Lecture 6: Convolutional Code

Scribe: 王傑生、鄭煦 Lecture Date: 4/12, 2017 Lecturer: I-Hsiang Wang

Agenda

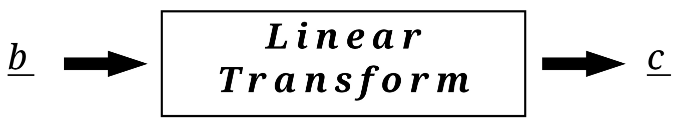
- Convolutional Encoding
- State Transition Diagram and Equivalent Trellis
- Decoding: maximum likeihood sequence detection(MLSD)
- MLSD via Viterbi Algorithm

Recap Last time we showed that ”random” linear block code combined with simple modulation (binary PAM). Can achieve rate-efficient and energy-efficient reliable communication.

- $\underline{b} \cdot \underline{G} = \underline{c}$
- $\underline{b} \in \{0,1\}^{1\times k}$
- $\underline{c} \in \{0,1\}^{1\times n}$
- $k \leq n$
- $rate\ R = \frac{k}{n}$

Introduction to Convolutional Encoding

- In convolutional encoding the Linear Transform is a ”linear filter” (c.t. In Linear block code the Linear trnasform is a ”matrix multiplication”)



Example 1

$\underline{b} = (b_1, b_2, ..., b_n, ...)$

$\underline{c} = (c_1^{(1)}, c_1^{(2)}, c_2^{(1)}, c_2^{(2)}, ..., c_n^{(1)}, c_n^{(2)}, ...)$

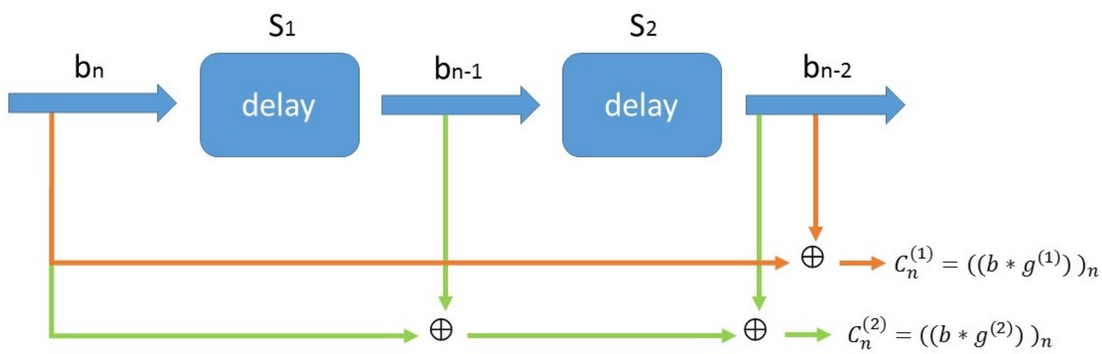
where

$$\begin{cases} c_n^{(1)} = b_n \oplus b_{n-1} = (b * g^{(1)})_n \text{(1st linear filter)} \\ c_n^{(2)} = b_n \oplus b_{n-1} \oplus b_{n-2} = (b * g^{(2)})_n \text{(2nd linear filter)} \end{cases}$$

note: $g_n^{(1)} = \begin{cases} 1, \ if \ n = 0, 2 \\ 0, \ else \end{cases}$ and $g_n^{(2)} = \begin{cases} 1, \ if \ n = 0, 1, 2 \\ 0, \ else \end{cases}$

- Block diagram of the two linear filters:

  1st & 2nd linear filter:



Rate of this linear encoding $R = \frac{1}{2}$

- In the following, we discuss how the 2 bits stored in the 2 shift registers change with the input bit stream.

  observation: length of the FIR filters = number of shift registers + 1

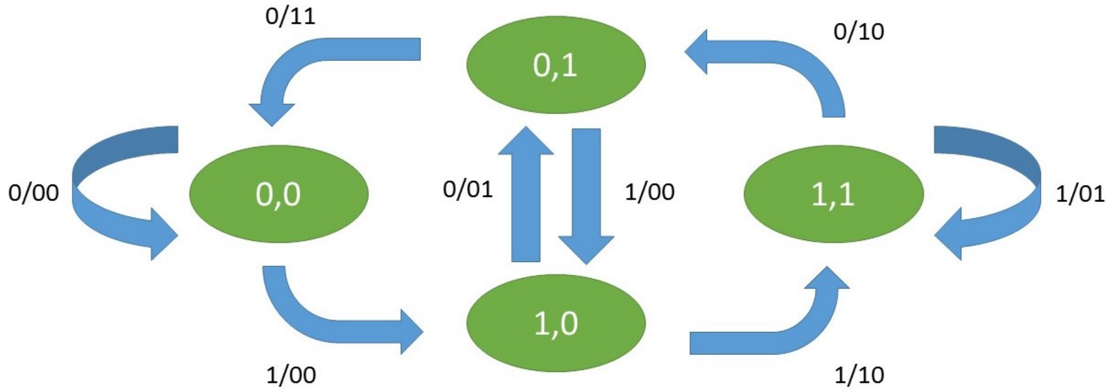
  A running example:

Input b	Before (S1 , S2)	After (S1 , S2)	Output of the encoder
0	00	00	00
1	00	10	11
0	10	01	01
1	01	10	00
1	10	11	10

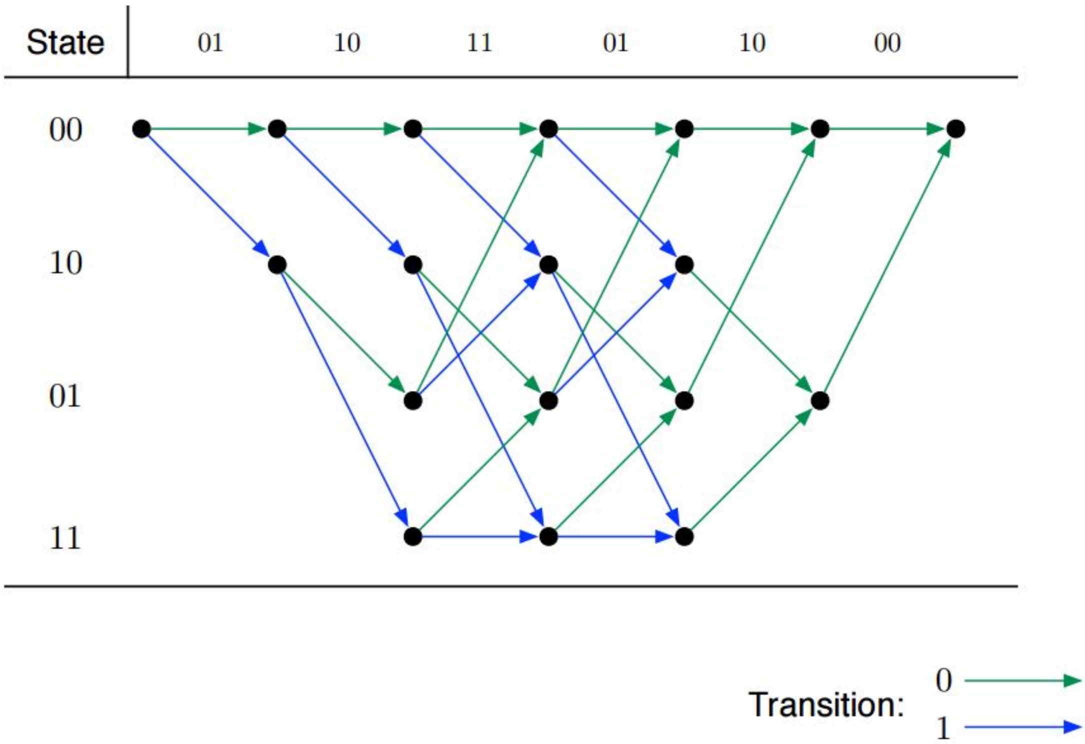
State trasition diagram and Equivalent Trellis

- state transition diagram:

  ”state”: (s_1, s_2)

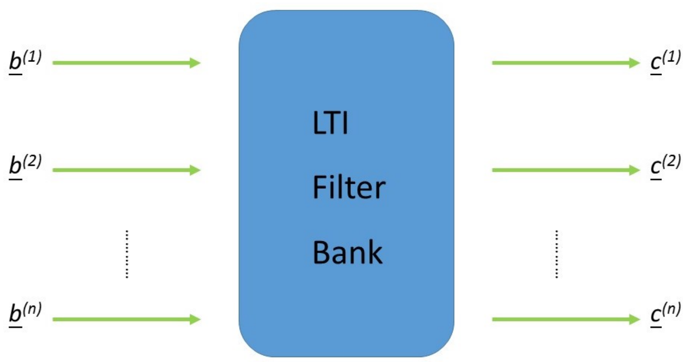


- Trellis Representation:



Generalization

- $\underline{c}^{(l)} = \sum_{i=1}^{k} \underline{b}^{(i)} * \underline{g}^{(i,k)}$



No Feed Back(FIR): G(z) is polynominal

With Feed Back(IIR): G(z) is rational

$R = \frac{k}{n}$

- Termination: In practice, due to packetization, we have to terminate at a given time $K_0 \gg L$ (L: number of shift registers, called "constrained length". When termination happens, one needs to "reset" the registers by appending L 0's at the end. $Rate = \frac{K_0}{2(K_0+L)} \approx \frac{1}{2}$ when $K_0 \gg L$

Decoding Convolutional Code

- Maximum Likelihood Sequence Detection (MLSD):

  Detection Problem:

  given

$$\begin{cases} y_1, y_2, ..., y_n (\text{soft-decision, } y \in \mathbb{R} \ or \ \mathbb{C}) \\ d_1, d_2, ..., d_n (\text{hard-decision, } d \in \{0,1\}) \end{cases}$$

  find the most likely input bit sequence $\hat{\underline{b}}$

  In the previous lecture, we already know that ML = MD where the distance metrics are Euclidean distance in soft-decision, Hamming distance in hard-decision

  thus

$$\hat{\underline{b}} = \begin{cases} arg \ min_{\underline{b}} \parallel \underline{y} - \underline{x}(\underline{b}) \parallel_2 \\ arg \ min_{\underline{b}} \ d_n(\underline{d}, \underline{c}(\underline{b})) \end{cases}$$

- Key observation

  The target function can be decomposed with respect to time

  - 1. For soft-decision, the target function is $\parallel \underline{y} - \underline{x}(\underline{b}) \parallel_2^2 = \sum_{i=1}^{n} |y_i - x_i(\underline{b})|^2$
  - 2. For hard-decision, the target function is $d_H(\underline{d}, \underline{c}(\underline{b})) = \sum_{i=1}^{n} \mathbb{1}\{d_i \neq c_i(\underline{b})\}$ where $\mathbb{1}\{\cdot\}$: indicator function.

  Unify the above two as a minimun cost problem $\hat{\underline{b}} = arg \ min_{All \ length-n \ path \ \underline{p} \ on \ the \ trellis} \{C_n(\underline{p}, \underline{s})\}$

- Viterbi Algorithm (Dynamic Programming for solving minimun cost problem on a trellis)

  let $V_k(s) \triangleq min_{All \ length-k \ path \ \underline{p} \ on \ the \ trellis} \{C_k(\underline{p}, \underline{s})\}$

  $V_{k+1}(s) = min_{r \to s}\{V_k(r) + U(r, s, d_k)\}$

  $P_{k+1}(s) = \arg \ min_{r \to s}\{V_k(r) + U(r, s, d_k)\}$

  where

$$\begin{cases} U(r, s, d_k) = d_H(C(r \to s), d_k) \\ U(r, s, y_k) = \parallel x(r \to s) - y_k \parallel_2^2 \end{cases}$$

  Complexity: linear in n, exponential in L