An Attempt to Reduce the Computational Complexity in the Ikegami-Kaji Algorithm for Maximum Likelihood Decoding

Yuta Tsuji* , Toshirou Yoshimizu , Sennosuke Watanabe** and Yoshihide Watanabe***

(Received 19 April, 2013)

Maximum likelihood decoding (MLD) is performed by choosing the error vector with the minimal Hamming weight among the set of vectors having the same syndromes as the received word. Thus, it is formulated as an integer programming (IP) problem on finite fields. There is a famous algorithm by Conti and Traverso for solving IP problems using the Gröbner basis in the polynomial ideal theory. Ikegami and Kaji have presented the algorithm for solving MLD using Gröbner basis; which is an analogue of the Conti and Traverso's algorithm. Once we have the Gröbner basis with respect to the adapted ordering of the ideal associated with the binary code, we efficiently perform MLD by computing the normal form of the monomials corresponding to the received word. However the algorithm has a serious problem on the computational complexity of the Gröbner basis. The present paper is an attempt to reduce the computational complexity of the Gröbner basis of the ideals associated with binary codes. We take two approaches: One is to apply the Gröbner basis conversion algorithm by Faugère et al. and the other is to determine the subset of binomials in the Gröbner basis that are indispensable for MLD. It is known that the Gröbner basis of the ideals associated with binary code with respect to the lexicographic ordering can be obtained easily. We convert this Gröbner basis to the Gröbner basis with respect to the adapted ordering by using FGLM algorithm. We have shown that this approach is not effective for our purpose. As for the second approach we have found the specified subset of the Gröbner basis that seems to be effective for the decoding.

Key words : Gröbner basis, maximum likelihood decoding, Ikegami-Kaji algorithm, FGLM algorithm

キーワード : グレブナー基底，最尤復号，池上・楫アルゴリズム，FGLM アルゴリズム

最尤復号に対する池上・楫アルゴリズムの計算量を減らすための試み

辻 雄 太・吉 水 敏 郎・渡 辺 扇 之 介・渡 邊 芳 英

1. はじめに

本論文の主題は，Gröbner 基底を用いた線形符号，特に BCH 符号の最尤復号である．最尤復号とは，受信語が与えられたときに，ハミング距離の意味で受信語に最も近い重みをもつ符号語を正しい送信語として復号する方法である．このような復号を行うためには，受信語と同じシンドロームをもつベクトルのなかでハミング重み（零でないビットの数）が最小となるベクトルを見出し，それを誤りベクトルとして，受信語ベ

* Graduate School of Science and Engineering, Doshisha University
Telephone:0774-65-6302, E-mail:dum0924@mail4.doshisha.ac.jp
** Graduate School of Science and Engineering, Doshisha University
*** Department of Mathematical Science, Doshisha University

クトルから引けばよい．このような誤りベクトルを見出す問題は，与えられたシンドロームをもつベクトルの中で，すなわち，シンドローム方程式を満たすベクトルの中で，重みが最小のベクトルを求めることとして定式化される．シンドローム方程式は，有限体上の線形方程式であるから，最尤復号の問題は有限体上の線形計画問題または整数計画問題として定式化できる．整数計画問題を Gröbner 基底を用いて解くアルゴリズムとしてよく知られているのは，Conti-Traverso のアルゴリズム [3] であるが，池上・楫は Conti-Traverso のアルゴリズムの類似を用いて，最尤復号問題を解くアルゴリズムを提唱した [1]．しかし，池上・楫アルゴリズムには大きな問題点がある．それは，最尤復号を行う符号の符号長が大きくなるにつれて，復号に必要な Gröbner 基底が飛躍的に大きくなることにある．現実には，現在開発されている計算機代数システムを利用した場合，符号長が $n = 31$ を超えるような符号に対して，復号に必要な Gröbner 基底は計算不可能であることが報告されている [4, 5]．

本論文の目的は，大きな符号長をもつ BCH 符号について，復号に必要な Gröbner 基底を計算するためにどのような工夫が可能かを探ることである．まず，池上・楫アルゴリズムにおいて，符号に付随して定義されるイデアルの辞書式順序に関する Gröbner 基底が符号の生成行列から簡単に計算できること [5]，およびそのイデアルが零次元であることに注目して，Gröbner 基底変換のアルゴリズムを用いる [2]．本論文では辞書式順序の Gröbner 基底を復号に使うための適合項順序（次数順序）の Gröbner 基底に変換する．実際の計算には，計算機代数システム Maple や Singular に実装されている Gröbner 基底変換アルゴリズムを用いた．しかし，BCH 符号に対する計算機実験の結果によれば，このような Gröbner 基底変換の利用は我々の目的にはほとんど役に立たないことが実証された．そこで，我々は別の方法を探ることとし，Gröbner 基底に含まれる 2 項式全体のなかで，どの 2 項式が実際の復号に使われるかを実例を用いて調べ，一つの知見を得た．この知見についても報告する．

2. 最尤復号法

2.1 シンドロームと最尤復号法

C を $\mathbb{F}_2$ 上の (n,k) 2 元線型符号とする．$\mathbb{F}_2$ 上の (n,k) 線型符号とは，有限体 $\mathbb{F}_2$ 上の n 次元数ベクトル空間 $\mathbb{F}_2^n$ の k 次元部分空間のことをいう．(n,k) 2 元線形符号 C の k 個の基底ベクトル $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k$ を行ベクトルと考えて，これらの行ベクトルを並べてできる $k \times n$ 行列 G を C の生成行列という．2 元線形符号 C の直交補空間は $n-k$ 次元の部分空間であり，これも線形符号となるが，これを双対符号といい，$C^\perp$ で表す．C の双対符号 $C^\perp$ の生成行列 H を C のパリティ検査行列と呼ぶ．パリティ検査行列 H は $(n-k) \times n$ 行列である．パリティ検査行列は符号語に誤りがあるかを検査する役割があり，

$$H\boldsymbol{c} = \boldsymbol{0}$$

と $\boldsymbol{c} \in C$ であることは同値である．H が (n,k) 符号 C のパリティ検査行列であるとき，ベクトル $\boldsymbol{r} \in \mathbb{F}_2^n$ のシンドローム $\boldsymbol{s}$ を次式で定義する．

$$\boldsymbol{s} = H\boldsymbol{r}$$

受信語が，符号語 $\boldsymbol{c}$ と誤りベクトル $\boldsymbol{e}$ の和 $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ であるならば，

$$\boldsymbol{s} = H(\boldsymbol{c} + \boldsymbol{e}) = H\boldsymbol{e}$$

最尤復号とは，受信語 $\boldsymbol{r}$ に最も近い符号語を見つけることで復号を行う方法である．そのためには受信語 $\boldsymbol{r}$ と同じシンドロームをもち，(Hamming) 重みが最小となるベクトルを誤りベクトル $\boldsymbol{e}$ とすればよい．ここで，ベクトル $\boldsymbol{e} \in \mathbb{F}_2^n$ の Hamming 重み $w(\boldsymbol{e})$ とは、$\boldsymbol{e}$ の成分のなかで，1(一般には非零元) の数のことである．

3. Gröbner 基底

3.1 単項式順序と Gröbner 基底

体 k の元を係数にもつ n 変数の多項式環を $k[x_1,\ldots,x_n]$ とする．$k[x_1,\ldots,x_n]$ の単項式を多重指数 $\alpha=(\alpha_1,\ldots,\alpha_n)$ を用いて次のように略記する．

$$x^\alpha = x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n} \qquad (\alpha_i\text{は非負の整数})$$

単項式 x^α, x^β の間の順序 $>$ が単項式順序であるとは，次の (1)〜(3) が満たされているときをいう．

(1) 全順序である．すなわち $\alpha\neq\beta \Rightarrow x^\alpha > x^\beta, x^\alpha < x^\beta$ のどちらかが成り立つ．

(2) 積と両立する．すなわち，$x^\alpha > x^\beta \Rightarrow x^\alpha x^\gamma > x^\beta x^\gamma$

(3) 整列順序である．すなわち，単項式順序から成る任意の集合に最小元がある．

変数順序を $x_1 > x_2 > \cdots > x_n$ と固定する．本論文は次のような単項式順序が必要となる．

1. 辞書式順序：$i = 1,\ldots,n$ のいずれかについて，$\alpha_1=\beta_1,\ldots,\alpha_{i-1}=\beta_{i-1}$ かつ $\alpha_i>\beta_i$ が成り立つとき $x^\alpha >_{lex} x^\beta$ と定義すると，$>_{lex}$ は単項式順序となる．これを辞書式順序という．

2. 次数順序: 多重指数 α について $|\alpha|=\alpha_1+\cdots+\alpha_n$ とおく．$|\alpha|$ を単項式 x^α の総次数という．適当な単項式順序 $>$ を一つ定める．そのとき新たな順序 $>_{deg}$ を次のように定める．$x^\alpha >_{deg} x^\beta$ であるのは，(i)$|\alpha|>|\beta|$ であるときまたは (ii)$|\alpha|=|\beta|$ かつ単項式順序 $>$ に関して $x^\alpha > x^\beta$ が成り立つときと定義する．このような順序は単項式順序となり，次数順序と呼ばれる．

3. ブロック単項式順序：2 種類の変数の組 $\boldsymbol{x} = (x_1,\ldots,x_n)$ および $\boldsymbol{y}=(y_1,\ldots,y_m)$ を導入し，$m+n$ 変数の多項式環 $R=k[x_1,\ldots,x_n,y_1,\ldots,y_m]$ を考える．多項式環 $k[x_1,\ldots,x_n]$ には単項式順序 $>_1$ が定義されているとし，多項式環 $k[y_1,\ldots,y_m]$ には単項式順序 $>_2$ が定義されているとする．そのとき二つの単項式順序を組み合わせて R における単項式順序 $>$ を次のように定める．$x^\alpha y^\mu > x^\beta y^\nu$ であるのは (i) $x^\alpha >_1 x^\beta$ であるか，または (ii) $x^\alpha = x^\beta$ かつ $y^\mu >_2 y^\nu$ であるときと定義する．そのとき $>$ は $m+n$ 変数多項式環 R における単項式順序となる．このような順序 $>$ を $\boldsymbol{x}\succ\boldsymbol{y}$ とするブロック順序とよび $(>_1,>_2)$ で表す．

多項式環 $k[x_1,\ldots,x_n]$ において，ある単項式順序 $>$ を固定する．そのとき $f\in k[x_1,\ldots,x_n]$ に対して，f は項 $a_\alpha x^\alpha$ $(a_\alpha\in k)$ の有限和であるから，f に含まれる項のうちで x^α が単項式順序 $>$ の意味で最大のものがある．そのような項を $\mathrm{LT}_>(f)=a_\alpha x^\alpha$ と表し，f の主項と呼ぶ．$I\subset k[x_1,\ldots,x_n]$ をイデアルとする．有限部分集合 $G=\{g_1,\cdots,g_t\}\subset I$ が単項式順序 $>$ に関するイデアル I の Gröbner 基底であるとは，任意の $f\in I$ に対して $\mathrm{LT}_>(g_i)$ が $\mathrm{LT}_>(f)$ を割り切るような $g_i\in G$ が存在するときをいう．任意の f を $g_1,\cdots,g_t$ で割った余りは一意的で，その余りを $\overline{f}^G$ と表す．任意の多項式 f について，$f\in I$ であるための必要十分条件は $\overline{f}^G=0$ となることである．イデアル I の有限個の生成元が与えられたとき，Buchberger のアルゴリズムを用いることにより，指定された単項式順序に関するイデアル I の Gröbner 基底を計算することができるが，そのアルゴリズムは Maple を始めとする多くの計算機代数システムにおいて実装されている．

3.2 零次元イデアルと Gröbner 基底変換

本論文で扱う 2 元符号の最尤復号に現れるイデアルは特別な性質をもつイデアルであり，それは零次元イデアルと呼ばれるものの例になっている．零次元イデアルの一つの定義を述べよう．k を体として，k 係数多項式環のイデアル $I\subset k[x_1,\ldots,x_n]$ が，零次元であるとは，イデアルによる商環 $V=k[x_1,\ldots,x_n]/I$ を k 上のベクトル空間とみたとき，それが有限次元であるときをいう．I が零次元イデアルなら $A=k[x_1,\ldots,x_n]/I$ のベクトル空間としての次元が有限であるので，線形代数的な考察を用いて，零次元イデアル I のある単項式順序に関する Gröbner 基底 G を別の単項式順序に関する Gröbner 基底 G' に変換することができる．こ

の手続きを Gröbner 基底変換と呼ぶ [2]．Gröbner 基底変換のアルゴリズムも Maple など多くの計算機代数システムで利用可能である．本論文では，まず少し特別な2元線形符号（組織符号）については，最尤復号を行う際に定義される零次元イデアル I の辞書式順序に関する Gröbner 基底 G が容易に求められることに注目する [5]．さらに，そのような辞書式順序の Gröbner 基底 G を，復号で必要となる適合順序に関する Gröbner 基底に変換する際に Gröbner 基底変換のアルゴリズムを用いる．

4. Gröbner 基底と最尤復号

4.1 池上・楫アゴリズム

C を 2 元線形符号とし，そのパリティ検査行列 H を列ベクトル $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n \in \mathbb{F}_2^m$ を用いて

$$H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n)$$

と表す．簡単のため $m = n - k$ とおく．すでに述べたように，最尤復号とは，受信語 $\boldsymbol{r}$ と同じシンドローム $\boldsymbol{s}$ を持ち，重みが最も小さい誤りベクトル $\boldsymbol{e}$ を見つけて復号することである．最尤復号法は，次のような $(0,1)$ 整数計画問題の類似物として定式化できる．

$$\begin{cases} \text{Minimize} & w(\boldsymbol{e}) = e_1 + \cdots + e_n \\ \text{Subject to} & H\boldsymbol{e} = \boldsymbol{s} \pmod 2 \end{cases} \tag{1}$$

この最尤復号問題を Gröber 基底を使って解く方法が次の池上・楫アルゴリズムである．

アルゴリズム (池上・楫アルゴリズム)．
2 元線形符号のパリティー検査行列を $H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n)$ とおく．受信語 $\boldsymbol{r}$ のシンドロームを $\boldsymbol{s} = H\boldsymbol{r}$ で定義する．受信語 $\boldsymbol{r}$ を既知としてその最尤復号 $\boldsymbol{e}$ を次のようなアルゴリズムで求める．

1. 二組の変数 $\boldsymbol{t} = (t_1, \ldots, t_m)$ と変数 $\boldsymbol{x} = (x_1, \ldots, x_n)$ を導入してイデアル $I \subset k[\boldsymbol{t}, \boldsymbol{x}]$ を次のように定義する．

$$I = \langle x_1 - \boldsymbol{t}^{\boldsymbol{h}_1}, \ldots, x_n - \boldsymbol{t}^{\boldsymbol{h}_n}, t_1^2 - 1, \ldots, t_m^2 - 1 \rangle$$

ただし，$\boldsymbol{t}^{\boldsymbol{h}} = t_1^{h_1} \cdots t_m^{h_m}$ とする．このようなイデアル I を2元線形符号 C に付随するイデアルという．

2. イデアル I から適合項順序に関する Gröbner 基底 G を求める．適合項順序とは，$\boldsymbol{t} \succ \boldsymbol{x}$ とするブロック順序であって，変数 $\boldsymbol{x}$ については次数順序を採用したものをいう．

3. $\boldsymbol{t}^{\boldsymbol{s}}$ を G で割った余り $\overline{\boldsymbol{t}^{\boldsymbol{s}}}^G$ を求める．$\overline{\boldsymbol{t}^{\boldsymbol{s}}}^G = \boldsymbol{x}^{\boldsymbol{e}}$ となったなら，$\boldsymbol{e}$ が重み最少の誤りベクトルとなる．よって，$\boldsymbol{r} - \boldsymbol{e}$ が最尤復号結果を与える．

本研究では．2元線形符号の例として，実際の問題に非常によく使われている BCH 符号を用いるが，紙面の都合で BCH 符号についての説明は省略する．

池上・楫のアルゴリズムについては，次のような利点と問題点がある．

利点 一つの2元線形符号に対してそれに付随するイデアル I の適合項順序に関する Gröbner 基底を，前もって計算しておけば，復号計算ごとの再計算は不要であり，その Gröbner 基底を用いることにより，得られた受信語のシンドロームに対して，最尤復号に必要な余りの計算を効率的に実行することができる．

問題点 符号長 n が大きくなるにつれて復号に必要な適合順序に関する Gröbner 基底の個数が飛躍的に増大し，それに伴って計算量も大きくなり，計算時間の面からもメモリーの制約からも計算が難しくなる．本研究で使用した計算機代数システム Maple では，$(n,k) = (31,26)$ の BCH 符号の Gröbner 基底は計算可能であったが，$(n,k) = (31,21)$ の BCH 符号においては，メモリー不足のため Gröbner 基底計算は途中で中断された．

4.2 池上・楫アルゴリズムと FGLM アルゴリズム

取り扱う符号が2元組織符号であるときは，池上・楫アルゴリズムにおいて復号において定義されるイデアル I の辞書式順序の Gröbner 基底 $\mathcal{G}$ を容易に求められることが知られている [5]．

定理 1. (n,k)2 元線形符号 C は，組織符号であるとする．すなわちその生成行列が $G=(E_k\ A)(E_k$ は k 次の単位行列) の形であると仮定する．そのとき生成行列 G の行ベクトルを $\boldsymbol{g}_i\ (i=1,\ldots,k)$ として，$G=(\boldsymbol{g}_1,\ldots,\boldsymbol{g}_k)^T$ とおく．そのときイデアル I の，変数順序 $t_1>\cdots>t_m>x_1>\cdots>x_n$ の辞書式順序に関する Gröbner 基底 $\mathcal{G}$ は，

$$\mathcal{G}=\{t_1^2-1,\ldots,t_m^2-1,x_1-f_1,\ldots,x_k-f_k,\\ x_{k+1}^2-1,\ldots,x_n^2-1\}$$

である．ただし，$f_i=\boldsymbol{x}^{\boldsymbol{g}_i-\boldsymbol{e}_i}(i=1,\ldots,k)$ である．$\boldsymbol{g}_i$ の 1 列目から k 列目までの成分のなかで 1 となるのは i 番目だけで他はすべて 0 であることに注意する．

以下に示す表は，与えられた符号長と情報長をもつ BCH 符号について符号に付随するイデアルの Gröbner 基底の数を辞書式順序と適合項順序について示したものである．定理 1 により得られる，辞書式順序に関する Gröbner 基底の数は符号のサイズが大きくなっても，それほど大きくはならない．一方，復号に必要となる適合順序に関する Gröbner 基底の数は，符号のサイズが大きくなると飛躍的に大きくなることが見て取れる．

Table 1. The number of binomials in the Gröbner basis.

(n,k)	adapted ordering	lex. ordering
(7,4)	31	10
(15,11)	124	19
(15,7)	364	23
(15,5)	970	25
(31,16)	126	21
(31,26)	383	36
(31,21)	4034	41

まず，I が零次元イデアルであることに注意して，定理 1 により得られるイデアル I の辞書式順序に関する Gröbner 基底を Gröbner 基底変換のアルゴリズム (FGLM アルゴリズム) を用いて，適合順序に関する Gröbner 基底に変換することにより，適合項順序に関する Gröbner 基底計算が困難になるという問題点の解決を試みた．計算処理ソフト Maple 15 によって得られた結果を次に示す．

Table 2. CPU time.

(n,k)	adapted ordering [s]	FGLM [s]
(7,4)	0.109	0.094
(15,11)	0.608	0.546
(15,7)	65.848	65.661
(15,5)	5485.105	5650.965
(31,16)	2.434	2.309
(31,26)	30.529	30.545

計算機実験の結果を述べると，簡単に計算できる辞書式順序の Gröbner 基底から出発して，FGLM アルゴリズムを適用して，適合項順序による Gröbner 基底を計算した所，その計算時間は，はじめから適合項順序によって Gröbner 基底を計算することと比べ，計算にかかる時間はほとんど変わらず，残念ながら，問題点が改善されたとは言えなかった．

4.3 池上・楫アルゴリズムによる近似最尤復号

これまでの計算機実験により，符号に付随するイデアル I の適合順序に関する Gröbner 基底の計算の効率よく計算することは極めて困難であるということが分かった．そこで，必要な計算を少しでも減らすためのだ 1 歩として，適合項順序に関する Gröbner 基底のなかで実際に復号に使われる基底がどのようなものであるかを簡単な例で具体的に調べてみることにした．まず，そのような具体的な例として，Hamming 符号を取り上げた．Hamming 符号は 1 個の誤り訂正能力しかない符号であるが，BCH 符号のなかで特別なクラスを作っており，表では符号長と情報長が $(7,4),(15,11),(31,26)$ で与えられるものがそれにあたる．Hamming 符号に対する最尤復号においては，池上・楫アルゴリズムに

おける Gröbner 基底計算は必要なく，単純にステップ (1) で定義されるイデアル I の生成元だけで復号が可能であることが確かめられた．この事実着目し，池上・楫アルゴリズムにより求められた Gröbner 基底の各２項式に対して，類似の形を持つ２項式をまとめてグループ分けを行い，それらのグループの２項式が，最尤復号にどのように貢献しているかを調べた．それを例について説明する．

例 **2.** $(n,k)=(15,7)$ の BCH 符号について，適合項順序に関する Gröbner 基底の総個数が 364 ある．それらを次に示すようなグループに類別する．

1. 変数 x_i について x_i^2-1 型．例えば

$$-1+x_{15}^2,-1+x_{14}^2,-1+x_{13}^2$$

2. 変数 x_i についての 3 次式と 2 次式の差．例えば

$$-x_7x_{11}+x_{15}x_{14}x_{13},\quad -x_8x_1+x_{15}x_{14}x_{12},$$
$$-x_{13}x_7+x_{15}x_{14}x_{11}$$

3. 変数 x_i について 3 次式の差．例えば

$$-x_{10}x_{15}x_3+x_{14}x_{13}x_9,\quad -x_9x_{10}x_{15}+x_{14}x_{13}x_3,$$
$$-x_{15}x_7x_5+x_{14}x_{12}x_9$$

4. 変数 x_i について 4 次式と 3 次式の差．例えば

$$-x_{12}x_{11}x_3+x_{15}x_{14}x_{10}x_5,$$
$$-x_7x_3x_1+x_{15}x_{14}x_9x_4,$$
$$-x_8x_{13}x_3+x_{15}x_{14}x_6x_2$$

5. t 変数を含む２項式．

$$-x_1+t_8,-x_2x_1+t_7,-x_{10}x_{14}+t_6$$

このなかで，1 のタイプの２項式は書き換え $x_1^2\mapsto 1(i=1,\ldots,n)$ を表し，5 のタイプの２項式は $\boldsymbol{t}$ 変数を $\boldsymbol{x}$ 変数に書き換える操作に必要である．残る２項式は，2, 4 のように，異なる次数の単項式の差で表されるグループ (I) と 3 のように同じ次数の単項式の差になっているグループ (II) に分けられる．これらが $\boldsymbol{t}^s$ の Gröbner 基底による余りを計算する過程において，それぞれ次の役割をはたすことがわかった．

(1) (I) のタイプの２項式 $\boldsymbol{x}^{\boldsymbol{\alpha}}-\boldsymbol{x}^{\boldsymbol{\beta}}$ $(|\boldsymbol{\alpha}|>|\boldsymbol{\beta}|)$ により，ある重みをもち，与えられたシンドローム $\boldsymbol{s}$ をもつベクトルは，より小さい重みをもち，同じシンドローム $\boldsymbol{s}$ をもつベクトルに書き換えられる．

(2) (II) のタイプの２項式 $\boldsymbol{x}^{\boldsymbol{\alpha}}-\boldsymbol{x}^{\boldsymbol{\beta}}$ $(|\boldsymbol{\alpha}|=|\boldsymbol{\beta}|)$ により，重みが同じであるが，単項式順序の意味で順序が低く，同じシンドロームをもつベクトルへの書き換えが行われる．

この結果を受けて，いくつかの例に対して，I の Gröbner 基底 G からタイプ (II) を除いたものを G' として，G' によって簡約化を行ったところ，重みが最小となるベクトルを誤りベクトルとするという点においては，最尤復号は正しく実行された．このような実験的検証を受けて，このような復号方法を，我々は，池上・楫アルゴリズムによる近似最尤復号と呼ぶ．

5. まとめ

本研究では，池上・楫アルゴリズムによる最尤復号において，最大の問題点である，復号に必要な Gröbner 基底のサイズが符号長の増加に伴って飛躍的に増大し，基底を求める計算ができなくなるという問題点の克服を目指した．まず，復号に際して定義されるイデアルの辞書式順序の Gröbner 基底が容易に求められることに注目し，その Gröbner 基底を復号に必要な適合項順序（次数順序）に関する Gröner 基底に，FGLM アルゴリズムを使って変換することを試みた．実際に計算機代数システム Maple に実装されている FGLM アルゴリズムを用いて，いくつかの符号長と情報長をもつ BCH 符号について，計算可能なものに限り，復号に必要な Gröbner 基底の計算時間を測ってみたが，直接適合項順序の Gröbner 基底を計算することに比べて，改善は見られず，満足な結果は得られなかった．

次に，復号に実際必要なGröbner基底の元はどのようなものであるかを，(15, 7) のBCH符号を例にとって詳しく調べてみた．その結果，Gröbner基底の全ての２項式が最尤復号に必要ではないことが分かった．実際復号に使われるGröbner基底 G の元は大きく分けて５つのグループに分別することができるが，そのうちGröbner基底 G を用いた最尤復号に関して不要となるものが明らかとなり（グループ (II)），それらを除いた２項式だけで最尤復号ができることが分かった．この事実は，一般に示されたわけではないが，もし，一般的に示されれば，復号に必要な基底の数が大幅に削減され，アルゴリズムの効率化に有効であると思われる．さらに，このようにGröbner基底の一部であって，復号に必要なものだけを効率よく計算するアルゴリズムの開発も重要な課題となる．

参 考 文 献

1) D. Ikegami and Y. Kaji, “*Maximum Likelihood Decording for Linear Block Dodes Using Gröebner Base*”, IEICE Trans. Fundamentales, **86-A-3**, 643-651 (2003).

2) D. Lazard, J. Faugére, P. Gianni and T. Mora, “*Efficient computation of zero-dimensional Gröbner base by change of ordering*”, J. Symbolic Comput., **16**, 329-344 (1993).

3) P. Conti and C. Traverso, “*Buchberger algorithm and integer programming*”, proceeding AAECC, **9**, 130-139 (1991).

4) 宇井友寿, “Gröebner基底を用いた最尤復号法の検討”, 同志社大学 工学研究科電気工学専攻修士論文, (2005).

5) 吉武純子, “Gröebner基底と最尤復号”, 同志社大学 工学研究科数理環境科学専攻修士論文, (2006).