On the structure of the syndrome variety

Emmanuela Orsini

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

$S^3$CM, 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
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 $\alpha$ 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 $\alpha$ be a primitive $n$-th root of unity in $\mathbb{F}_{q^m}$.

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

$$\nu(x) = \nu_0 + \cdots + \nu_{n-1}x^{n-1} \quad \text{received polynomial}$$

$$e(x) = \nu(x) - c(x) \quad \text{error polynomial}$$
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 $\alpha$ be a primitive $n$-th root of unity in $\mathbb{F}_{q^m}$.

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

$$\nu(x) = \nu_0 + \cdots + \nu_{n-1}x^{n-1} \quad \text{received polynomial}$$

$$e(x) = \nu(x) - c(x) \quad \text{error polynomial}$$

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

$$e = \left(0, \ldots, 0, e_{l_1}, 0, \ldots, 0, e_{l_i}, 0, \ldots, 0, e_{l_\mu}, 0, \ldots, 0\right),$$

where $l_1 = 1$, $l_i = i$, $l_\mu = \mu$, and $l_{n-1-\mu}$.
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 $\alpha$ be a primitive $n$-th root of unity in $\mathbb{F}_{q^m}$.

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

$$\nu(x) = \nu_0 + \cdots + \nu_{n-1}x^{n-1} \quad \text{received polynomial}$$

$$e(x) = \nu(x) - c(x) \quad \text{error polynomial}$$

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

$$e = (0, \ldots, 0, e_{l_1}, 0, \ldots, 0, e_{l_i}, 0, \ldots, 0, e_{l_\mu}, 0, \ldots, 0),$$

$$L = \{l \mid e_l \neq 0, 0 \leq l \leq n-1\} = \{l_1, \ldots, l_\mu\} \quad \text{set of error positions}$$

$$\{e_l \mid l \in L\} \quad \text{set of the error magnitudes}$$
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 $s_j = e(\alpha^j) = \sum_{l \in L} e_l \alpha^{ij} = \sum_{l \in L} e_l (\alpha^l)^j$, $j = 1, \ldots, n - k$.

where

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

\[ Hv^T = H(c^T + e^T) = Hc^T + He^T = 0 + He^T = s^T \]

\[ He^T = \begin{pmatrix}
1 & \alpha_1^i & \alpha_2^i & \cdots & \alpha_{n-1}^i \\
1 & \alpha_2^i & \alpha_2^i & \cdots & \alpha_{n-1}^i \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \alpha_{n-k}^i & \alpha_{n-k}^i & \cdots & \alpha_{n-1}^i_{n-k} \\
\end{pmatrix}
\begin{pmatrix}
e_0 \\
e_1 \\
\vdots \\
e_{n-1} \\
\end{pmatrix} = \begin{pmatrix}
e(\alpha_1^i) \\
e(\alpha_2^i) \\
\vdots \\
e(\alpha_{n-k}^i) \\
\end{pmatrix} \]

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

where \( \{ \alpha^l \mid l \in L \} \) set of the error locations

\[ \sigma(z) = \prod_{l \in L} (1 - z \alpha^l) \quad \text{classical error locator polynomial} \]

\[ L_e(z) = \prod_{l \in L} (z - \alpha^l) \quad \text{plain error locator polynomial} \]
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:

- Chen, Reed, Helleseth, Truong (1994).
- Caboara, Mora (2002).

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 $\bar{s} = (s_1, \ldots, s_{2t-1}) \in (\mathbb{F}_{2^m})^{2t}$ 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.

$$\mathcal{F}_C : \{ f_i = \sum_{j=1}^{t} z_j^{2^i-1} - s_{2i-1}, \quad 1 \leq i \leq t \}$$
Decoding cyclic codes: the Cooper philosophy

Let $C$ be a binary BCH code with

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

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

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

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:

- Chen, Reed, Helleseth, Truong (1994).
- Caboara, Mora (2002).

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

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

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

<table>
<thead>
<tr>
<th>variables</th>
<th>representant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_r$</td>
<td>correctable syndromes</td>
</tr>
<tr>
<td>$z_1, \ldots, z_t$</td>
<td>error locations</td>
</tr>
<tr>
<td>$y_1, \ldots, y_t$</td>
<td>error values</td>
</tr>
</tbody>
</table>
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

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

<table>
<thead>
<tr>
<th>variables</th>
<th>representant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_r$</td>
<td>correctable syndromes</td>
</tr>
<tr>
<td>$z_1, \ldots, z_t$</td>
<td>error locations</td>
</tr>
<tr>
<td>$y_1, \ldots, y_t$</td>
<td>error values</td>
</tr>
</tbody>
</table>
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

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

<table>
<thead>
<tr>
<th>variables</th>
<th>representant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_r$</td>
<td>correctable syndromes</td>
</tr>
<tr>
<td>$z_1, \ldots, z_t$</td>
<td>error locations</td>
</tr>
<tr>
<td>$y_1, \ldots, y_t$</td>
<td>error values</td>
</tr>
</tbody>
</table>
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

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

<table>
<thead>
<tr>
<th>variables</th>
<th>representant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_r$</td>
<td>correctable syndromes</td>
</tr>
<tr>
<td>$z_1, \ldots, z_t$</td>
<td>error locations</td>
</tr>
<tr>
<td>$y_1, \ldots, y_t$</td>
<td>error values</td>
</tr>
</tbody>
</table>

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

We denote by $I$ the ideal

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

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.
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 $\mathcal{Q} := \mathbb{F}_q[x_1, \ldots, x_{n-k}]$.

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. $\mathcal{T}(g_1) < \cdots < \mathcal{T}(g_s)$.

For any $i \leq t$, let $G_i$ be $G \cap (\mathcal{Q}[z_t, \ldots, z_i]) \setminus (\mathcal{Q}[z_t, \ldots, z_{i+1}])$ and

$$\forall \ell \in \mathbb{N}, \ G_{i\ell} := \{g \in G_i \mid \deg_{z_i}(g) = \ell\},$$
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 $i \leq t$, let $G_i$ be $G \cap (\mathcal{Q}[z_t, \ldots, z_i] \setminus \mathcal{Q}[z_t, \ldots, z_{i+1}])$ and

$$\forall \ell \in \mathbb{N}, \ G_{i\ell} := \{g \in G_i \mid \deg_{z_i}(g) = \ell\},$$

so that each $G_i$ can be decomposed into blocks of polynomials according to their degree with respect to the variable $z_i$: $G_i = \bigsqcup \ell G_{i\ell}$. If $g \in G_{i\ell}$:

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

Let $\mathbb{F}_q[x_1, \ldots, x_{n-k}]$. 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. $T(g_1) < \cdots < T(g_s)$.

For any $\iota \leq t$, let $G_\iota$ be $G \cap (\mathbb{F}_q[z_t, \ldots, z_\iota] \setminus \mathbb{F}_q[z_t, \ldots, z_{\iota+1}])$ and:

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

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

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

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

$$G_{\iota\ell} := \{g_{\iota\ell1}, \ldots, g_{\iota\ell j_{\iota\ell}}\}, \ T(g_{\iota\ell1}) < \cdots < T(g_{\iota\ell j_{\iota\ell}}).$$
Gröbner basis structure. (THEOREM)

With the above notation, we have:

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

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

$$Lp(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \iff g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\mu}) \neq 0.$$
Gröbner basis structure. (THEOREM)

With the above notation, we have:

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

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

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

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

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

With the above notation, we have:

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

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

$$Lp(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \iff g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\mu}) \neq 0.$$  
If the error has weight $\mu$, then, for each $g \in G_{\iota\iota}$,

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

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

For each \( g \in G_{\iota \iota} \),

\[ L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \iff g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\mu}) \neq 0. \]

If the error has weight \( \mu \), then, for each \( g \in G_{\iota \iota} \),

1. if \( \iota < \mu \) then \( g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\iota}) = 0 \);
2. if \( \iota = \mu \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[ 0 \neq g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\mu}) = z_{\mu}^\iota L_e(z_{\mu}); \]
3. if \( \iota = \mu + 1 \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[ g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\iota}) = z_{\iota} \cdot (z_{\mu}^{\iota} L_e(z_{\mu})); \]
4. if \( \iota > \mu + 1 \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[ z_{\iota} \cdot (z_{\mu}^{\iota} L_e(z_{\mu})) \mid g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\iota}). \]
Gröbner basis structure. (THEOREM)

With the above notation, we have:

- If \( \ell < \iota \) then \( G_{\ell \iota} = \emptyset \);
- If \( \ell > \iota \) then \( \ell = n + 1, G_{\ell \iota} = \{ z_\iota^{n+1} - z_\iota \} \)

For each \( g \in G_{\ell \iota} \),

\[
L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \iff g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_\mu) \neq 0.
\]

If the error has weight \( \mu \), then, for each \( g \in G_{\ell \iota} \),

1. If \( \ell < \mu \) then \( g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_\iota) = 0 \);
2. If \( \ell = \mu \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[
   0 \neq g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_\mu) = z_\mu^\mu L_e(z_\mu);
   \]
3. If \( \ell = \mu + 1 \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[
   g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_\iota) = z_\iota \cdot (z_\mu^\mu L_e(z_\mu));
   \]
4. If \( \ell > \mu + 1 \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[
   z_\iota \cdot (z_\mu^\mu L_e(z_\mu)) | g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_\iota).
   \]
Gröbner basis structure. (THEOREM)

With the above notation, we have:

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

For each \( g \in G_{\iota\iota}, \)

\[
L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \iff g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\mu}) \neq 0.
\]

If the error has weight \( \mu \), then, for each \( g \in G_{\iota\iota}, \)

1. if \( \iota < \mu \) then \( g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\iota}) = 0; \)
2. if \( \iota = \mu \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[
   0 \neq g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\mu}) = z_{\mu}^L e(z_{\mu});
   \]
3. if \( \iota = \mu + 1 \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[
   g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\iota}) = z_{\iota} \cdot (z_{\mu}^L e(z_{\iota}));
   \]
4. if \( \iota > \mu + 1 \) and \( L_p(g)(s_1, \ldots, s_{n-k}, 0, \ldots, 0) \neq 0 \) then
   \[
   z_{\iota} \cdot (z_{\mu}^L e(z_{\iota})) \mid g(s_1, \ldots, s_{n-k}, 0, \ldots, 0, z_{\iota}).
   \]
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 \n
> 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

\[ J; \]
\[ J[1]=x_1^16+x_1 \]
\[ J[2]=x_3^16+x_3 \]
\[ J[3]=x_5 \cdot x_3^{10}+x_5 \cdot x_3^8 \cdot x_1^6+x_5 \cdot x_3^5+x_5 \cdot x_3^4 \cdot x_1^3+x_5 \cdot x_3^2 \cdot x_1^9+x_5 \cdot x_3^10 \cdot x_1^5+x_3^8 \cdot x_1^{11}+x_3^5 \cdot x_1^5+x_3^4 \cdot x_1^{14}+x_3^3 \cdot x_1^2+x_1^1^5 \]
\[ J[4]=x_5^3+x_5^2 \cdot x_1^5+x_5 \cdot x_3^{10}+x_3^8 \cdot x_1^6+x_3^5+x_3^4 \cdot x_1^3+x_3^2 \cdot x_1^9+x_3^3 \cdot x_1^{12}+x_1^1^5 \]
\[ J[5]=z_3^3 \cdot x_3+z_3^3 \cdot x_1^3+z_3^2 \cdot x_3^2 \cdot x_1^4+z_3^3 \cdot x_5+z_3^3 \cdot x_3^2 \cdot x_1^2 \cdot x_5+x_3^2 \cdot x_1^1^2 \]
\[ J[6]=z_3^3 \cdot x_5+z_3^3 \cdot x_1^5+z_3^2 \cdot x_5^2 \cdot x_1^3+z_3^3 \cdot x_3^9 \cdot x_1^{10}+z_3^3 \cdot x_3^8 \cdot x_1^{13}+z_3^3 \cdot x_3^4 \cdot x_1^1^1 \]
\[ J[7]=z_3 \cdot x_3^1^6+z_3 \]
\[ J[8]=z_2^2 \cdot x_3+z_2^2 \cdot x_1^3+z_2^3 \cdot x_3^2 \cdot x_1^3+z_2^2 \cdot x_3 \cdot x_1^1^4+z_2^3 \cdot x_3^2 \cdot x_1^1^2 \cdot x_3^3 \cdot x_1^{1^2} \]
\[ J[9]=z_2^2 \cdot x_5+z_2^2 \cdot x_1^5+z_2^2 \cdot x_3^5 \cdot x_3^5 \cdot x_1^1^5+z_2^2 \cdot x_5 \cdot x_1^6+z_3^2 \cdot x_5^2 \cdot x_3^1^3 \]
\[ J[10]=z_2^2 \cdot x_3^9 \cdot x_3^1^6+z_2^2 \cdot x_3^8 \cdot x_3^9 \cdot x_3^4 \cdot x_3^8 \cdot x_3^4 \cdot x_3^1^4+z_2^3 \cdot x_3^2 \cdot x_3^1^9+x_5^2 \cdot x_3^1^4 \cdot x_3^2 \cdot x_3^1^1 \]
\[ J[11]=z_2^2 \cdot z_3^2 \cdot x_3^1^6+z_2 \]
\[ J[12]=z_1+z_2+z_3+x_1 \]
Example

\[ J[1] = x_1^16 + x_1 \]
\[ J[2] = x_3^16 + x_3 \]
\[ J[3] = x_5 \cdot x_3^10 + x_5 \cdot x_3^8 \cdot x_1^6 + x_5 \cdot x_3^5 + x_5 \cdot x_3^4 \cdot x_1^3 + x_5 \cdot x_3^2 \cdot x_1^9 + x_5 \cdot x_3 \cdot x_1^{12} + x_5 + x_3^{10} \cdot x_1^{15} + x_3^8 \cdot x_1^{11} + x_3^5 \cdot x_1^{15} + x_3^4 \cdot x_1^8 + x_3^2 \cdot x_1^{14} + x_3 \cdot x_1^{12} \]
\[ J[4] = x_5^3 + x_5^2 \cdot x_1^5 + x_5 \cdot x_1^{10} + x_3^{10} + x_3^8 \cdot x_1^6 + x_3^5 + x_3^4 \cdot x_1^3 + x_3^2 \cdot x_1^9 + x_3 \cdot x_1^{12} + x_1^{15} \]
\[ J[5] = z_3^3 \cdot x_3 + z_3^3 \cdot x_1^3 + z_3^2 \cdot x_3 \cdot x_1 + z_3^2 \cdot x_1^4 + z_3 \cdot x_3 \cdot x_1^2 + x_5 \cdot x_3^2 + x_3^2 + x_1^{13} \]
\[ J[6] = z_3^3 \cdot x_3^5 + z_3^3 \cdot x_1^5 + z_3^2 \cdot x_3 \cdot x_1^6 + z_3 \cdot x_3^5 + z_3 \cdot x_1^{10} + z_3 \cdot x_3^8 \cdot x_1^{13} + z_3 \cdot x_3^4 \cdot x_1^{11} \]
\[ J[7] = z_3^3 \cdot x_3^10 + z_3 \cdot x_3^8 \cdot x_1^6 + z_3 \cdot x_3^5 + z_3 \cdot x_3^4 \cdot x_1^3 + z_3 \cdot x_3^2 \cdot x_1^9 + z_3 \cdot x_3 \cdot x_1^{12} + x_3 \cdot x_1^{11} + x_3^8 \cdot x_1^{14} + x_3^4 \cdot x_1^{11} \]
\[ J[8] = z_2^2 \cdot x_3^3 + z_2^2 \cdot x_1^3 + z_2 \cdot z_3 \cdot x_3^3 \cdot x_1^{13} + z_2 \cdot z_3 \cdot x_3^2 \cdot x_1^{14} + z_2 \cdot x_3^2 \cdot x_1^4 + z_3 \cdot x_3^2 \cdot x_3^2 \cdot x_1^2 \]
\[ J[9] = z_2^2 \cdot x_3^5 + z_2^2 \cdot x_1^5 + z_2 \cdot z_3 \cdot x_3^5 + z_2 \cdot z_3 \cdot x_3^4 \cdot x_1^6 + z_3 \cdot x_3^2 \cdot x_5 + z_3^{10} \cdot x_1^{15} \]
\[ J[10] = z_2^2 \cdot z_3^3 + z_2^2 \cdot x_1^3 + z_2 \cdot z_3^2 + z_2 \cdot x_1^2 + z_3^2 \cdot x_1^2 + z_3 \cdot x_1^{12} + x_3 + x_1^7 \]
\[ J[11] = z_2 \cdot x_1^{16} + z_2 \]
\[ J[12] = z_1 + z_2 + z_3 + x_1 \]
Example

\[ g_{3,3,1} = z_3^3(x_3 + x_1^3) + z_3^2x_3x_1 + z_3^2x_1^4 + z_3x_5 + z_3x_3x_1^2 + x_5x_1 + x_3^2 + x_3x_1^3 + x_1^6 \]
\[ g_{3,3,2} = z_3^3(x_5 + x_1^5) + z_3^2x_5x_1 + z_3^2x_1^6 + z_3x_5x_3 + z_3^2x_3^2x_1 + z_3x_5x_3^2 + z_3x_5x_3x_1^2 + z_3x_5x_1^2 + z_3x_3x_1^{10} + z_3x_3x_1^{13} + z_3x_3x_1^4 + z_3x_3x_1^7 + x_5^2x_3 + x_5x_3x_1 + x_5x_3x_1^2 + x_5x_3x_1^4 + x_5x_3x_1^6 + x_5x_3x_1^9 + x_5x_3x_1^{11} + x_5x_3x_1^{14} + x_3x_1^4 \]
\[ g_{3,16,1} = z_3^{16} + z_3 \]
\[ g_{2,1,1} = z_2^2(x_3 + x_1^3) + z_2x_3x_3 + z_2x_3x_1 + z_2x_1^4 + z_2x_3 + z_2^2x_3 + z_2x_3x_1 + z_2x_3x_1^2 + z_2x_3x_1^4 + x_5 + x_3x_1^2 \]
\[ g_{2,2,2} = z_2^2(x_5 + x_1^5) + z_2x_5x_3 + z_2x_3x_1^5 + z_2x_5x_1 + z_2x_1^6 + z_2x_3^2x_5 + z_2x_3x_5 + z_3x_5x_1 + z_3x_1^6 + x_5x_3 + x_5x_3x_1^3 + x_5x_3x_1^6 + x_5x_3x_1^9 + x_5x_3x_1^{12} + x_5x_3x_1^{15} + x_3x_1^4 + x_1^7 \]
\[ g_{2,2,3} = z_2^2(x_3 + x_1) + z_2x_3^2 + z_2x_1^2 + z_3x_1^2 + x_3 + x_1^3 \]
\[ g_{2,16,1} = z_2^{16} + z_2 \]
\[ g_{1,1,1} = z_1 + z_2 + z_3 + x_1 \]

\[ G_3 = \{ G_{3,3}, G_{3,16} \} \quad G_{3,3} = \{ g_{3,3,1}, g_{3,3,2} \}, \quad G_{3,16} = \{ g_{3,16,1} \} \]
\[ G_2 = \{ G_{2,2}, G_{2,16} \} \quad G_{2,2} = \{ g_{2,2,1}, g_{2,2,2}, g_{2,2,3} \}, \quad G_{2,16} = \{ g_{2,16,1} \} \]
\[ G_1 = \{ G_{1,1} \} \quad G_{1,1} = \{ g_{1,1,1} \} \]
**Decoding algorithm**

**Input** $\mu := t, g := 1,$

**Repeat**

1. $j := 0$
2. **Repeat** $j := j + 1$
3. **Until** $L_p(g_{\mu j})(s, 0) \neq 0$ or $j > j_{\mu \mu}$
4. if $j > j_{\mu \mu}$ then $\mu := \mu - 1$ else
5. if $T_p(g_{\mu j})(s, 0) = 0$ do $\mu := \mu - 1$
6. else $g(z) := g_{\mu j}(s, 0, z)$
7. **Until** $g \neq 1$ or $\mu = 0$

**Output** $\mu, x^\mu g(x^{-1})$

**Table:** Decoding algorithm
Remark

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

$$(z_1, \ldots, z_\mu, 0, \ldots, 0, y_1, \ldots, y_\mu, y_1, \ldots, y_{t-\mu}),$$

where $\bar{y}_j$ 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 $\mu \leq t - 2$

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

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
Let $C$ be an $[n, k, d]_q$ linear code and $t$ its correction capability. Let $d \geq 3$ and $(n, q) = 1$. Let $\alpha$ 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 $s = (\overline{s}_1, \ldots, \overline{s}_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$, 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 $s$, then the roots of $\mathcal{L}(s, z)$ are exactly $\{\alpha^{l_1}, \ldots, \alpha^{l_\mu}, 0, \ldots, 0\}$.
General error locator polynomial

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 $\alpha$ 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 $s = (s_1, \ldots, s_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$, 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 $s$, then the roots of $\mathcal{L}(s, z)$ are exactly $\{\alpha^{l_1}, \ldots, \alpha^{l_\mu}, 0, \ldots, 0\}$.
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 $\alpha$ 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 $s = (\bar{s}_1, \ldots, \bar{s}_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$, 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 $s$, then the roots of $\mathcal{L}(s, z)$ are exactly $\{\alpha^{l_1}, \ldots, \alpha^{l_\mu}, 0, \ldots, 0\}$.
Let $C$ be an $[n, k, d]_q$ linear code and $t$ its correction capability. Let $d \geq 3$ and $(n, q) = 1$. Let $\alpha$ be a primitive $n$-th root of unity in $\mathbb{F}_{q^m}$.

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

1. $L(S, z) = z^t + a_{t-1}z^{t-1} + \cdots + a_0$, with $a_j \in \mathbb{F}_q[S]$, $0 \leq j \leq t - 1$, that is, $L$ 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 $s = (\bar{s}_1, \ldots, \bar{s}_{n-k}) \in (\mathbb{F}_{q^m})^{n-k}$, 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 $s$, then the roots of $L(s, z)$ are exactly $\{\alpha^{l_1}, \ldots, \alpha^{l_\mu}, 0, \ldots, 0\}$. 

$\alpha^{t-\mu}$
Stratified ideals

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

$$S = (s_1, \ldots, s_H), \quad A = (a_1, \ldots, a_L), \quad T = (t_1, \ldots, t_K).$$
Stratified ideals

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

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

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

$$a_1 > a_2 > \cdots > a_L$$
Stratified ideals

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

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

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

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

We use the usual notation for the elimination ideals:

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

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

$$\vdots$$

$$J_{S, A} = J_{S, a_L, \ldots, a_1} = J \cap \mathbb{K}[S, a_L, \ldots, a_1] = J \cap \mathbb{K}[S, A]$$
Properties of stratified ideals

Stratified ideals

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

\[
\Sigma_{j}^{h-1} = \{(\bar{s}_1, \ldots, \bar{s}_N, \bar{a}_L, \ldots, \bar{a}_h) \in \mathcal{V}(J_S, a_L, \ldots, a_h) \mid \exists \text{ exactly } j \text{ distinct values } \\
\{\bar{a}_{h-1}^{(1)}, \ldots, \bar{a}_{h-1}^{(j)}\}, \text{s.t. } (\bar{s}_1, \ldots, \bar{s}_N, \bar{a}_L, \ldots, \bar{a}_h, \bar{a}_{h-1}^{(i)}) \in \mathcal{V}(J_S, a_L, \ldots, a_{h-1}), \\
1 \leq i \leq j\}. 
\]
Properties of stratified ideals

Stratified ideals

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

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

Then it holds:

- \[ \mathcal{V}(J_S) = \bigsqcup_{j=1}^{\lambda(L)} \Sigma_j^L \]
- \[ \mathcal{V}(J_S, a_L, \ldots, a_h) = \bigsqcup_{j=1}^{\lambda(h-1)} \Sigma_j^{h-1} , \quad 2 \leq h \leq L. \]

For any arbitrary zero-dimensional ideal \( J \) nothing can be said about \( \lambda(h) \), except that \( \lambda(h) \geq 1 \) for any \( 2 \leq h \leq L \).
Stratified ideals

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

1. $\lambda(h) = h$, $1 \leq h \leq L$, (the number of distinct extensions is at most $h$ for any point in $\mathcal{V}(J_{S,a_L,...,a_h})$) 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 $A$ variables.
Stratified ideals. Example

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

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

Let $J = J(Z) \subset \mathbb{C}[s_1, a_3, a_2, a_1, t_1]$ 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:

$$V(J_S) = \{1, 2, 3, 5\}$$
$$V(J_S, a_3) = \{(1, 2), (1, 4), (1, 6), (2, 5), (3, 1), (3, 3), (5, 2)\}$$
$$V(J_S, a_3, a_2) = \{(1, 2, 1), (1, 2, 2)(1, 4, 0), (1, 6, 0), (2, 5, 0), (3, 1, 0), (3, 3, 0), (5, 2, 0)\}$$
$$V(J_S, a_3, a_2, a_1) = \{(1, 2, 1, 0), (1, 2, 2, 0)(1, 4, 0, 0), (1, 6, 0, 0), (2, 5, 0, 0), (3, 1, 0, 0), (3, 3, 0, 0), (5, 2, 0, 0)\}$$
Let \( S = \{s_1\} \), \( A = \{a_1, a_2, a_3\} \) (\( L = 3 \)) and \( T = \{t_1\} \) s.t. \( a_1 > a_2 > a_3 \).

Let \( J = J(Z) \subset \mathbb{C}[s_1, a_3, a_2, a_1, t_1] \) 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:

\[
V(J_S) = \{1, 2, 3, 5\}
\]
\[
V(J_{S,a_3}) = \{(1, 2), (1, 4), (1, 6), (2, 5), (3, 1), (3, 3), (5, 2)\}
\]
\[
V(J_{S,a_3,a_2}) = \{(1, 2, 1), (1, 2, 2)(1, 4, 0), (1, 6, 0), (2, 5, 0), (3, 1, 0), (3, 3, 0), (5, 2, 0)\}
\]
\[
V(J_{S,a_3,a_2,a_1}) = \{(1, 2, 1, 0), (1, 2, 2, 0)(1, 4, 0, 0), (1, 6, 0, 0), (2, 5, 0, 0), (3, 1, 0, 0), (3, 3, 0, 0), (5, 2, 0, 0)\}
\]

Let us consider the projection \( \pi : V(J_{S,a_3}) \rightarrow V(J_S) \). Then:

\[
|\pi^{-1}(\{5\})| = 1, \quad |\pi^{-1}(\{2\})| = 1, \quad |\pi^{-1}(\{3\})| = 2, \quad |\pi^{-1}(\{1\})| = 3
\]

so

\[
\sum^3_1 = \{2, 5\}, \quad \sum^3_2 = \{3\}, \quad \sum^3_3 = \{1\} \text{ and } \sum^3_i = \emptyset, \quad i > 3.
\]
Stratified ideals. Example

Figure: $\mathscr{V}(J_{S,a_1})$ in a stratified case
Let $G$ be a reduced Gröbner basis of $J$ w.r.t. $>$. The elements of $G \cap (K[S, a_L, \ldots, a_1] \setminus K[S])$ can be collected into non-empty blocks $\{G_i\}_{1 \leq i \leq L}$ and each $\{G_i\}$ can be decomposed into blocks of polynomials according to their degree with respect to the variable $a_i$: 

$$G_i = \bigsqcup \ell G_{i\ell}.$$
Structure theorem

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

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

**Proposition**

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

- $G_i = \bigsqcup_{\delta=1}^{i} G_{i\delta}$ 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 $G_i$;
- $T(g_{ii1}) = a_i^i$. 


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

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

**Proposition**

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

- $G_i = \bigsqcup_{\delta=1}^{i} G_{i\delta}$ 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 $G_i$;
- $T(g_{ii1}) = a_i^i$. 

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

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

**Proposition**

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

- $G_i = \bigsqcup_{\delta=1}^i G_{i\delta}$ 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 $G_i$;
- $T(g_{ii1}) = a_i^i$. 

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

**Proposition**

**Let \( J \) be a stratified ideal w.r.t. the \( A \) variable.** Let \( G \) be a reduced Gröbner basis of \( J \) w.r.t. \( > \). Then

- \( G_i = \bigcup_{\delta=1}^i G_{i\delta} \) 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 \( G_i \);
- \( T(g_{ii1}) = a_i^i \).
A new syndrome variety

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.
A new syndrome variety

**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 \(p(n, z_l, z_\tilde{l}) \in \mathbb{F}_q[z_1, \ldots, z_t]\) the polynomial:

\[
p(n, z_l, z_\tilde{l}) = \frac{z_l^n - z_{\tilde{l}}^n}{z_l - z_{\tilde{l}}}, \quad 1 \leq l < \tilde{l} \leq t.
\]
A new syndrome variety

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 \(p(n, z_l, z_{\tilde{l}}) \in \mathbb{F}_q[z_1, \ldots, z_t]\) the polynomial:

\[
p(n, z_l, z_{\tilde{l}}) = \frac{z_l^n - z_{\tilde{l}}^n}{z_l - z_{\tilde{l}}}, \quad 1 \leq l < \tilde{l} \leq t.
\]

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

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

\(V(I')\) is a new syndrome variety.
General error locator polynomial for cyclic codes

These polynomials remove all the spurious 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 $\mu$ errors;
- $a_{t-l}(s) = 0$ for $l > \mu$ and $a_{t-\mu}(s) \neq 0$;
- $g(s, z_t) = z^{t-\mu} (L_e(z))$;
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 \( \mu \) errors;
- \( a_{t-l}(s) = 0 \) for \( l > \mu \) and \( a_{t-\mu}(s) \neq 0 \);
- \( g(s, z_t) = z^{t-\mu}(L_e(z)) \);
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 $\mu$ errors;
- $a_{t-l}(s) = 0$ for $l > \mu$ and $a_{t-\mu}(s) \neq 0$;
- $g(s, z_t) = z^{t-\mu}(L_e(z))$;
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 $\mu$ 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 $\sigma(z) = z^{\mu} g(s, z^{-1})$. 
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 $\mu$ errors;
- $a_{t-l}(s) = 0$ for $l > \mu$ and $a_{t-\mu}(s) \neq 0$;
- $g(s, z_t) = z_t^{t-\mu} (L_e(z))$;

and imply that $\sigma(z) = z^{\mu} g(s, z^{-1})$. This means that $g$ is a monic polynomial in $\mathbb{Q}[z]$ 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 $\mu$ error locations plus zero counted with multiplicity $t - \mu$,

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

A new syndrome variety
Once we have computed a general error locator polynomial for the code $C$, the decoding algorithm is straightforward:

<table>
<thead>
<tr>
<th>Input</th>
<th>$s = (s_1, \ldots, s_{n-k})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu = t$</td>
<td></td>
</tr>
<tr>
<td><strong>While</strong></td>
<td>$a_{t-\mu}(s_1, \ldots, s_{n-k}) = 0$ <strong>do</strong></td>
</tr>
<tr>
<td>$\mu := \mu - 1$;</td>
<td></td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>$\mu, L_e(z)$</td>
</tr>
</tbody>
</table>

**Table:** Decoding algorithm
A new 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.
SINGULAR /
A Computer Algebra System for Polynomial Computations / version 3-0-4
FB Mathematik der Universitaet, D-67653 Kaiserslautern \n
> 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); }
. 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
A new decoding algorithm

Example

\[ J[1] = x_1^{16} + x_1 \]
\[ J[2] = x_3^{16} + x_3 \]
\[ J[3] = x_5 x_3^{10} + x_5 x_3^8 x_1^6 + x_5 x_3^5 + x_5 x_3^4 x_1^3 + x_5 x_3^2 x_1^9 + x_5 x_3^3 x_1^{12} + x_5 + x_3^{10} x_1^5 + x_3^8 x_1^1 + x_3^6 x_1^{12} + x_3^5 x_1^{15} + x_3^4 x_1^3 + x_3^2 x_1^9 + x_3 x_1^{12} + x_1^{15} \]
\[ J[4] = x_5^{3} + x_5^{2} x_1^6 + x_3^{10} x_1^{12} + x_3^8 x_1^6 + x_3^5 x_3^4 x_1^3 + x_3^2 x_1^{12} + x_3 x_1^{12} + x_1^{15} \]
\[ J[5] = z_3 (x_3^{15} x_1^{15} + x_3^{13} x_1^{15} + x_3^{12} x_1^9 + x_3^{11} x_1^{12} + x_3^{10} x_1^{15} + x_3^9 x_1^{13} + x_3^8 x_1^{16} + x_3^7 x_1^{19} + x_3^6 x_1^{12} + x_3^5 x_1^{15} + x_3^4 x_1^{12} + x_3^3 x_1^{15} + x_3^2 x_1^{12} + x_3 x_1^{15} + 1) \]
\[ J[6] = z_3^2 (x_3^{15} x_1^{15} + x_3^{14} x_1^{13} + x_3^{13} x_1^{16} + x_3^{12} x_1^9 + x_3^{11} x_1^{12} + x_3^{10} x_1^{15} + x_3^9 x_1^{13} + x_3^8 x_1^{16} + x_3^7 x_1^{19} + x_3^6 x_1^{12} + x_3^5 x_1^{15} + x_3^4 x_1^{12} + x_3^3 x_1^{15} + x_3^2 x_1^{12} + x_3 x_1^{15} + 1) \]
\[ J[7] = z_3^3 + z_3^2 x_1 + z_3 (x_5 x_3^9 + x_5 x_3^8 x_1^3 + x_5 x_3^4 x_1^9 + x_3^{15} x_1^{15} + x_3^{14} x_1^{12} + x_3^{13} x_1^{15} + x_3^{12} x_1^9 + x_3^{11} x_1^{12} + x_3^{10} x_1^{15} + x_3^9 x_1^{13} + x_3^8 x_1^{16} + x_3^7 x_1^{19} + x_3^6 x_1^{12} + x_3^5 x_1^{15} + x_3^4 x_1^{12} + x_3^3 x_1^{15} + x_3^2 x_1^{12} + x_3 x_1^{15} + 1) \]
\[ J[8] = z_2 (x_3^{15} x_1^{15} + x_3^{13} x_1^{15} + x_3^{12} x_1^9 + x_3^{11} x_1^{12} + x_3^{10} x_1^{15} + z_3 x_3^{13} x_1^{15} + z_3 x_3^{12} x_1^9 + z_3 x_3^{11} x_1^{12} + z_3 x_3^{10} x_1^{15} + z_3 x_3^{9} x_1^{13} + z_3 x_3^{8} x_1^{16} + z_3 x_3^{7} x_1^{19} + z_3 x_3^{6} x_1^{12} + z_3 x_3^{5} x_1^{15} + z_3 x_3^{4} x_1^{12} + z_3 x_3^{3} x_1^{15} + z_3 x_3^{2} x_1^{12} + z_3 x_3^{1} x_1^{15} + z_3) \]
\[ J[9] = z_2 x_3^{14} x_1^{15} + z_3 x_3^{13} x_1^{15} + z_3 x_3^{12} x_1^9 + z_3 x_3^{11} x_1^{12} + z_3 x_3^{10} x_1^{15} + z_3 x_3^{9} x_1^{13} + z_3 x_3^{8} x_1^{16} + z_3 x_3^{7} x_1^{19} + z_3 x_3^{6} x_1^{12} + z_3 x_3^{5} x_1^{15} + z_3 x_3^{4} x_1^{12} + z_3 x_3^{3} x_1^{15} + z_3 x_3^{2} x_1^{12} + z_3 x_3^{1} x_1^{15} + z_3 \]
\[ J[10] = z_2^2 + z_2 (z_3 x_1 + z_3 x_1^{15} + z_3 x_1^{12} + z_3 x_1^{11} + z_3 x_1^{10} + z_3 x_1^{9} + z_3 x_1^{8} + z_3 x_1^{7} + z_3 x_1^{6} + z_3 x_1^{5} + z_3 x_1^{4} + z_3 x_1^{3} + z_3 x_1^{2} + z_3 x_1 + 1) \]
\[ J[11] = z_1 + z_2 + z_3 + x_1 \]
Example

\[ J[1] = x_1^{16} + x_1 \]
\[ J[2] = x_3^{16} + x_3 \]
\[ J[3] = x_5^3 + x_3^2 + x_1 + z_3(1) \]
\[ J[4] = x_5^3 + x_3^2 + x_1 + z_3(2) \]
\[ J[5] = x_5^3 + x_3^2 + x_1 + z_3(3) \]
\[ J[6] = x_5^3 + x_3^2 + x_1 + z_3(4) \]
\[ J[7] = x_5^3 + x_3^2 + x_1 + z_3(5) \]
\[ J[8] = x_5^3 + x_3^2 + x_1 + z_3(6) \]
\[ J[9] = x_5^3 + x_3^2 + x_1 + z_3(7) \]
\[ J[10] = x_5^3 + x_3^2 + x_1 + z_3(8) \]
\[ J[11] = x_5^3 + x_3^2 + x_1 + z_3(9) \]

A new decoding algorithm
Example

\[ g_{3,1,1} = z_3(x_3^{15} x_1^{15} + x_3^{15} + x_1^{15} + 1) \]
\[ g_{3,2,1} = z_2(x_3^{15} + x_3^{14} x_1^{3} + x_3^{13} x_1^{6} + x_3^{12} x_1^{9} + x_3^{11} x_1^{12} + x_3^{10} x_1^{15} + x_3^{9} x_1^{3} + x_3^{8} x_1^{6} + x_3^{7} x_1^{9} + x_3^{6} x_1^{12} + x_3^{5} x_1^{15} + x_3^{4} x_1^{3} + x_3^{3} x_1^{6} + x_3^{2} x_1^{9} + x_3 x_1^{12} + x_1^{15} + 1) + z_3(x_3^{15} x_1 + x_3^{14} x_1^{3} + x_3^{13} x_1^{6} + x_3^{12} x_1^{9} + x_3^{11} x_1^{12} + x_3^{10} x_1^{15} + x_3^{9} x_1^{3} + x_3^{8} x_1^{6} + x_3^{7} x_1^{9} + x_3^{6} x_1^{12} + x_3^{5} x_1^{15} + x_3^{4} x_1^{3} + x_3^{3} x_1^{6} + x_3^{2} x_1^{9} + x_3 x_1^{12} + x_1^{15} + 1) \]
\[ g_{3,3,1} = z_3^2 + z_3 x_1 + z_3(x_5 x_3^9 + x_5 x_3^8 x_1^3 + x_5 x_3^4 + x_5 x_3^9 + x_3^{15} x_1^2 + x_3^{14} x_1^5 + x_3^{13} x_1^8 + x_3^{12} x_1^{11} + x_3^{11} x_1^{14} + x_3^{10} x_1^2 + x_3^7 x_1^{11} + x_3^6 x_1^{14} + x_3^{5} x_1^{8} + x_3^{3} x_1^{11} + x_1^2) + x_5 x_3^9 x_1 + x_5 x_3^8 x_1^4 + x_5 x_3^4 x_1 + x_5 x_3^3 x_1^10 + x_3^{15} x_1^3 + x_3^{14} x_1^6 + x_3^{13} x_1^9 + x_3^{12} x_1^{12} + x_3^{11} x_1^{15} + x_3^{10} x_1^3 + x_3^{9} x_1^6 + x_3^{8} x_1^9 + x_3^{7} x_1^{12} + x_3^{6} x_1^{15} + x_3^{5} x_1^3 + x_3^{4} x_1^6 + x_3^{3} x_1^{9} + x_3^{2} x_1^{12} + x_3 x_1^{15} + x_1^2 \]
\[ g_{2,1,1} = z_2(x_3^{15} x_1^{15} + x_3^{15} + x_1^{15} + 1) \]
\[ g_{2,1,2} = z_2(z_3 x_3^9 + z_3 x_3^8 x_1^3 + z_3 x_3^{13} x_1^6 + z_3 x_3^{12} x_1^9 + z_3 x_3^{11} x_1^{12} + z_3 x_3^{10} x_1^{15} + z_2 z_3 x_3^9 x_1^3 + z_3 x_3^8 x_1^6 + z_3 x_3^7 x_1^9 + z_3 x_3^{12} x_1^{15} + z_3 x_3^6 x_1^{15} + z_3 x_3^5 x_1^{12} + z_3 x_3^4 x_1^9 + z_3 x_3^3 x_1^6 + z_3 x_3^2 x_1^3 + z_3 x_1^6 + z_3 x_1^3 + z_3 x_1^{12} + z_3 x_1^{15} + z_3) \]
\[ g_{2,2,1} = z_2^2 + z_2(z_3 + x_1) + z_2^2 + z_3 x_1 + x_5 x_3^9 + x_5 x_3^8 x_1^3 + x_5 x_3^4 x_1 + x_5 x_3^{15} x_1^2 + x_5 x_3^{14} x_1^5 + x_3^{13} x_1^8 + x_3^{12} x_1^9 + x_3^{11} x_1^{12} + x_3^{10} x_1^{15} + x_3^{9} x_1^{3} + x_3^{8} x_1^{6} + x_3^{7} x_1^{9} + x_3^{6} x_1^{12} + x_3^{5} x_1^{15} + x_3^{4} x_1^{3} + x_3^{3} x_1^{6} + x_3^{2} x_1^{9} + x_3 x_1^{12} + x_1^2 \]
\[ g_{1,1,1} = z_1 + z_2 + z_3 + x_1 \]

\[ G_3 = \{ G_{3,3}, G_{3,2}, G_{3,1} \} \]
\[ G_{3,3} = \{ g_{3,3,1} \}, G_{3,2} = \{ g_{3,2,1} \}, G_{3,1} = \{ g_{3,1,1} \} \]
\[ G_2 = \{ G_{2,2}, G_{2,1} \} \]
\[ G_{2,2} = \{ g_{2,2,1} \}, G_{2,1} = \{ g_{2,1,1}, g_{2,1,2} \} \]
\[ G_1 = \{ G_{1,1} \} \]
\[ G_{1,1} = \{ g_{1,1,1} \} \]
A new decoding algorithm

Example

\[ g_{3,3,1} = z_3^3 + z_3^2 x_1 + z_3 (x_5 x_3^9 + x_5 x_3^8 x_1^3 + x_5 x_3^4 + x_5 x_3 x_1^9 + x_3^15 x_1 + x_3^{14} x_1 + x_3^{13} x_1 + x_3^{12} x_1 + x_3^{11} x_1 + x_3^{10} x_1 + x_3^9 x_1 + x_3^8 x_1 + x_3^7 x_1 + x_3^6 x_1 + x_3^5 x_1 + x_3^4 x_1 + x_3^3 x_1 + x_3^2 x_1 + x_3 x_1 + x_3 + 1) \]

\[ g_{3,3,1} = z_3^3 + a^{13} z_3^2 + a^9 z_3 + a^5 x_3^{10} x_1 + a^{14} x_3^8 x_1 + a^{13} x_3^7 x_1 + a^{12} x_3^6 x_1 + a^{11} x_3^5 x_1 + a^{10} x_3^4 x_1 + a^9 x_3^3 x_1 + a^8 x_3^2 x_1 + a^7 x_3 x_1 + a^6 x_1 + a^5 + a^4 + a^3 + a^2 + a + 1 \]
Example

$$g_{3,3,1} = z_3^3 + z_3^2 x_1 + z_3 (x_5 x_3^9 + x_5 x_3^8 x_1^3 + x_5 x_3^4 + x_5 x_3^x_1^9 + x_3 x_1^{15} + x_3 x_1^{14} + x_3 x_1^{13} x_1^8 + x_3 x_1^{12} x_1^{11} + x_3 x_1^{11} x_1^{14} +$$

$$x_3 x_1^{10} + x_3 x_1^{7} + x_3 x_1^{4} + x_3 x_1^{3} + x_3 x_1^{2} + x_3 x_1^{1} + x_1^{2}) + x_5 x_3 x_1 + x_5 x_3^8 x_1 + x_5 x_3^4 x_1 + x_5 x_3^{10} + x_3 x_1^{15} x_1^3 + x_3 x_1^{14} x_1^6 +$$

$$x_3 x_1^{13} x_1^9 + x_3 x_1^{12} x_1^{12} + x_3 x_1^{11} x_1^{15} + x_3 x_1^{10} x_1^{3} + x_3 x_1^{7} x_1^{12} + x_3 x_1^{6} x_1^{15} + x_3 x_1^{5} x_1^{3} + x_3 x_1^{3} x_1^9 + x_3 x_1^{12} + x_3$$

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

$$x_1 = \alpha^{13} \quad x_3 = \alpha^{10} \quad x_5 = \alpha^{10}$$
A new decoding algorithm

Example

\[ g_{3,3,1} = z_3^3 + z_3^2 x_1 + z_3(x_5 x_3^9 + x_5 x_3 x_1^3 + x_5 x_1^4 + x_5 x_3 x_1^9 + x_3 x_1^2 + x_3 x_1^5 + x_3 x_1^3 x_1^8 + x_3 x_1^{12} x_1^{11} + x_3 x_1^{11} x_1^{14} + x_3 x_1^{10} x_1^2 + x_3 x_1^7 x_1^{11} + x_3 x_1^6 x_1^{14} + x_3 x_1^5 x_1^2 + x_3 x_1^3 x_1^{11} + x_3 x_1^2 x_1^{11} + x_1^2) + x_5 x_3 x_1 + x_5 x_3 x_1^4 + x_5 x_3 x_1^3 x_1^10 + x_3 x_1^3 x_1^{15} + x_3 x_1^{14} x_1^6 + x_3 x_1^3 x_1^{9} + x_3 x_1^2 x_1^{12} + x_3 x_1 x_1^{15} + x_3 x_1^{10} x_1^3 + x_3 x_1^{7} x_1^{12} + x_3 x_1^{6} x_1^{15} + x_3 x_1^3 x_1^{3} + x_3 x_1^2 x_1^{9} + x_3 x_1 x_1^{2} x_1^{12} + x_3 \]

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
   \[ x_1 = \alpha^{13} \quad x_3 = \alpha^{10} \quad x_5 = \alpha^{10} \]

\[
> \text{subst(subst(subst(g,x_1,a^13),x_3,a^10),x_5,a^10));}
\]
\[
z_3^3+\text{a}^{13}\times z_3^2+\text{a}^9\times z_3+\text{a}^5
\]
\[
> \text{poly gs= z_3^3+a^13*z_3^2+a^9*z_3+a^5;}
\]
\[
> \text{subst(gs,z_3,1);}
0
\]
\[
> \text{subst(gs,z_3,a);}
\text{a}^3
\]
\[
> \text{subst(gs,z_3,a^2);}
0
\]
\[
> \text{subst(gs,z_3,a^3);}
0
\]
Example

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

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

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

```plaintext
> 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 \( c = (0, 0, \ldots, 0) \) is the transmitted word.
   Let \( v = (0, 1, 0, 0, 0, \ldots, 0) \) be the received vector, then \( \mu = 1 \) and
   \[
   x_1 = \alpha \quad x_3 = \alpha^3 \quad x_5 = \alpha^5
   \]
We suppose the \( c = (0, 0, \ldots, 0) \) is the transmitted word. Let \( v = (0, 1, 0, 0, 0, \ldots, 0) \) be the received vector, then \( \mu = 1 \) and

\[
\begin{align*}
    x_1 &= \alpha \\
    x_3 &= \alpha^3 \\
    x_5 &= \alpha^5
\end{align*}
\]

> subst(subst(subst(g,x_1,a),x_3,a^3),x_5,a^5);

\[
z_3^3 + a \cdot z_3^2
\]

> 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
Remark 1

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.
Remark 1

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.
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.
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.

Remark 2

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

Let $C$ be an $[n, k, d]_q$ cyclic code with defining set $S_C = \{i_1, \ldots, i_{n-k}\}$. Let $\tau$ be the number of errors, $\nu$ 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.
Remark 2

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

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

We denote by $\{\alpha^l | 1 \leq l \leq \tau\}$ the set of the error locations and by $\{\alpha^h | 1 \leq h \leq \nu\}$ 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.

<table>
<thead>
<tr>
<th>variables</th>
<th>representant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_{n-k}$</td>
<td>correctable syndromes</td>
</tr>
<tr>
<td>$z_1, \ldots, z_\tau$</td>
<td>error locations</td>
</tr>
<tr>
<td>$y_1, \ldots, y_\tau$</td>
<td>error values</td>
</tr>
<tr>
<td>$w_1, \ldots, w_\nu$</td>
<td>erasure locations</td>
</tr>
<tr>
<td>$u_1, \ldots, u_\nu$</td>
<td>erasure values</td>
</tr>
</tbody>
</table>
Let $C$ be an $[n, k, d]_q$ cyclic code with defining set $S_C = \{i_1, \ldots, i_{n-k}\}$. Let $\tau$ be the number of errors, $\nu$ be the number of erasures s.t. $2\tau + \nu < d$. We denote by $\{\alpha^l | 1 \leq l \leq \tau\}$ the set of the error locations and by $\{\alpha^h | 1 \leq h \leq \nu\}$ 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.
Remark 2

Let $C$ be an $[n, k, d]_q$ cyclic code with defining set $S_C = \{i_1, \ldots, i_{n-k}\}$. Let $\tau$ be the number of errors, $\nu$ 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.

<table>
<thead>
<tr>
<th>variables</th>
<th>representant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_{n-k}$</td>
<td>correctable syndromes</td>
</tr>
<tr>
<td>$z_1, \ldots, z_\tau$</td>
<td>error locations</td>
</tr>
<tr>
<td>$y_1, \ldots, y_\tau$</td>
<td>error values</td>
</tr>
<tr>
<td>$w_1, \ldots, w_\nu$</td>
<td>erasure locations</td>
</tr>
<tr>
<td>$u_1, \ldots, u_\nu$</td>
<td>erasure values</td>
</tr>
</tbody>
</table>
Remark 2

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

<table>
<thead>
<tr>
<th>variables</th>
<th>representant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_{n-k}$</td>
<td>correctable syndromes</td>
</tr>
<tr>
<td>$z_1, \ldots, z_\tau$</td>
<td>error locations</td>
</tr>
<tr>
<td>$y_1, \ldots, y_\tau$</td>
<td>error values</td>
</tr>
<tr>
<td>$w_1, \ldots, w_\nu$</td>
<td>erasure locations</td>
</tr>
<tr>
<td>$u_1, \ldots, u_\nu$</td>
<td>erasure values</td>
</tr>
</tbody>
</table>
Remark 2

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

<table>
<thead>
<tr>
<th>Variables</th>
<th>Representant</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, \ldots, x_{n-k}$</td>
<td>correctable syndromes</td>
</tr>
<tr>
<td>$z_1, \ldots, z_\tau$</td>
<td>error locations</td>
</tr>
<tr>
<td>$y_1, \ldots, y_\tau$</td>
<td>error values</td>
</tr>
<tr>
<td>$w_1, \ldots, w_\nu$</td>
<td>erasure locations</td>
</tr>
<tr>
<td>$u_1, \ldots, u_\nu$</td>
<td>erasure values</td>
</tr>
</tbody>
</table>
We rewrite previous equations in terms of $X$, $Y$, $Z$, $W$ and $U$, as:

$$\mathcal{F}^{\tau,\nu} = \left\{ \left\{ \sum_{l=1}^{\tau} y_l Z_l^i + \sum_{h=1}^{\nu} u_h W_h^i - x_i \right\}_{i \in S_C}, \right.$$

$$\left. \{ z_l^{n+1} - z_l \}_{l=1,...,\tau}, \right.$$  

$$\left. \{ u_h^q - u_h \}_{h=1,...,\nu}, \right.$$  

$$\left. \{ x_i^{q^m} - x_i \}_{i \in S_C}, \right.$$  

$$\left. \{ y_i^{q-1} - 1 \}_{i=1,...,\tau}, \right.$$  

$$\left. \{ w_h^n - 1 \}_{h=1,...,\nu}, \right.$$  

$$\left. \{ p(n, w_h, W_i) \}_{h \neq i, h, i=1,...,\nu}, \right.$$  

$$\left. \{ z_l p(n, z_l, w_h) \}_{l=1,...,\tau, h=1,...,\nu}, \right.$$  

$$\left. \{ z_l z_k p(n, z_l, z_k) \}_{l \neq k, l, k=1,...,\tau} \right\}$$
We rewrite previous equations in terms of $X, Y, Z, W$ and $U$, as:
\[
\mathcal{F}^{\tau,\nu} = \left\{ \left\{ \sum_{l=1}^{\tau} y_l z_l^i + \sum_{h=1}^{\nu} u_h w_h^i - x_i \right\}_{i \in S_C}, \right. \\
\left. \{ z_l^{n+1} - z_l \}_{l=1,\ldots,\tau}, \\
\{ u_h^q - u_h \}_{h=1,\ldots,\nu}, \\
\{ x_i^{q^m} - x_i \}_{i \in S_C}, \\
\left\{ z_l p(n, z_l, w_h) \right\}_{l=1,\ldots,\tau, h=1,\ldots,\nu}, \\
\left\{ z_l z_k p(n, z_l, z_k) \right\}_{l \neq k, l, k=1,\ldots,\tau} \right\} \\
\]
\[
\mathcal{I} = \mathcal{I}(\mathcal{F}^{\tau,\nu}) \text{ depends only on code } C \text{ and on } \nu.
\]

**Lemma**

Ideal $\mathcal{I}$ is stratified.
Remark 2

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

$$
\mathcal{F}^{\tau,\nu} = \left\{ \left\{ \sum_{l=1}^{\tau} y_l z_i^l + \sum_{h=1}^{\nu} u_h w_i^h - x_i \right\}_{i \in S_c}, \right. \\
\left. \left\{ z_i^{n+1} - z_i \right\}_{l=1,\ldots,\tau}, \right. \\
\left. \left\{ u_h^q - u_h \right\}_{h=1,\ldots,\nu}, \right. \\
\left. \left\{ x_i^{q^m} - x_i \right\}_{i \in S_c}, \right. \\
\left. \left\{ z_l p(n, z_l, w_h) \right\}_{l=1,\ldots,\tau, h=1,\ldots,\nu}, \right. \\
\left. \left\{ z_{l,k} p(n, z_l, z_k) \right\}_{l \neq k, l, k=1,\ldots,\tau}. \right. 
$$

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

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.
Remark 3

It is possible to extend Cooper’s ideas to decode *affine–variety codes*.
Remark 3

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

$$E_q[X] = \{x_1^q - x_1, x_2^q - x_2, \ldots, x_m^q - x_m\} \subseteq I.$$
Remark 3

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

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

Let $P_1, P_2, \ldots, P_n$ be the points of the variety defined by $I$. 
Remark 3

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)

$$\phi : R = \mathbb{F}_q[x_1, \ldots, x_m]/I \longrightarrow (\mathbb{F}_q)^n$$

$$\phi : f \quad \longmapsto \quad (f(P_1), \ldots, f(P_n)).$$
Remark 3

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

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

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)

$$\phi : R = \mathbb{F}_q[x_1, \ldots, x_m]/I \longrightarrow (\mathbb{F}_q)^n$$

$$\phi : f \longrightarrow (f(P_1), \ldots, f(P_n)).$$

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.
Remark 3

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 $C^\perp(I, L)$. 
Remark 3

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

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

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 $C^\perp(I', L')$).
Remark 3

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

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

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 \( H \).
Let $c = (c_0, \ldots, c_{n-1})$, $v = (v_0, \ldots, v_{n-1})$ and $e = (e_0, \ldots, e_{n-1})$. 
Remark 3

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.
Let \( c = (c_0, \ldots, c_{n-1}) \), \( \nu = (\nu_0, \ldots, \nu_{n-1}) \) and \( e = (e_0, \ldots, e_{n-1}) \).

From \( Hv^T = He^T = s \), we get

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

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.
Remark 3

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.
Remark 3

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.
Remark 3

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.

\[
I_{*}^{C, t} = \left\langle \left\{ \sum_{j=1}^{t} e_j b_i(x_{j1}, \ldots, x_{jm}) - s_i \right\}_{1 \leq i \leq r}, \left\{ e_j^{q-1} - 1 \right\}_{1 \leq j \leq t}, \left\{ g_h(x_{j1}, \ldots, x_{jm}) \right\}_{1 \leq h \leq l}, \left\{ x_{jl}^q - x_{jl} \right\}_{1 \leq j \leq t, 1 \leq l \leq m}, \left\{ x_{jl} x_{j\tilde{l}} \prod_{1 \leq l \leq m} ((x_{jl} - x_{j\tilde{l}})^{q-1} - 1) \right\}_{1 \leq j < j' \leq t} \right\rangle
\]
Remark 3

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

\[
l_{*}^{C, t} = \left\langle \left\{ \sum_{j=1}^{t} e_j b_i(x_{j1}, \ldots, x_{jm}) - s_i \right\}_{1 \leq i \leq r}, \left\{ e_j^{q-1} - 1 \right\}_{1 \leq j \leq t}, \left\{ g_h(x_{j1}, \ldots, x_{jm}) \right\}_{1 \leq h \leq l}, \left\{ x_{jl}^q - x_{jl} \right\}_{1 \leq j \leq t, 1 \leq l \leq m}, \left\{ x_{jl} x_{\tilde{j}l} \prod_{1 \leq l \leq m} ((x_{jl} - x_{\tilde{j}l})^{q-1} - 1) \right\}_{1 \leq j < \tilde{j} \leq t} \right\rangle
\]

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 \( l_{*}^{C, t} \) (they are the polynomials with leading terms of type \( x_{i}^{t_i} \)).
Remark 3

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.

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

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 $x_{i}^{t_i}$).
Remark 4

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.
Remark 4

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.
Remark 4

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 some special classes of cyclic 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.
Remark 4

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 some special classes of cyclic 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.
Remark 4

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

$$y^2 + y = x^3$$

over $\mathbb{F}_4$

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

$$L_{C,1} = x_1^2 + x_1(s_4^2 s_1 + s_4 s_2 s_3^2 + s_4 s_2 + s_2 s_1^2) +$$

$$s_5^2 s_3 + s_5 s_3 s_2 + s_4^2 s_3 s_2 + s_4^2 s_3 s_2 s_1 + s_4^2 s_3 s_2 s_1 + s_4^2 s_3 s_2 s_1 + s_4 s_3 s_1 + s_4 s_3 s_1 +$$

$$s_4 s_3 s_1 + s_4 s_3 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1 +$$

$$s_4 s_3 s_1 + s_4 s_3 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1 +$$

$$s_4 s_3 s_1 + s_4 s_3 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1$$

$$L_{C,2} = y_1^2 + y_1 +$$

$$x_1 s_4^2 s_2 s_1^3 + x_1 s_4 s_3^2 s_1^3 + x_1 s_4 s_3^2 s_1^3 + x_1 s_3 s_2^2 s_1^3 + x_1 s_3 s_2^2 s_1^3 +$$

$$s_5^2 s_4 s_2 + s_5 s_4 s_3 + s_5 s_4 s_3 + s_5 s_3 s_2^2 s_1 + s_4^2 s_3 s_2^2 s_1 + s_4^2 s_3 s_2^2 s_1 + s_4^2 s_3 s_2^2 s_1 +$$

$$s_4 s_3 s_2 s_1 + s_4 s_3 s_2 s_1 + s_4 s_3 s_2 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1 +$$

$$s_4 s_3 s_2 s_1 + s_4 s_3 s_2 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1 + s_3^2 s_2 s_1.
Example

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

\[ "L_{C,1}" = x_1^2 + x_1(s_4 s_1 + s_4 s_2 s_1^3 + s_4 s_2^2 + s_2 s_1^2 + s_2^2 s_3) + s_4^2 s_2 + s_2^2 s_1 + s_4 / s_1 \]

\[ "L_{C,2}" = y_1^2 + y_1 + x_1^3 \]
References

M. Caboara, T. Mora “The Chen-Reed-Helleseth-Truong Decoding Algorithm and the Gianni-Kalkbrenner Shape Theorem” , J AAECC, 13 (2002).


T. Mora, E. Orsini “Decoding cyclic codes. The Cooper philosophy”, accepted.