^{15} + x_3^9x_1^3 + x_3^8x_1^6 + x_3^7x_1^9 + x_3^6x_1^{12} + x_3^5x_1^{15} + x_3^4x_1^3 +\\
& x_3^3x_1^6 + x_3^2x_1^9 + x_3x_1^{12} + x_1^{15} + 1) + z_3(x_3^{15}x_1 + x_3^{14}x_1^4 + x_3^{13}x_1^7 + x_3^{12}x_1^{10} + x_3^{11}x_1^{13} +\\
& x_3^{10}x_1 + x_3^9x_1^4 + x_3^8x_1^7 + x_3^7x_1^{10} + x_3^6x_1^{13} + x_3^5x_1 + x_3^4x_1^4 + x_3^3x_1^7 + x_3^2x_1^{10} + x_3x_1^{13}\\
g_{3,3,1} =& \mathbf{z_3^3} + z_3^2x_1 + z_3(x_5x_3^9 + x_5x_3^8x_1^3 + x_5x_3^4 + x_5x_3x_1^9 + x_3^{15}x_1^2 + x_3^{14}x_1^5 + x_3^{13}x_1^8 + x_3^{12}x_1^{11} + x_3^{11}x_1^{14} +\\
& x_3^{10}x_1^2 + x_3^7x_1^{11} + x_3^6x_1^{14} + x_3^5x_1^2 + x_3^3x_1^8 + x_3^2x_1^{11} + x_1^2) + x_5x_3^9x_1 + x_5x_3^8x_1^4 + x_5x_3^4x_1 + x_5x_3x_1^{10} + x_3^{15}x_1^3 + x_3^{14}x_1^6 +\\
& x_3^{13}x_1^9 + x_3^{12}x_1^{12} + x_3^{11}x_1^{15} + x_3^{10}x_1^3 + x_3^7x_1^{12} + x_3^6x_1^{15} + x_3^5x_1^3 + x_3^3x_1^9 + x_3^2x_1^{12} + x_3\\
g_{2,1,1} =& \mathbf{z_2}(x_3^{15}x_1^{15} + x_3^{15} + x_1^{15} + 1)\\
g_{2,1,2} =& \mathbf{z_2}(z_3x_3^{15} + z_3x_3^{14}x_1^3 + z_3x_3^{13}x_1^6 + z_3x_3^{12}x_1^9 + z_3x_3^{11}x_1^{12} + z_3x_3^{10}x_1^{15} + z_2z_3x_3^9x_1^3 +\\
& z_3x_3^8x_1^6 + z_3x_3^7x_1^9 + z_3x_3^6x_1^{12} + z_3x_3^5x_1^{15} + z_3x_3^4x_1^3 + z_3x_3^3x_1^6 + z_2z_3x_3^2x_1^9 + z_3x_3x_1^{12} + z_3x_1^{15} + z_3)\\
g_{2,2,1} =& \mathbf{z_2^2} + z_2(z_3 + x_1) + z_3^2 + z_3x_1 + x_5x_3^9 + x_5x_3^8x_1^3 + x_5x_3^4 + x_5x_3x_1^9 + x_3^{15}x_1^2 + x_3^{14}x_1^5 + x_3^{13}x_1^8 +\\
& x_3^{12}x_1^{11} + x_3^{11}x_1^{14} + x_3^{10}x_1^2 + x_3^7x_1^{11} + x_3^6x_1^{14} + x_3^5x_1^2 + x_3^3x_1^8 + x_3^2x_1^{11} + x_1^2\\
g_{1,1,1} =& \mathbf{z_1} + z_2 + z_3 + x_1
\end{aligned}
$$

$$
\begin{aligned}
G_3 &= \{G_{3,3}, G_{3,2}, G_{3,1}\} & G_{3,3} &= \{g_{3,3,1}\}, G_{3,2} = \{g_{3,2,1}\}, G_{3,1} = \{g_{3,1,1}\}\\
G_2 &= \{G_{2,2}, G_{2,1}\} & G_{2,2} &= \{g_{2,2,1}\}, G_{2,1} = \{g_{2,1,1}, g_{2,1,2}\}\\
G_1 &= \{G_{1,1}\} & G_{1,1} &= \{g_{1,1,1}\}
\end{aligned}
$$

Example

$$
\begin{aligned}
g_{3,3,1} = & \mathbf{z_3^3} + z_3^2 x_1 + z_3 (x_5 x_3^9 + x_5 x_3^8 x_1^3 + x_5 x_3^4 + x_5 x_3 x_1^9 + x_3^{15} x_1^2 + x_3^{14} x_1^5 + x_3^{13} x_1^8 + x_3^{12} x_1^{11} + x_3^{11} x_1^{14} + \\
& x_3^{10} x_1^2 + x_3^7 x_1^{11} + x_3^6 x_1^{14} + x_3^5 x_1^2 + x_3^3 x_1^8 + x_3^2 x_1^{11} + x_1^2) + x_5 x_3^9 x_1 + x_5 x_3^8 x_1^4 + x_5 x_3^4 x_1 + x_5 x_3 x_1^{10} + x_3^{15} x_1^3 + x_3^{14} x_1^6 + \\
& x_3^{13} x_1^9 + x_3^{12} x_1^{12} + x_3^{11} x_1^{15} + x_3^{10} x_1^3 + x_3^7 x_1^{12} + x_3^6 x_1^{15} + x_3^5 x_1^3 + x_3^3 x_1^9 + x_3^2 x_1^{12} + x_3
\end{aligned}
$$

Example

$$
\begin{aligned}
g_{3,3,1} = {} & \mathbf{z_3^3} + z_3^2 x_1 + z_3 (x_5 x_3^9 + x_5 x_3^8 x_1^3 + x_5 x_3^4 + x_5 x_3 x_1^9 + x_3^{15} x_1^2 + x_3^{14} x_1^5 + x_3^{13} x_1^8 + x_3^{12} x_1^{11} + x_3^{11} x_1^{14} + \\
& x_3^{10} x_1^2 + x_3^7 x_1^{11} + x_3^6 x_1^{14} + x_3^5 x_1^2 + x_3^3 x_1^8 + x_3^2 x_1^{11} + x_1^2) + x_5 x_3^9 x_1 + x_5 x_3^8 x_1^4 + x_5 x_3^4 x_1 + x_5 x_3 x_1^{10} + x_3^{15} x_1^3 + x_3^{14} x_1^6 + \\
& x_3^{13} x_1^9 + x_3^{12} x_1^{12} + x_3^{11} x_1^{15} + x_3^{10} x_1^3 + x_3^7 x_1^{12} + x_3^6 x_1^{15} + x_3^5 x_1^3 + x_3^3 x_1^9 + x_3^2 x_1^{12} + x_3
\end{aligned}
$$

1. We suppose the $c = (0, 0, \ldots, 0)$ is the transmitted word.
   Let $v = (1, 0, 1, 1, 0, \ldots, 0)$ be the received vector, then $\mu = 3$ and
   $$x_1 = \alpha^{13} \qquad x_3 = \alpha^{10} \qquad x_5 = \alpha^{10}$$

Example

$$
\begin{aligned}
g_{3,3,1} = {} & \mathbf{z_3^3} + z_3^2 x_1 + z_3 (x_5 x_3^9 + x_5 x_3^8 x_1^3 + x_5 x_3^4 + x_5 x_3 x_1^9 + x_3^{15} x_1^2 + x_3^{14} x_1^5 + x_3^{13} x_1^8 + x_3^{12} x_1^{11} + x_3^{11} x_1^{14} + \\
& x_3^{10} x_1^2 + x_3^7 x_1^{11} + x_3^6 x_1^{14} + x_3^5 x_1^2 + x_3^3 x_1^8 + x_3^2 x_1^{11} + x_1^2) + x_5 x_3^9 x_1 + x_5 x_3^8 x_1^4 + x_5 x_3^4 x_1 + x_5 x_3 x_1^{10} + x_3^{15} x_1^3 + x_3^{14} x_1^6 + \\
& x_3^{13} x_1^9 + x_3^{12} x_1^{12} + x_3^{11} x_1^{15} + x_3^{10} x_1^3 + x_3^7 x_1^{12} + x_3^6 x_1^{15} + x_3^5 x_1^3 + x_3^3 x_1^9 + x_3^2 x_1^{12} + x_3
\end{aligned}
$$

1. We suppose the $c = (0, 0, \ldots, 0)$ is the transmitted word.
   Let $v = (1, 0, 1, 1, 0, \ldots, 0)$ be the received vector, then $\mu = 3$ and

   $$x_1 = \alpha^{13} \qquad x_3 = \alpha^{10} \qquad x_5 = \alpha^{10}$$

```
> subst(subst(subst(g,x_1,a^13),x_3,a^10),x_5,a^10);
z_3^3+a^13*z_3^2+a^9*z_3+a^5
> poly gs=z_3^3+a^13*z_3^2+a^9*z_3+a^5;
> subst(gs,z_3,1);
0
> subst(gs,z_3,a);
a^3
> subst(gs,z_3,a^2);
0
> subst(gs,z_3,a^3);
0
```

Example

2. We suppose the $c = (0, 0, \dots, 0)$ is the transmitted word.
   Let $v = (1, 0, 0, 1, 0, \dots, 0)$ be the received vector, then $\mu = 2$ and

$$x_1 = \alpha^{14} \quad x_3 = \alpha^{7} \quad x_5 = 0$$

Example

2. We suppose the $c = (0, 0, \ldots, 0)$ is the transmitted word.
   Let $v = (1, 0, 0, 1, 0, \ldots, 0)$ be the received vector, then $\mu = 2$ and

$$x_1 = \alpha^{14} \quad x_3 = \alpha^7 \quad x_5 = 0$$

```
> subst(subst(subst(g,x_1,a^14),x_3,a^7),x_5,0);
z_3^3+a^14*z_3^2+a^3*z_3
> poly gs=z_3^2+a^14*z_3+a^3;
> subst(gs,z_3,1);
0
> subst(gs,z_3,a);
a^13
> subst(gs,z_3,a^2);
a^14
> subst(gs,z_3,a^3);
0
```

Example

3. We suppose the $c = (0, 0, \dots, 0)$ is the transmitted word.
   Let $v = (0, 1, 0, 0, 0, \dots, 0)$ be the received vector, then $\mu = 1$ and

$$x_1 = \alpha \quad x_3 = \alpha^3 \quad x_5 = \alpha^5$$

Example

3. We suppose the $c = (0, 0, \dots, 0)$ is the transmitted word.
   Let $v = (0, 1, 0, 0, 0, \dots, 0)$ be the received vector, then $\mu = 1$ and

$$x_1 = \alpha \quad x_3 = \alpha^3 \quad x_5 = \alpha^5$$

```
> subst(subst(subst(g,x_1,a),x_3,a^3),x_5,a^5);
z_3^3+a*z_3^2
> poly gs=z_3+a;
```

Outline

1. Introduction
   - Notation and preliminaries
   - Syndrome variety
   - A decoding algorithm
2. General error locator polynomial
   - General error locator polynomial
   - Properties of stratified ideals
   - A new syndrome variety
   - A new decoding algorithm
3. Conclusions
   - General error locator polynomial for linear codes
   - Correcting erasures via the syndrome variety
   - Multidimensional general error locator polynomials
   - Efficiency of the proposed algorithm

Remark1

We note that the definition of general error locator polynomial are for generic linear code, so general error locator polynomials can be used to decode any linear code, if it possesses them.

Remark1

We note that the definition of general error locator polynomial are for generic linear code, so general error locator polynomials can be used to decode any linear code, if it possesses them.

Remark1

We note that the definition of general error locator polynomial are for generic linear code, so general error locator polynomials can be used to decode any linear code, if it possesses them.

It is important to note that even if in some special cases the decoding with the general error locator polynomial is very fast, this nice behavior cannot be generalized to all linear codes.

Remark1

We note that the definition of general error locator polynomial are for generic linear code, so general error locator polynomials can be used to decode any linear code, if it possesses them.

It is important to note that even if in some special cases the decoding with the general error locator polynomial is very fast, this nice behavior cannot be generalized to all linear codes.

N. Bruck and M. Naor, *The hardness of decoding linear codes with preprocessing*, IEEE Trans. Inform. Theory 36 (1990), 381 – 385.

Remark2

Let C be an $[n, k, d]_q$ cyclic code with defining set $S_C = \{i_1, \ldots, i_{n-k}\}$. Let τ be to the number of errors, ν be the number of erasures s.t. $2\tau + \nu < d$.

Remark2

Let C be an $[n,k,d]_q$ cyclic code with defining set $S_C = \{i_1, \ldots, i_{n-k}\}$. Let τ be to the number of errors, ν be the number of erasures s.t. $2\tau + \nu < d$.
We denote by $\{\alpha^l \mid 1 \le l \le \tau\}$ the set of the error locations and by $\{\alpha^h \mid 1 \le h \le \nu\}$ the set of the erasure locations.

Remark2

Let C be an $[n,k,d]_q$ cyclic code with defining set $S_C = \{i_1, \dots, i_{n-k}\}$. Let τ be to the number of errors, ν be the number of erasures s.t. $2\tau + \nu < d$.
We denote by $\{\alpha^l \mid 1 \le l \le \tau\}$ the set of the error locations and by $\{\alpha^h \mid 1 \le h \le \nu\}$ the set of the erasure locations.

$$\sum_{l=1}^{\tau} a_l(\alpha^l)^i + \sum_{h=1}^{\nu} c_h(\alpha^h)^i - s_i, \quad i \in S_C,$$

where $\{\alpha^l\}, \{a_l\}$ and $\{c_h\}$ are unknown and $\{s_i\}, \{\alpha^h\}$ are known.

Remark2

Let C be an $[n,k,d]_q$ cyclic code with defining set $S_C = \{i_1, \dots, i_{n-k}\}$. Let τ be to the number of errors, ν be the number of erasures s.t. $2\tau + \nu < d$.
We denote by $\{\alpha^l \mid 1 \leq l \leq \tau\}$ the set of the error locations and by $\{\alpha^h \mid 1 \leq h \leq \nu\}$ the set of the erasure locations.

$$\sum_{l=1}^{\tau} a_l(\alpha^l)^i + \sum_{h=1}^{\nu} c_h(\alpha^h)^i - s_i, \quad i \in S_C,$$

where $\{\alpha^l\}, \{a_l\}$ and $\{c_h\}$ are unknown and $\{s_i\}, \{\alpha^h\}$ are known.

variables	representant
$x_1, \dots, x_{n-k}$	correctable syndromes
$z_1, \dots, z_\tau$	error locations
$y_1, \dots, y_\tau$	error values
$w_1, \dots, w_\nu$	erasure locations
$u_1, \dots, u_\nu$	erasure values

Remark2

Let C be an $[n,k,d]_q$ cyclic code with defining set $S_C = \{i_1, \dots, i_{n-k}\}$. Let τ be to the number of errors, ν be the number of erasures s.t. $2\tau + \nu < d$.
We denote by $\{\alpha^l \mid 1 \le l \le \tau\}$ the set of the error locations and by $\{\alpha^h \mid 1 \le h \le \nu\}$ the set of the erasure locations.

$$\sum_{l=1}^{\tau} a_l (\alpha^l)^i + \sum_{h=1}^{\nu} c_h (\alpha^h)^i - s_i, \quad i \in S_C,$$

where $\{\alpha^l\}, \{a_l\}$ and $\{c_h\}$ are unknown and $\{s_i\}, \{\alpha^h\}$ are known.

variables	representant
$x_1, \dots, x_{n-k}$	correctable syndromes
$z_1, \dots, z_\tau$	error locations
$y_1, \dots, y_\tau$	error values
$w_1, \dots, w_\nu$	erasure locations
$u_1, \dots, u_\nu$	erasure values

Remark2

Let C be an $[n,k,d]_q$ cyclic code with defining set $S_C = \{i_1,\dots,i_{n-k}\}$. Let τ be to the number of errors, ν be the number of erasures s.t. $2\tau+\nu<d$.
We denote by $\{\alpha^l \mid 1 \le l \le \tau\}$ the set of the error locations and by $\{\alpha^h \mid 1 \le h \le \nu\}$ the set of the erasure locations.

$$\sum_{l=1}^{\tau} a_l(\alpha^l)^i + \sum_{h=1}^{\nu} c_h(\alpha^h)^i - s_i, \quad i \in S_C,$$

where $\{\alpha^l\}, \{a_l\}$ and $\{c_h\}$ are unknown and $\{s_i\}, \{\alpha^h\}$ are known.

variables	representant
$x_1,\dots,x_{n-k}$	correctable syndromes
$z_1,\dots,z_\tau$	error locations
$y_1,\dots,y_\tau$	error values
$w_1,\dots,w_\nu$	erasure locations
$u_1,\dots,u_\nu$	erasure values

Remark2

Let C be an $[n,k,d]_q$ cyclic code with defining set $S_C = \{i_1, \dots, i_{n-k}\}$. Let τ be to the number of errors, ν be the number of erasures s.t. $2\tau + \nu < d$.
We denote by $\{\alpha^l \mid 1 \le l \le \tau\}$ the set of the error locations and by $\{\alpha^h \mid 1 \le h \le \nu\}$ the set of the erasure locations.

$$\sum_{l=1}^{\tau} a_l(\alpha^l)^i + \sum_{h=1}^{\nu} c_h(\alpha^h)^i - s_i, \quad i \in S_C,$$

where $\{\alpha^l\}, \{a_l\}$ and $\{c_h\}$ are unknown and $\{s_i\}, \{\alpha^h\}$ are known.

variables	representant
$x_1, \dots, x_{n-k}$	correctable syndromes
$z_1, \dots, z_\tau$	error locations
$y_1, \dots, y_\tau$	error values
$w_1, \dots, w_\nu$	erasure locations
$u_1, \dots, u_\nu$	erasure values

Remark2

Let C be an $[n,k,d]_q$ cyclic code with defining set $S_C = \{i_1,\dots,i_{n-k}\}$. Let τ be to the number of errors, ν be the number of erasures s.t. $2\tau+\nu<d$.
We denote by $\{\alpha^l \mid 1 \le l \le \tau\}$ the set of the error locations and by $\{\alpha^h \mid 1 \le h \le \nu\}$ the set of the erasure locations.

$$\sum_{l=1}^{\tau} a_l(\alpha^l)^i + \sum_{h=1}^{\nu} c_h(\alpha^h)^i - s_i, \quad i \in S_C,$$

where $\{\alpha^l\},\{a_l\}$ and $\{c_h\}$ are unknown and $\{s_i\},\{\alpha^h\}$ are known.

variables	representant
$x_1,\dots,x_{n-k}$	correctable syndromes
$z_1,\dots,z_\tau$	error locations
$y_1,\dots,y_\tau$	error values
$w_1,\dots,w_\nu$	erasure locations
$u_1,\dots,u_\nu$	erasure values

Remark2

We rewrite previous equations in terms of X, Y, Z, W and U, as:

$$\mathcal{F}^{\tau,\nu} = \begin{array}{ll} \{\{\sum_{l=1}^{\tau} y_l z_l^i + \sum_{h=1}^{\nu} u_h w_h^i - x_i\}_{i \in S_C}, & \\ \{z_l^{n+1} - z_l\}_{l=1,\dots,\tau}, & \{y_l^{q-1} - 1\}_{l=1,\dots,\tau}, \\ \{u_h^q - u_h\}_{h=1,\dots,\nu}, & \{w_h^n - 1\}_{h=1,\dots,\nu}, \\ \{x_i^{q^m} - x_i\}_{i \in S_C}, & \{\mathrm{p}(n, w_h, w_i)\}_{h \neq i, h,i=1,\dots,\nu}, \\ \{z_l \mathrm{p}(n, z_l, w_h)\}_{l=1,\dots,\tau, h=1,\dots,\nu}, & \{z_l z_k \mathrm{p}(n, z_l, z_k)\}_{l \neq k, l,k=1,\dots,\tau}\} \end{array}$$

Remark2

We rewrite previous equations in terms of X, Y, Z, W and U, as:

$$
\begin{array}{lll}
\mathcal{F}^{\tau,\nu} = & \{\{\sum_{l=1}^{\tau} y_l z_l^i + \sum_{h=1}^{\nu} u_h w_h^i - x_i\}_{i\in S_C}, & \\
& \{z_l^{n+1} - z_l\}_{l=1,\dots,\tau}, & \{y_l^{q-1} - 1\}_{l=1,\dots,\tau}, \\
& \{u_h^q - u_h\}_{h=1,\dots,\nu}, & \{w_h^n - 1\}_{h=1,\dots,\nu}, \\
& \{x_i^{q^m} - x_i\}_{i\in S_C}, & \{\mathrm{p}(n, w_h, w_i)\}_{h\neq i, h,i=1,\dots,\nu}, \\
& \{z_l \mathrm{p}(n, z_l, w_h)\}_{l=1,\dots,\tau, h=1,\dots,\nu}, & \{z_l z_k \mathrm{p}(n, z_l, z_k)\}_{l\neq k, l,k=1,\dots,\tau}\}
\end{array}
$$

Ideal $I = \mathcal{I}(\mathcal{F}^{\tau,\nu})$ depends only on code C and on ν.

Lemma

Ideal I is stratified.

Remark2

We rewrite previous equations in terms of X, Y, Z, W and U, as:

$$\mathcal{F}^{\tau,\nu} = \begin{array}{ll} \Big\{ \{ \sum_{l=1}^{\tau} y_l z_l^i + \sum_{h=1}^{\nu} u_h w_h^i - x_i \}_{i \in S_C}, & \\ \{ z_l^{n+1} - z_l \}_{l=1,\dots,\tau}, & \{ y_l^{q-1} - 1 \}_{l=1,\dots,\tau}, \\ \{ u_h^q - u_h \}_{h=1,\dots,\nu}, & \{ w_h^n - 1 \}_{h=1,\dots,\nu}, \\ \{ x_i^{q^m} - x_i \}_{i \in S_C}, & \{ \mathrm{p}(n, w_h, w_i) \}_{h \neq i, h,i=1,\dots,\nu}, \\ \{ z_l \mathrm{p}(n, z_l, w_h) \}_{l=1,\dots,\tau, h=1,\dots,\nu}, & \{ z_l z_k \mathrm{p}(n, z_l, z_k) \}_{l \neq k, l,k=1,\dots,\tau} \Big\} \end{array}$$

Ideal $I = \mathcal{I}(\mathcal{F}^{\tau,\nu})$ depends only on code C and on ν.

Lemma

Ideal I is stratified.

Let G be the reduced Gröbner basis of I w.r.t. $>$.

In G there is a unique polynomial of type

$$g = z_\tau^\tau + a_{\tau-1} z^{\tau-1} + \cdots + a_0, \quad a_i \in \mathbb{F}_q[X, W].$$

Moreover g is an **extended general error locator polynomial**.

Remark3

It is possible to extend Cooper's ideas to decode affine–variety codes.

Remark3

It is possible to extend Cooper's ideas to decode affine–variety codes.
Let $m \geq 1$ and $I \subseteq \mathbb{F}_q[X] = \mathbb{F}_q[x_1, \ldots, x_m]$ be an ideal such that

$$E_q[X] = \{x_1^q - x_1, x_2^q - x_2, \ldots, x_m^q - x_m\} \subset I.$$

Remark3

It is possible to extend Cooper's ideas to decode affine–variety codes.
Let $m \geq 1$ and $I \subseteq \mathbb{F}_q[X] = \mathbb{F}_q[x_1, \ldots, x_m]$ be an ideal such that

$$E_q[X] = \{x_1^q - x_1, x_2^q - x_2, \ldots, x_m^q - x_m\} \subset I.$$

Let $P_1, P_2, \ldots, P_n$ be the points of the variety defined by I.

Remark3

It is possible to extend Cooper's ideas to decode affine–variety codes.
Let $m \geq 1$ and $I \subseteq \mathbb{F}_q[X] = \mathbb{F}_q[x_1, \dots, x_m]$ be an ideal such that

$$E_q[X] = \{x_1^q - x_1, x_2^q - x_2, \dots, x_m^q - x_m\} \subset I.$$

Let $P_1, P_2, \dots, P_n$ be the points of the variety defined by I.

There is an isomorphism of $\mathbb{F}_q$-vector spaces (an evaluation)

$$\begin{array}{lccc} \phi : & R = \mathbb{F}_q[x_1, \dots, x_m]/I & \longrightarrow & (\mathbb{F}_q)^n \\ \phi : & f & \longmapsto & (f(P_1), \dots, f(P_n)). \end{array}$$

Remark3

It is possible to extend Cooper's ideas to decode affine–variety codes.
Let $m \geq 1$ and $I \subseteq \mathbb{F}_q[X] = \mathbb{F}_q[x_1, \ldots, x_m]$ be an ideal such that

$$E_q[X] = \{x_1^q - x_1, x_2^q - x_2, \ldots, x_m^q - x_m\} \subset I.$$

Let $P_1, P_2, \ldots, P_n$ be the points of the variety defined by I.

There is an isomorphism of $\mathbb{F}_q$-vector spaces (an evaluation)

$$\begin{array}{lccc} \phi : & R = \mathbb{F}_q[x_1, \ldots, x_m]/I & \longrightarrow & (\mathbb{F}_q)^n \\ \phi : & f & \longmapsto & (f(P_1), \ldots, f(P_n)). \end{array}$$

Let L be a linear subspace of R over $\mathbb{F}_q$ of dimension r.

Definition

The **affine-variety code** $C(I, L)$ is the image $\phi(L)$, and the affine-variety code $C^\perp(I, L)$ is its dual code.

Remark3

If $b_1, \ldots, b_r$ is a linear basis for L over $\mathbb{F}_q$, then the matrix

$$\begin{pmatrix} b_1(P_1) & b_1(P_2) & \ldots & b_1(P_n) \\ \vdots & \vdots & \ldots & \vdots \\ b_r(P_1) & b_r(P_2) & \ldots & b_r(P_n) \end{pmatrix}$$

is a generator matrix for $C(I, L)$ and a parity-check matrix for $C^{\perp}(I, L)$.

Remark3

If $b_1, \ldots, b_r$ is a linear basis for L over $\mathbb{F}_q$, then the matrix

$$\begin{pmatrix} b_1(P_1) & b_1(P_2) & \ldots & b_1(P_n) \\ \vdots & \vdots & \ldots & \vdots \\ b_r(P_1) & b_r(P_2) & \ldots & b_r(P_n) \end{pmatrix}$$

is a generator matrix for $C(I, L)$ and a parity-check matrix for $C^{\perp}(I, L)$.

Theorem (F-L,1998)

Every linear code may be represented as an affine-variety code (both as $C(I, L)$ and as $C^{\perp}(I', L')$).

Remark3

If $b_1, \ldots, b_r$ is a linear basis for L over $\mathbb{F}_q$, then the matrix

$$\begin{pmatrix} b_1(P_1) & b_1(P_2) & \ldots & b_1(P_n) \\ \vdots & \vdots & \ldots & \vdots \\ b_r(P_1) & b_r(P_2) & \ldots & b_r(P_n) \end{pmatrix}$$

is a generator matrix for $C(I,L)$ and a parity-check matrix for $C^\perp(I,L)$.

Theorem (F-L,1998)

Every linear code may be represented as an affine-variety code (both as $C(I,L)$ and as $C^\perp(I',L')$).

Let $C = C^\perp(I,L)$ be an affine variety code with dimension $r = n-k$, distance d and parity-check matrix H.

Remark3

Let $c = (c_0, \ldots, c_{n-1})$, $v = (v_0, \ldots, v_{n-1})$ and $e = (e_0, \ldots, e_{n-1})$.

Remark3

Let $c = (c_0, \dots, c_{n-1})$, $v = (v_0, \dots, v_{n-1})$ and $e = (e_0, \dots, e_{n-1})$.

From $Hv^T = He^T = s$, we get

$$s_i = \sum_{j=1}^{n} v_j b_i(P_j) = \sum_{j=1}^{t} e_j b_i(P_j), \quad 1 \leq i \leq r,$$

where t is the correction capability of the code.

Remark3

Let $c = (c_0, \ldots, c_{n-1})$, $v = (v_0, \ldots, v_{n-1})$ and $e = (e_0, \ldots, e_{n-1})$.

From $Hv^T = He^T = s$, we get

$$s_i = \sum_{j=1}^{n} v_j b_i(P_j) = \sum_{j=1}^{t} e_j b_i(P_j), \quad 1 \leq i \leq r,$$

where t is the correction capability of the code.

- $S = (s_1, \ldots, s_r)$ for the syndromes
- $Z_t = (z_{t,1}, \ldots, z_{t,m})$, $\ldots$, $Z_1 = (z_{1,1}, \ldots, z_{1,m})$ for the error locations
- $E = (e_1, \ldots, e_t)$ for the error values.

Remark3

By changing the classical ideal for decoding affine–variety codes, previously suggested by Fitzgerald-Lax (1998), it is possible to prove the existence of multi-dimensional general error locator polynomials for any affine-code.

Remark3

By changing the classical ideal for decoding affine–variety codes, previously suggested by Fitzgerald-Lax (1998), it is possible to prove the existence of multi-dimensional general error locator polynomials for any affine-code.

Multidimensional general error locator polynomials *are the multidimensional analogue of general error locator polynomials. Once the syndromes are received, they permit direct computations of the error locations by simply evaluating some polynomials in the received syndrome.*

Remark3

Let $C^{\perp}(I, L)$ be an affine variety code, we denote by $I_*^{C,t}$ the ideal in $\mathbb{F}_q[s_1, \ldots, s_r, X_1, \ldots, X_t, e_1, \ldots, e_t]$ s.t.

$$I_*^{C,t} = \Big\langle \quad \left\{ \sum_{j=1}^{t} e_j b_i(x_{j1}, \ldots, x_{jm}) - s_i \right\}_{1 \le i \le r}, \left\{ e_j^{q-1} - 1 \right\}_{1 \le j \le t}, \{g_h(x_{j1}, \ldots, x_{jm})\}_{\substack{1 \le h \le l, \\ 1 \le j \le t}}, \left\{ x_{jl}^q - x_{jl} \right\}_{\substack{1 \le j \le t, \\ 1 \le l \le m}} \left\{ x_{jl} x_{\tilde{j}l} \prod_{1 \le l \le m} ((x_{jl} - x_{\tilde{j}l})^{q-1} - 1) \right\}_{\substack{1 \le j < \tilde{j} \le t \\ i \le l \le m}} \Big\rangle$$

Remark3

Let $C^{\perp}(I,L)$ be an affine variety code, we denote by $I_*^{C,t}$ the ideal in $\mathbb{F}_q[s_1,\dots,s_r,X_1,\dots,X_t,e_1,\dots,e_t]$ s.t.

$$I_*^{C,t} = \Big\langle \quad \Big\{\sum_{j=1}^t e_j b_i(x_{j1},\dots,x_{jm}) - s_i\Big\}_{1\le i\le r}, \Big\{e_j^{q-1}-1\Big\}_{1\le j\le t}, \\ \{g_h(x_{j1},\dots,x_{jm})\}_{\substack{1\le h\le l,\\ 1\le j\le t}}, \Big\{x_{jl}^q - x_{jl}\Big\}_{\substack{1\le j\le t,\\ 1\le l\le m}} \\ \Big\{x_{jl}x_{\tilde{j}l}\prod_{1\le l\le m}((x_{jl}-x_{\tilde{j}l})^{q-1}-1)\Big\}_{\substack{1\le j<\tilde{j}\le t\\ i\le l\le m}} \Big\rangle$$

Theorem

1. *multidimensional general error locator polynomials exist for* ***any*** *affine-variety code;*
2. *they can be easily found in a suitable Gröbner basis of $I_*^{C,t}$ (they are the polynomials with leading terms of type $x_i^{t_i}$).*

Remark3

Let $C^{\perp}(I,L)$ be an affine variety code, we denote by $I_*^{C,t}$ the ideal in $\mathbb{F}_q[s_1,\dots,s_r,X_1,\dots,X_t,e_1,\dots,e_t]$ s.t.

$$I_*^{C,t} = \Big\langle \quad \left\{\sum_{j=1}^{t} e_j b_i(x_{j1},\dots,x_{jm}) - s_i\right\}_{1\le i\le r}, \left\{e_j^{q-1}-1\right\}_{1\le j\le t}, \\ \{g_h(x_{j1},\dots,x_{jm})\}_{\substack{1\le h\le l,\\ 1\le j\le t}}, \left\{x_{jl}^q - x_{jl}\right\}_{\substack{1\le j\le t,\\ 1\le l\le m}} \\ \left\{x_{jl}x_{\tilde{j}l}\prod_{1\le l\le m}((x_{jl}-x_{\tilde{j}l})^{q-1}-1)\right\}_{\substack{1\le j<\tilde{j}\le t\\ i\le l\le m}} \Big\rangle$$

Theorem

1. *multidimensional general error locator polynomials exist for* **any** *affine-variety code;*
2. *they can be easily found in a suitable Gröbner basis of $I_*^{C,t}$ (they are the polynomials with leading terms of type $x_i^{t_i}$).*

Remark4

The efficiency of the algorithm depends on two factors:

1. The computation of the associated Gröbner basis can be quite beyond present means already for medium-size codes;
2. Even if we compute a general error locator, it could be so dense that its use would be impractical.

Remark4

The efficiency of the algorithm depends on two factors:

1. The computation of the associated Gröbner basis can be quite beyond present means already for medium-size codes;
2. Even if we compute a general error locator, it could be so dense that its use would be impractical.

Remark4

The efficiency of the algorithm depends on two factors:

1. The computation of the associated Gröbner basis can be quite beyond present means already for medium-size codes;
2. Even if we compute a general error locator, it could be so dense that its use would be impractical.

Sparsity: it is possible to obtain a sparse representation of the general error polynomial for same special classes of cylic codes. This can be done by studying the associated syndrome variety and defining set of the code. Moreover in these cases it is possible to obtain a general error locator without computing a Gröbner basis, but simply using the structure of the code

Remark4

The efficiency of the algorithm depends on two factors:

1. The computation of the associated Gröbner basis can be quite beyond present means already for medium-size codes;
2. Even if we compute a general error locator, it could be so dense that its use would be impractical.

Sparsity: it is possible to obtain a sparse representation of the general error polynomial for same special classes of cylic codes. This can be done by studying the associated syndrome variety and defining set of the code. Moreover in these cases it is possible to obtain a general error locator without computing a Gröbner basis, but simply using the structure of the code

Remark4

The efficiency of the algorithm depends on two factors:

1. The computation of the associated Gröbner basis can be quite beyond present means already for medium-size codes;
2. Even if we compute a general error locator, it could be so dense that its use would be impractical.

These two apparently different problems may have one common solution: to identify our polynomials without computing any Gröbner basis, but using the "structure of the code".

Example

Example: Let us consider the Hermitian code C defined previously:

$$y^2 + y = x^3 \quad \text{over } \mathbb{F}_4$$

with monomials $L = \{1, x, y, x^2, xy\}$. C can correct up to $t = 2$ errors.

Example

Example: Let us consider the Hermitian code C defined previously:

$$y^2 + y = x^3 \quad \text{over } \mathbb{F}_4$$

with monomials $L = \{1, x, y, x^2, xy\}$. C can correct up to $t = 2$ errors.
Let us consider the lex term-ordering with

$$e_1 > e_2 > y_2 > x_2 > y_1 > x_1 > s_5 > s_4 > s_3 > s_2 > s_1$$

and the ideal

$$I_*^{C,t} \subset \mathbb{F}_2[s_1, s_2, s_3, s_4, s_5, x_1, y_1, x_2, y_2, e_1, e_2].$$

The multidimensional general error locator polynomials for C:

$$\begin{aligned}
\mathcal{L}_{C,1} = \quad & \mathbf{x_1}^2 + \mathbf{x_1}(s_4^2 s_1 + s_4 s_2^2 s_1^3 + s_4 s_2^2 + s_2 s_1^2) + \\
& s_5^2 s_3 + s_5 s_3 s_2 + s_4^2 s_3^3 s_2 + s_4^2 s_3^2 s_2 s_1 + s_4^2 s_3 s_2 s_1^2 + s_4^2 s_2 + s_4 s_3^3 s_1^2 + s_4 s_3^2 s_1^3 + \\
& s_4 s_3 s_1 + s_4 s_1^2 + s_3^3 s_2^2 s_1 + s_3^2 s_2^2 s_1^2 + s_3 s_2^2 s_1^3 + s_3 s_2^2 + s_2^2 s_1 \\
\mathcal{L}_{C,2} = \quad & \mathbf{y_1}^2 + \mathbf{y_1} + \\
& x_1 s_4^2 s_2 s_1^3 + x_1 s_4 s_3^2 s_1^3 + x_1 s_4 s_3^2 + x_1 s_3 s_2^2 s_1^3 + x_1 s_3 s_2^2 + s_5^3 + \\
& s_5 s_4^2 s_3^2 s_2 + s_5 s_4 s_3 + s_5 s_3^3 s_2^2 + s_4^3 s_3^2 s_1 + s_4^3 s_3 s_2^3 s_1^2 + s_4^3 s_3 s_1^2 + s_4^2 s_2^2 s_1^2 + \\
& s_4 s_3^3 s_2 s_1 + s_4 s_3 s_2 s_1^3 + s_4 s_2 s_1 + s_3^3 s_2^3 + s_3^3 s_1^3 + s_3^2 s_2^3 s_1 + s_3^2 s_1 + s_3 s_1^2
\end{aligned}$$

Example

However these polynomials are by far not random and some direct manipulations shows that actual

$$
\begin{aligned}
{}''\mathcal{L}_{C,1}'' = \quad & \mathbf{x_1}^2 + \mathbf{x_1}(s_4^2 s_1 + s_4 s_2^2 s_1^3 + s_4 s_2^2 + s_2 s_1^2 + s_5^2 s_3) + \\
& s_4^2 s_2 + s_2^2 s_1 + s_4 / s_1
\end{aligned}
$$

$$
{}''\mathcal{L}_{C,2}'' = \quad \mathbf{y_1}^2 + \mathbf{y_1} + x_1^3
$$

References

- M. Caboara, T. Mora *"The Chen-Reed-Helleseth-Truong Decoding Algorithm and the Gianni-Kalkbrenner Shape Theorem"* ,J AAECC, 13 (2002).
- A. B.III Cooper, *"Direct solution of BCH decoding equations"* In E.Arikan (Ed.) Communication, Control and Signal Proccessing, 281–286, Elsevier (1990).
- A. B.III Cooper, *"Finding BCH error locator polynomials in one step"* Electronic Letters, 27 (1991) 2090-2091.
- E. Orsini, M. Sala, *"Correcting errors and erasures via the syndrome variety"* Journal of Pure and Applied Algebra, V. 200, p. 191–226, 1 August 2005.
- E. Orsini, *"New decoding algorithm for cyclic codes"*, Proceeding of Miriam Workshop, 2005.
- E. Orsini, M. Sala *"General error locator polynomials for binary cyclic code with $t \leq 2$ and $n < 63$"*, IEEE Trans.on Information Theory, 53 (2007), no.3, 1095-1107.
- T. Mora, E. Orsini *"Decoding cyclic codes. The Cooper philosophy"*, accepted.
- E. Orsini, M. Sala, *"Improved decoding of affine-variety codes"*, accepted Journal of Pure and Applied Algebra.