Error Control Coding - Decoding of Convolutional Codes

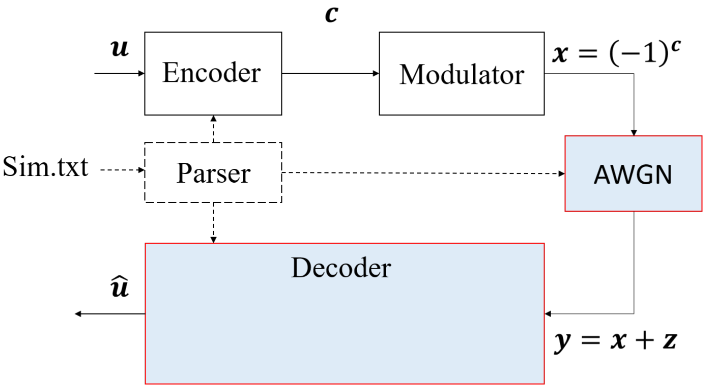
Chao-Yu Chen (陳昭羽)

國立成功大學
電機工程學系/電腦與通信工程研究所

May 3, 2023

Decoding of Convolutional Codes

- Consider the $(2,1,6)$ convolutional code with generator matrix $\boldsymbol{G}(D) = \begin{pmatrix} 1 + D^2 + D^3 + D^5 + D^6 & 1 + D + D^2 + D^3 + D^6 \end{pmatrix}$.
- Block Diagram:



Decoding of Convolutional Codes

$\underline{y} = \underline{x} + \underline{n}$

$\underline{x} \rightarrow E_s \rightarrow 1$, $\underline{n} \rightarrow \frac{N_0}{2} \rightarrow \underbrace{\frac{N_0}{2} \times \frac{1}{E_s}}_{\sigma^2}$

$\underline{u}$ (k) $\rightarrow \boxed{} \rightarrow$ $\underline{x}$ (n)

$k E_b = n E_s$

$\Rightarrow \frac{k}{n} E_b = E_s$, $\frac{k}{n} = R$: $R E_b$

Additive white Gaussian noise (AWGN) channel:

- For a binary code of rate R with BPSK modulation, the noise variance σ^2 is given by

$E_s = R E_b$

$$\sigma^2 = \left(2R\frac{E_b}{N_0}\right)^{-1}$$

$$\sigma^2 = \left(2\frac{E_s}{N_0}\right)^{-1}$$

  where E_b/N_0 is the bit signal-to-noise ratio (SNR).
- E.g., $\sigma^2 = 0.3981$ when $E_b/N_0 = 4$ dB and $R = 1/2$.

$10 \log_{10}\left(\frac{E_b}{N_0}\right) = 4\,\text{dB}$

$\Rightarrow \frac{E_b}{N_0} = 10^{0.4}$

$\frac{E_b}{N_0} \uparrow, \ \sigma^2 \downarrow$

Decoding of Convolutional Codes

- Please use the following pseudo code:

```
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define NTAB 32
#define NDIV (1+(IM-1)/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0-EPS)

main()
{
    ...
    long *idum;
    idum = (long *)malloc(sizeof(long));
    *idum = SEED;  //SEED must be a negative integer.
    ...
    ...
}
```

Decoding of Convolutional Codes

- Please use **normal()** to output two independent normal random variables, n_1 and n_2.

$$
\begin{array}{l}
\texttt{normal}(n_1, n_2, \sigma) \\
\{ \\
\quad \texttt{do}\{ \\
\qquad x_1 = \texttt{ran1(idum)}; \\
\qquad x_2 = \texttt{ran1(idum)}; \\
\qquad x_1 = 2x_1 - 1; \\
\qquad x_2 = 2x_2 - 1; \\
\qquad s = x_1^2 + x_2^2; \\
\quad \}\ \texttt{while}\ (s \geq 1.0) \\
\quad n_1 = \sigma x_1 \sqrt{-2 \ln s / s}; \\
\quad n_2 = \sigma x_2 \sqrt{-2 \ln s / s}; \\
\}
\end{array}
$$

$n_1 \sim N(0, \sigma^2)$

$n_2 \sim N(0, \sigma^2)$

Decoding of Convolutional Codes

- Please use **ran1()** to generate a random variable uniformly distributed in the interval $(0, 1)$.

```
ran1(long *idum)
{
    int j;
    long k;
    static long iy=0;
    static long iv[NTAB];
    double temp;
    if (*idum <= 0 || !iy){
       if (-(*idum) < 1) *idum=1;
       else *idum = -(*idum);
       for (j=NTAB+7;j>=0;j--){
           k=(*idum)/IQ;
           *idum=IA*(*idum-k*IQ)-IR*k;
           if (*idum < 0) *idum += IM;
           if (j < NTAB) iv[j] = *idum;
       }
       iy=iv[0];
    }
    k=(*idum)/IQ;
    *idum=IA*(*idum-k*IQ)-IR*k;
    if (*idum < 0) *idum += IM;
    j=iy/NDIV;
    iy=iv[j];
    iv[j] = *idum;
    if ((temp=AM*iy) > RNMX) return RNMX;
    else return temp;
}
```

Decoding of Convolutional Codes

- Input Sim.txt:

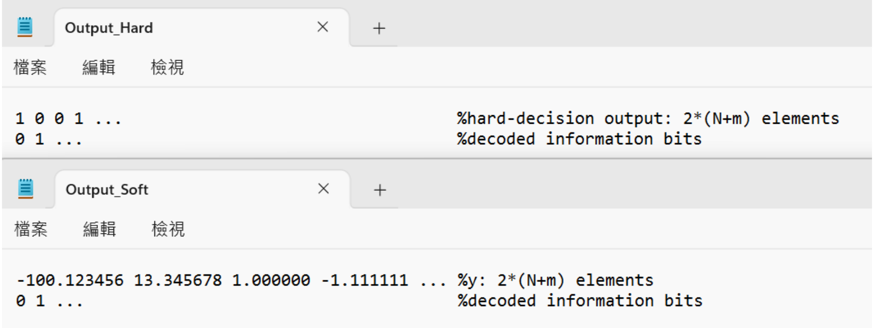
Sim

檔案 編輯 檢視

```
1000  %N: number of decoded bits
32    %L: truncation window length
1.0   %SNR: bit signal-to-noise ratio (dB)
-2023 %:Seed: a negative integer
0     %0=>Hard-decsion; 1=>Soft-decision
```

Decoding of Convolutional Codes

- Output “Output_Hard.txt” or “Output_Soft.txt”:



Output_Hard

檔案 編輯 檢視

```
1 0 0 1 ...                                  %hard-decision output: 2*(N+m) elements
0 1 ...                                      %decoded information bits
```

Output_Soft

檔案 編輯 檢視

```
-100.123456 13.345678 1.000000 -1.111111 ... %y: 2*(N+m) elements
0 1 ...                                      %decoded information bits
```

Encoding of Convolutional Codes

- Use the recursion

$$u_{l+6} = u_{l+1} \oplus u_l, \qquad \text{for } l \geq 0$$

  with the initial conditions

$$u_0 = 1, u_1 = u_2 = u_3 = u_4 = u_5 = 0$$

  to generate N information bits.
- The generated sequence is $100000100001\ldots$ with period 63.

Decoding of Convolutional Codes

程式DEMO方式

- 5/10開始DEMO。
- 私訊檔案給“助教_高子傑”。
- 檔案名稱格式: Project1_學號_姓名.zip 例如: Project1_E94081042_許博士.zip
- 壓縮檔內包含所有的.cpp 以及 一個執行檔 .exe。
- 助教會拿你編譯過的執行檔做測試，再回覆pass or fail。
- 如果fail，最多三次機會(含第一次)，請在5/16 17:00前完成DEMO測試並把修改過的壓縮檔(包含註解)傳給助教。
- 請特別留意，你讀檔跟寫檔的格式是否正確 (請參考上面給的範例Sim.txt、Output_Hard.txt、Output_Soft.txt)

Decoding of Convolutional Codes

注意事項

- 採用Best-state作法。
- Truncation window length = 32代表survivor儲存32 bits。
- ACS遇到同樣大小的metric時，總是選最上面的branch。
- Best-state的選擇，若是遇到同樣的metric，也是選最上面的state。
- State的排列順序爲 $(s_1, s_2, s_3, s_4, s_5, s_6) = (0, 0, 0, 0, 0, 0)$, $(1, 0, 0, 0, 0, 0)$, $(0, 1, 0, 0, 0, 0)$, $(1, 1, 0, 0, 0, 0)$, $\ldots$, $(0, 1, 1, 1, 1, 1)$, $(1, 1, 1, 1, 1, 1)$.
- 如果要使用其他的隨機變數做其他的作用時，請勿使用 **ran1()**.
- **normal()**產生的兩個變數n_1, n_2 要同時使用在AWGN output上。也就是 $y_1 = x_1 + n_1$ and $y_2 = x_2 + n_2$。

Decoding of Convolutional Codes

報告繳交內容

- 5/19 17:00前私訊檔案給“助教_高子傑”。
- 檔案名稱格式: Project1_學號_姓名.pdf 例如: Project1_E94081042_許博士.pdf
- 報告必要內容 (Baseline):
  - 系統架構圖 (細部blcok diagram);
  - 程式流程解釋 (流程或是Pseudo code);
  - 參數設定: Truncation window length = 32等其他參數;
  - 模擬數據 (表格): SNR (dB)、No. decoded bits、No. decided bit errors、bit error rate (BER);
  - 模擬圖: Hard-decision and Soft-Decision;
  - Coding gain at BER $= 10^{-4}$ (Soft-Decision over Hard-Decision).
- 報告加分內容 (Bonus):
  - 不同 Truncation window length的比較;
  - Best-state、fixed-state、majority-vote的比較;
  - 其他觀察或比較。

Decoding of Convolutional Codes

參考數值：

- Hard-decision:
  - BER = 2.1×10^{-3} at $E_b/N_0 = 4.5$ dB;
  - BER = 6.4×10^{-4} at $E_b/N_0 = 5.0$ dB.
- Soft-decision:
  - BER = 2.2×10^{-3} at $E_b/N_0 = 2.5$ dB;
  - BER = 5.3×10^{-4} at $E_b/N_0 = 3.0$ dB.