Maximaを用いたBCH符号の実装

2008MI037 日比野　晃洋
指導教員：小籐　俊幸

1　はじめに

現代の情報社会において，携帯電話や無線 LAN などでデジタル・データを正確に送受信することは，言うまでもなく，我々の生活に必要不可欠である．伝送路には歪みや雑音が必ず存在してしまうため，それによる誤りの検出・訂正をしなくてはならない．これは，誤り訂正技術を用いることにより，デジタル信号の高品質な伝送や記録・再生を実現し，さらにある一定の送信電力でより優れた品質を実現している．あるいは所要送信電力を低減するためにも，誤り制御技術は必要である．誤り制御技術は携帯電話や衛星通信などのデジタル通信システムや，CD や DVD などの AV 機器（記録装置）に広く用いられている．通信システムにおける誤り訂正技術は，FEC 方式と ARQ 方式大別される．また，ARQ 方式に FEC を用いた Hybrid ARQ(HARQ) 方式もある．[1]

今回は FEC 方式として代表的である誤り訂正符号の中から，特に BCH 符号について注目し，説明していき，数式処理システムである「Maxima」を用い，実際に BCH 符号により情報の誤りを訂正する，その実装を確認したい．

2　誤り訂正符号

図 1　誤り訂正符号の種類

2.1　線形ブロック符号

ブロック符号とは，符号語と呼ばれる固定長ベクトルからなる場合である．ベクトルの要素数を符号長と呼び n で表す．符号語の要素は l 個の要素からなる符号アルファベットから選ばれる．符号アルファベットが 0,1 の 2 つの元からなるとき，その符号は 2 元符号と呼ばれ，符号語の各要素はビットと呼ばれる．符号長 n の 2 元符号の符号語として取り得るベクトルの個数は 2^n である．この 2^n 個のベクトルから $M = 2^k (k < n)$ 個のベクトルを選択して符号を構成する．このようにして得られるブロック符号を (n,k) ブロック符号という．[1]

2.2　畳み込み符号

ブロック符号が一定の情報長 (ブロック) 単位で符号化する符号であるのに対し，畳み込み符号は過去の数ビットを用いて現時点での符号化ビットを得る符号であり，情報系列と符号系列の対応が逐次的である．符号化率 $|R| = |k/n|$ の畳み込み符号では，情報ビット k ビットに対し，符号化ビット n ビットを出力する．各 n ビットは，現在及び過去の情報ビットへの依存長を拘束長 $|K|$ と一般に表す．拘束長の定義として，過去の情報ビットへの依存長だけを表わす場合もある．[1]

2.3　BCH 符号

BCH 符号とは，他の誤り訂正符号よりも多くの誤りを訂正ができ，冗長度が低く，複合も容易という利点を持っている．

本論は誤り訂正符号の一例として，二重誤り訂正可能な BCH 符号を述べる．

3　Maxima

Maxima は 1960 年代の MIT の Project MAC で開発された MACSYMA(MAC's SYmbolic MAnipulation system) の DOE(エネルギー省) 版を Texas 大学の Schelter 氏が「Common Lisp: The language 第 1 版」(cltl1 と略記) に対応した GCL に移植したものであり，当初は Schelter 氏が Maxima の開発と管理を行っていたが，Schelter 氏の死後はメイリングリストを中心に Maxima の保守・管理と開発が進められている．

このソフトウェアは，数式処理ソフトウェアと呼ばれ，同種の市販のソフトウェアには，Mathematica や Mapple などがある．これらとの大きな違いは，表計算ソフトウェアは，数値計算を主目的としてやっているため，常に数値近似であることに対して，Maxima は数学的にも厳密な計算を主目的としていることが挙げられる．

市販されている他の数式処理ソフトウェアに比べ，ユーザも少なく，アップデートなども自分でチェックしなくてはいけない不便さもあるが，何より高額な代金を支払う必要がなく，無料であり，数式処理ソフトウェアとしては他のものに引けを取らない．[2]

4　ASCII

American Standard Code for Information Interchange(略:ASCII)

1963 年にアメリカ規格協会（ANSI）が定めた，情報交換用の文字コードの体系であり，67 年に国際標準化機構（ISO）で定められた情報交換用符号の国際規格「ISO 646」とほぼ同じものである．7 ビットで表示され，128 種類のローマ字，数字，記号，制御コードで構成されている．実際には 1 文字を 8 ビット（1 バイト）で表現するため，256 種類の文字を扱うことができるが，ASCII が定めていない 128 文字分の拡張領域には，コンピュータメーカーや国によって異なる文字が収録されている．日

本では，拡張領域にカナ文字を収録したコード体系が規格化されている．[3]

5　実装

今回の実装は，本稿 2 章で使用した BCH 符号を，Maxima を用い行った．

今回確認した実装は 2 つあり，2 重誤り訂正符号で，7 ビットのデータを 15 ビットの符号語にした後の，s0,s1 のシンドロームの計算部分から，その誤りを検出，修正し，その複合が正しく行われたことが確認できるものと，

どのような誤りの数にも対応しており，誤りの検出までを行える実装を確認した．

下に示したのは，後者の実装である．つまり，15 ビットの符号語を入力することによって，誤っている箇所のデータが 0 と表示される．これをもとに復号を行うことができる．関数名を BCH() とし，入力する 15 ビットの符号語を b とした．

関数を定めた後に，符号語 b を入力．その後に，関数を実行することにより，その実装が確認できるようにした．

```
[08mi037@08mi037 08mi037]$ maxima

BCH(b):=block(\\
[g,g0,b1,v,s0,s1,x,i,e,c,q,X,u,p,confirming],\\
modulus:2,\\
g:x^8+x^7+x^6+x^4+1,g0:x^4+x+1,b1:b,\\
v:conjugate(b).makelist(x^i,i,0,14),\\
s0:remainder(v,g0),\\
s1:remainder(subst(x^3,x,v),g0),\\
e:{eif(s0):=if s0 = x+1 then 4 else \\
if s0 = x^2+x then 5 else \\
if s0=x^3+x^2 then 6 else \\
if s0=x^3+x+1 then 7 else \\
if s0=x^2+1 then 8 else \\
if s0=x^3+x then 9 else \\
if s0=x^2+x+1 then 10 else \\
if s0=x^3+x^2+x then 11 else \\
if s0=x^3+x^2+x+1 then 12 else \\
if s0=x^3+x^2+1 then 13 else \\
if s0=x^3+1 then 14 else \\
if s0=1 then 1 else 0 },e:eif(s0),\\
c:remainder(x^(15-e)*(s0^3+s1),g0),\\
q:X^2+s0*X+c,\\
makelist\\
(remainder(subst(x^i,X,q),g0),i,0,14),\\
ml:makelist\\
(remainder(subst(x^i,X,q),g0),i,0,14),\\
u:ratsimp(v+x^3+x^11),p:quotient(u,g),\\
remainder(u,g),\\
confirming:remainder(u,g),return([ml]));\\
```

関数をこのように宣言しておく．

```
(%i5) b:[1,0,0,1,1,1,0,0,1,0,0,0,0,0,0];
(%o5)
[1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0]
(%i6) BCH(b);
          3   2          3   3    2       2   3
(%o6) [[x , x  + 1, 1, x , x  + x  + 1, x , x  + 1,
  2       3        3    2       3    2   3    2
 x  + 1, x  + 1,  x  + x  + 1, x  + x , x  + x ,
   2
  x , 1, 0]]
```

ここに出力されているのが検出されたデータであり，この場合は，訂正すべきデータは 15 ビット目である．誤りは 1 つであり，2 重誤り訂正符号の場合は 0 と出力される箇所が 2 つになる．

6　おわりに

本論文の中では，Maxima を用いて，15 ビットの受信データを打ち込むことにより，そのデータを正しく訂正できたかどう　か確認できる実装を行うことができた．

現代社会において不可欠でありながらも，あまり身近に感じることのできない誤り訂正符号について説明してきたが，本論中にも記したように，多くの種類があり，その使用用途もさまざまであった．今回は特に BCH 符号に着目していったが，その他の符号も社会にとって重要であり，追求していくことは多くあると感じた．

具体的にどのように活用されているのかとても興味があったが，衛星や地デジ，携帯電話など，大枠で使われているものはわかったものの，どのようなプログラムで，どのような問題点を乗り越え，使用されているのか，今後は具体例にも触れ，その問題点を探り，考察していきたい．

参考文献

[1] 大槻　知明:http://www.ohtsuki.ics.keio.ac.jp/pdf/class.pdf

[2] http://www.eonet.ne.jp/kyo-ju/maxima.pdf

[3] http://ja.wikipedia.org/wiki/ASCII