誤り訂正符号

研究学生　吉本　覚　　担当教員　笠原　禎也

1.はじめに

近年の IT 革命を支える技術の 1 つとして、「誤り訂正が可能」というデジタル化にともなうメリットがある.これは、デジタル化した情報に何らかの影響で誤りが起きたとしても、元の情報に自動的に復元することができるという技術であり、情報通信の品質向上や記憶メディアの高密度記録化に応用されている.そこで、実際にどのような方法で誤り訂正を行っているのか興味があり、調べることにした.今回は、代表的なハミング符号を用いた方法と畳み込み符号を用いた方法をプログラムで作成し、特性を比較・検討することを目的とする.

2.ハミング符号を用いた方法

ハミング符号は k ビットの情報符号に、ある規則にしたがいmビットの検査符号を付加し、全体が n ビット(n=k+m)で構成されている.

今回は(7,4)ハミング符号を作成した.

シンドローム (s_1, s_2, s_3) を(1),(2),(3)式により計算した.これを 2 進数で読むと、どのビットが誤りなのかを知ることができる.よって、そのビットを 0→1、1→0 に反転することで誤りを元の情報に復元することができる.

$$\begin{cases} s_1 = x_4 \oplus c_1 \oplus c_2 \oplus c_3 & \cdots(1) \\ s_2 = x_2 \oplus x_3 \oplus c_2 \oplus c_3 & \cdots(2) \\ s_3 = x_1 \oplus x_3 \oplus c_1 \oplus c_3 & \cdots(3) \end{cases}$$

3.畳み込み符号による方法

ある時刻 t の情報ビット入力を s_t、その時の符号ビット出力が($\hat{s}_t$　p_t)で表される.ここで、 p_t は過去の情報ビット入力の影響を受ける.今回は 1 つ前の時刻の情報ビットまでの影響を受けるように(4),(5)式で符号化した.

$$\begin{cases} \hat{s}_t = s_t & \cdots(4) \\ p_t = s_t \oplus s_{t-1} & \cdots(5) \end{cases}$$

復号化には、ビタビ復号法が用いられる.

(今回は時間が足りず、復号化を実装できなかった)

4.実験(シミュレーション)

以下のプログラムを作成した.

(1)符号化プログラム

(2)ランダムに誤りを起こすプログラム

(3)復号化プログラム

これらのプログラムを順に動かし、図 2 の send.txt のデータの変化の様子をシミュレーションした.

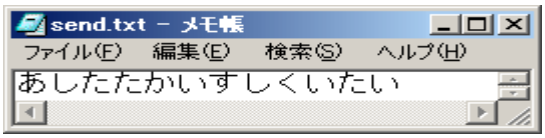

図 2　send.txt

5.実験結果・考察

(3)復号化のプログラムを動かした後の txt ファイルの結果を図 3 に示す.　(1),(2),畳み込み符号の結果は省略.

(誤りを起こすビットの割合を約 0.3 に設定)

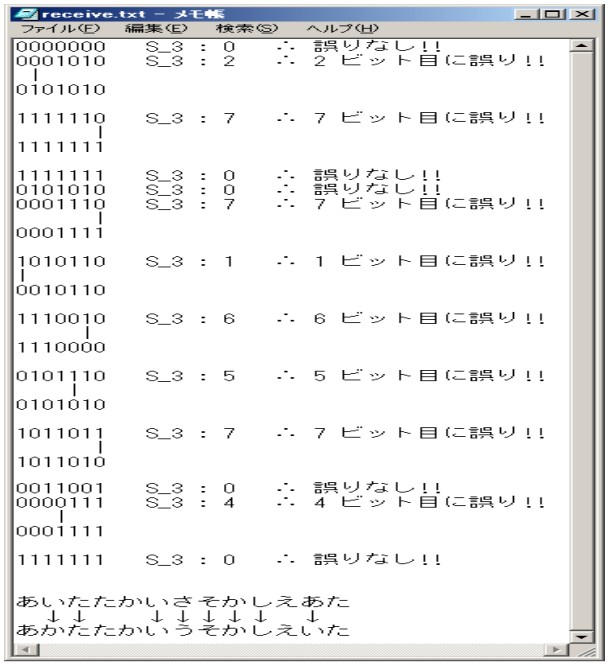

図 7　receive.txt

誤りを起こしたビット数が 2 以上になると、うまく誤り訂正ができないことが分かる.これは、誤りを起こしたビット数が 2 以上になると、シンドロームを計算に間違いが生じ、結果として違うビットを反転させてしまうからだと考えられる.

6.まとめ・反省点・残された課題

自主課題研究はプログラムを作ることだったので、プログラミングのスキルを上達することができた.また、自分のペースで研究を進めることができたので､1 つのことに対して十分に理解することができた.

課題として、作成する予定だった畳み込み符号の復号化プログラムを作成することが残った.

参考文献

TECHI Vol.7 やり直しのための工業数学　三谷政昭著、CQ 出版社