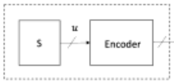
18th lesson

Tuesday, 24 November 2020 11:48

Convolutional codes: encoder

- The encoder of a k, n convolutional code works as n linear filters in GF(2). Each of the n outputs of the encoder is a linear combination of the input
- The filter impulse response for the j-th output bit, the j-th code generator, is a series of 0 and 1.



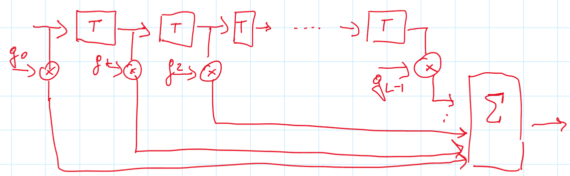
$u \xrightarrow[k]{} \boxed{\text{Block Encoder}} \xrightarrow[n]{} d$ $d = uG$

Convolutional encoder is a linear filter in GF(2)

$x(n) \rightarrow \boxed{g} \rightarrow Y(n)$

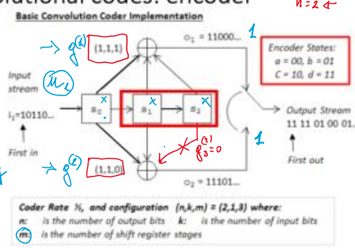
$\rightarrow Y(n) = \sum_{\ell=0}^{L-1} g(\ell)\, x(n-\ell)$

GF(2) ⇒ $Y \in \{0,1\}$, $g \in \{0,1\}$, $x\{0,1\}$



Convolutional codes: encoder

K=1
n=2

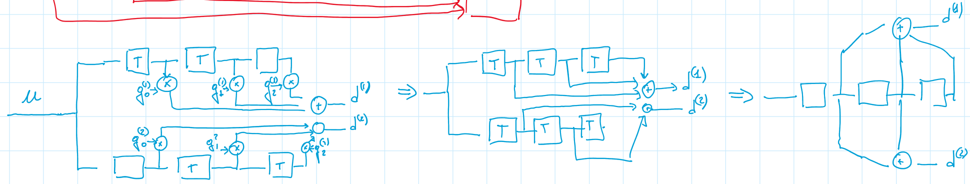


states $2^{(L-1)}$

L constraint length # of memory cells in the encoder

d_ℓ

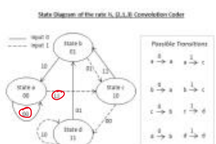
u_ℓ



$u_\ell \rightarrow d_\ell^{(1)}, d_\ell^{(2)}$
$\rightarrow d_{\ell+1}^{(1)}\ d_{\ell+1}^{(2)}$
$\rightarrow d_{\ell+2}^{(1)}\ d_{\ell+2}^{(2)}$
$\}$ L n bits (L=3, n=2) = 6 bits

$u_{\ell+1}$ $d_{\ell+1}^{(1)}\ d_{\ell+1}^{(2)}$
$d_{\ell+2}^{(1)}\ d_{\ell+2}^{(2)}$
$d_{\ell+3}^{(1)}\ d_{\ell+3}^{(2)}$
$\}$ L n

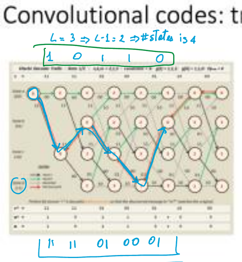
Convolutional codes: state diagram



- The encoder can be represented as a finite-state machine.
- The output of the encoder depends on two elements:
  - The input bit;
  - The state of the encoder: the content of the memory cells of the shift register.
- Each incoming bit determines
  - A new output sequence;
  - A new state
- The state diagram captures the transitions in the encoder.

Convolutional codes: trellis diagram

L = 3 ⇒ L-1 = 2 ⇒ # states is 4



- The time evolution of the encoder is captured by a *trellis* diagram.
- At each signaling time, the trellis represents all achievable states.
  - At time $t = 0$, the encoder starts from state 00
  - At time $t = 1$ the encoder can be either in state 00 or state 10
  - At time $t = 2$ the encoder can be in any of the four states.
  - At any time $t > 2$ the trellis has four states and for each state there are two paths in and two paths out.
- Any given input sequence and the corresponding encoded word can be represented as a path on the trellis.

Convolutional codes: decoder

- Unlike in block codes, the coded bits are not organized in blocks but they are a continuous flow of data.
- A transmission of N codewords implies that the sequence $\boldsymbol{u}$ of kN bits has been encoded into a sequence $\boldsymbol{d}$ of nN bits.
- By using L shift registers at the encoder, each input bit impacts on $L\,n$ coded bits so that the whole sequence needs to be *jointly* decoded.
- The decoder's task is to select among all the possible 2^{kN} possible convolutionally encoded sequences the one that minimize the distance from the received sequence $\widehat{\boldsymbol{x}}$ of nN bits

$$\widehat{\boldsymbol{d}} = \arg\min d(\boldsymbol{d}, \widehat{\boldsymbol{x}})$$

N transmissions ⇒ KN bits encoded in nN bit

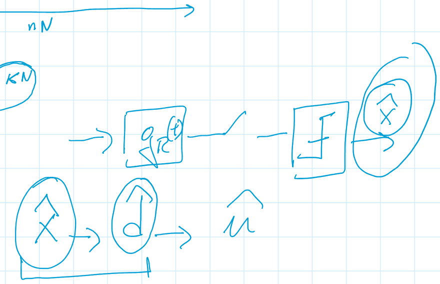
nN

K = 1, n = 2

N = 1000

$2^{1000} = (2^{10})^{100} \approx 10^{100}$

2^{KN}



Convolutional codes: the Viterbi algorithm

- Until the discovery of the Viterbi algorithm in 1967, the use of powerful convolutional codes was limited by the exponential complexity of the decoder.
- The Viterbi algorithm is an iterative algorithm that scales down the complexity from exponential to linear in N.
- The main idea is that of all the 2^{kN} possible paths on the

Convolutional codes: the Viterbi algorithm

- Until the discovery of the Viterbi algorithm in 1967, the use of powerful convolutional codes was limited by the exponential complexity of the decoder.
- The Viterbi algorithm is an iterative algorithm that scales down the complexity from exponential to linear in N.
- The main idea is that of all the 2^{kN} possible paths on the trellis, a very large number of them can be discarded because non relevant.
- At each step of the algorithm, the algorithm selects a number of surviving paths on the trellis equal to the number of states of the encoder.

Andrew James Viterbi (born Andrea Giacomo Viterbi in Bergamo, Italy) is Professor of Electrical Engineering at the University of Southern California's Viterbi School of Engineering, which was named after him in recognition of his **$52 million** gift.

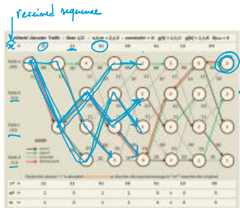
$\hat{x} \rightarrow \hat{d} \rightarrow \hat{u}$

Convolutional codes: the Viterbi algorithm

- The algorithm's objective is to find the path on the trellis that minimizes the distance from the received sequence of bits.
- Starting and finishing in a known state, the algorithm:
  1. Computes all the possible state transitions on the trellis and for each transition the distance from the corresponding received sequence of n bits (*branch metrics*);
  2. Assuming that there is a single path arriving at each state, computes the accumulated distance for each branch out of a given state. The distance is the sum of the cumulated metric of the path arriving at the state and the branch metric;
  3. Discard all the paths leading to one state except the one with minimum accumulated distance (*survivor path*).
- The survivor path leading to the last state is algorithm's output and the final cumulated metric is the number of corrected errors.

$n = 2$

received sequence



4

Given the sequence $\hat{x}$ of nN received bits my objective is to find on the trellis the path which has minimum distance from $\hat{x}$

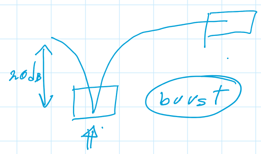
↳ Complexity is linear in N

$2^{L-1} N$

Interleaving

- Convolutional codes are mostly suitable for memoryless channels with random error events.
- Error correcting codes perform well when the errors are uniformly distributed and uncorrelated.
- Fading channels tend to cause bursty errors: when a channel is in a deep fade, there is a statistical dependence among successive error events.
- *Interleaving* makes the channel looks like as a memoryless channel at the decoder and tends to *decorrelate* error events.

S $\xrightarrow{u}$ Encoder → Interleaver $\xrightarrow{d}$ Channel $\xrightarrow{x}$ Deinterleaver → Decoder $\xrightarrow{\hat{u}}$



Time

S → Cod → INT → MAP → $g_T(t)$ → $s(t)$

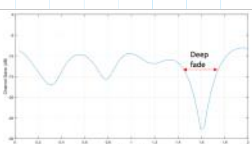
Frequency (OFDM)

S → Cod → INT → MAP → S/P → IDFT → P/S

Interleaving ...

- Interleaving is achieved by spreading the coded symbols in time or frequency before transmission.
- The reverse is done at the receiver by deinterleaving the received sequence.
- Interleaving makes bursty errors look like random, so that convolutional codes perform best.
- The price to pay with interleaving is the large *latency*: both at transmitter and at the receiver it is necessary to have the entire block of data to start the encoding/decoding process.
- There is a trade-off: the larger the interleaver depth, the more decorrelated are the errors but also the longer is the latency and delay.
- Types of interleaving:
  - Block interleaving
  - Convolutional or cross interleaving

Block interleaver example



$\{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P\}$

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

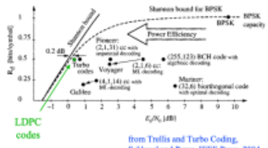
A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	P

$\{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P\}$ → Interleaver → $\{A, E, I, M, B, F, J, N, C, G, K, O, D, H, L, P\}$

4 positions of distance between two consecutively encoded bits

Turbo codes and LDPC

- In the channel capacity theorem, Shannon uses infinitely long codes, i.e. the code rate $R = k/n$ is fixed but $k \rightarrow \infty$ and $n \rightarrow \infty$.
- In practical system, the length of the code is limited because of the complexity of decoding: the performance of physical systems are far from Shannon theoretical limits.
- Around the turning of the century there have been two major breakthroughs:
  - Turbo codes (1993)
  - Low-density parity check codes (LDPC, 1999).



from Trellis and Turbo Coding,
Schlegel and Perez, IEEE Press, 2004