ADVANCED COMMUNICATION SYSTEMS

ELEN90051 (LECTURER MARGRETA KUIJPER)

Introduction to channel coding; convolutional codes

1st Semester 2018

Written by Margreta Kuijper; see Chapters 7 and 8 of "Digital Communications" by Proakis & Salehi, 2008

Figures from the textbook "Digital Communications" by Proakis and Salehi, 2008; to be used with the text of Chapter 8 of the textbook; pages 491-500; 505-523; 548-549

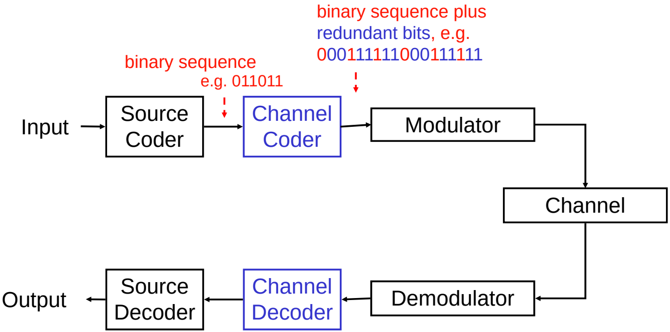
IN PREVIOUS LECTURES:

- The received signal y is different from the transmitted signal x because of channel noise.
- but still reliable communication is possible if the transmission rate is less than channel capacity

IN THIS PART OF THE LECTURE NOTES:

- We need practical channel codes to achieve reliable communication over a noisy channel.
- How? Through transmitting redundant bits
- An (n,k) channel code transmits n coded bits for every k message bits; its **rate** is defined as k/n.
- The purpose is to lower the information bit error probability P_b (as compared to the "no coding" scenario)

Channel coding put into context:



Three different types of channel coding:

- block codes– every block of k information bits is mapped into a block of n coded bits
- convolutional codes– k streams of information bits are convoluted into n streams of coded bits
- trellis coded modulation– error control is combined with modulation

Two different ways of describing codes:

- via matrices and/or polynomials (= [Ch. 7 PS08])
- via trellises and/or graphs (= [Ch. 8 PS08])

TWO MAIN TYPES OF DECODING:

- hard-decision decoding: decoder operates on hard decisions produced by the detector (see sect. 7.5 PS08)
- soft-decision decoding: decoder operates on real numbers, namely the received values produced by the demodulator (so bypassing the detector; see sect. 7.4 PS08)

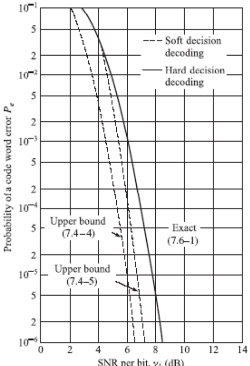


FIGURE 7.6–2
Comparison of soft-decision decoding versus hard-decision decoding for a (23, 12) Golay code.

Up till now restricted to bits. Let's now operate more generally and deal with information symbols, coded symbols, etcetera.

We assume that symbols are from a field $\mathbb{F}$

- $\mathbb{F} = \{0, 1\}$ (considered up till now)
- $\mathbb{F} = \{0, 1, 2, \ldots, 9, X\}$
- $\mathbb{F} = \{000, 100, 010, 001, 110, 011, 111, 101\}$
- and many more....

- A block code such as the (mk, k) repetition code transmits mk coded bits for every block of k message bits; its **rate** is k/mk.
- Block codes are not the only way to insert redundancy. Here we introduce a different type of code, namely convolutional code—its encoder does not chop the information symbol stream up into blocks. Instead, it encodes the whole stream via shift registers.
- Self-reading material is: Chapter 8 of the Proakis and Salehi 2008 textbook; pages 491-500; 505-523; 548-549; 558-571 (downloadable under LMS-"Additional Material").

EXAMPLE OF A CONVOLUTIONAL CODE

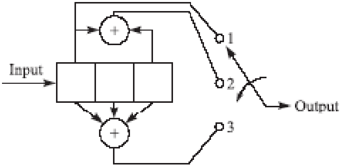


FIGURE 8.1–2
$K = 3, k = 1, n = 3$ convolutional encoder.

Recall from your Signals & Systems subject: such a shift register is alternatively written as a state representation

$$\begin{aligned} x(t+1) &= x(t)A + u(t)B \\ c(t) &= x(t)P + u(t)F, \end{aligned}$$

where $t \in \mathbb{Z}$ is "time". In the above example $x(t)$ is a vector of size 1×2 (state at time t) and $c(t)$ is a vector of size 1×3 (output at time t)

In this example $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 0 \end{bmatrix}$. What are P and F?

IN GENERAL: (n, k) CONVOLUTIONAL CODE SHIFT REGISTER DIAGRAM:

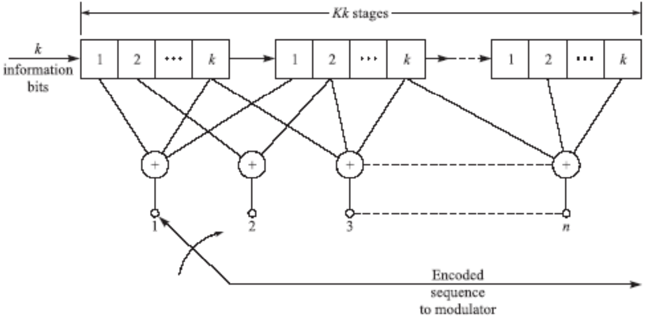


FIGURE 8.1–1
Convolutional encoder.

TRELLIS REPRESENTATION FOR THE EXAMPLE:

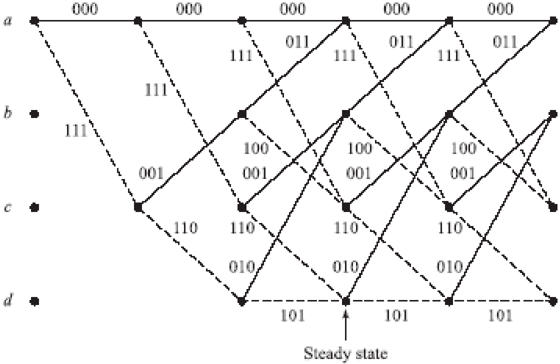


FIGURE 8.1–6
Trellis diagram for rate $1/3$, $K = 3$ convolutional code.

EXAMPLE OF A RECURSIVE CONVOLUTIONAL CODE

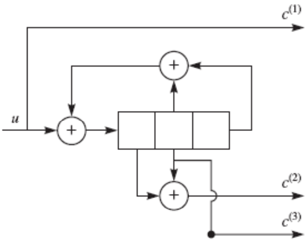


FIGURE 8.1–16
Realization of $G'(D)$ using feedback shift register.

EXAMPLE OF A CATASTROPHIC CONVOLUTIONAL CODE

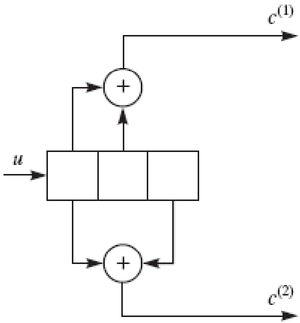


FIGURE 8.1–17
A catastrophic convolutional encoder.

EXAMPLE TRELLIS REPRESENTATION

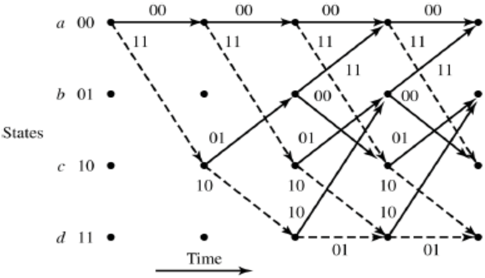


Figure 9.27

Trellis diagram for the encoder of Figure 9.25.

(from “Proakis and Salehi "‘ Communications Systems Engineering"’, 2002)

The generator matrix for this example code is:

$$G(D) = \left[\begin{array}{cc} 1 + D^2 & 1 + D + D^2 \end{array} \right]$$

VITERBI DECODING FOR THIS EXAMPLE:

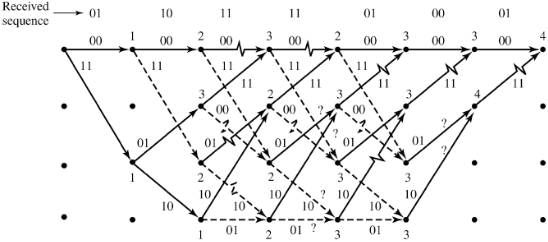


Figure 9.33

The trellis diagram for Viterbi decoding of the sequence (01101111010001).

- here the input sequence to be decoded has length 5 with an extra two padded zeros at the end
- recall: the main idea of the Viterbi algorithm is to choose a survivor whenever two branches enter a state.
- the above figure shows hard decision decoding, it uses bit inputs and the Hamming metric
- alternatively we can do soft decision decoding, it uses real valued inputs and the Euclidean metric.

PERFORMANCE:

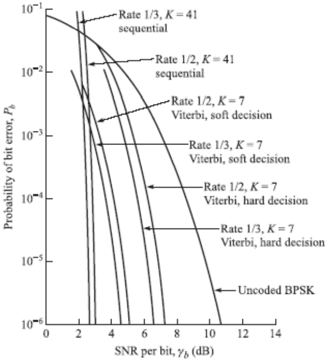
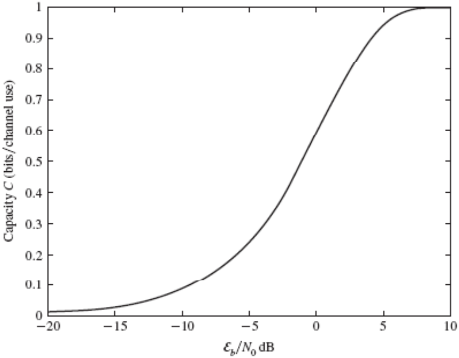


FIGURE 8.6–1
Performance of rate 1/2 and rate 1/3 Viterbi and sequential decoding. [*From Omura and Levitt (1982).* © *1982 IEEE.*]

So to achieve a BER (=Bit Error Rate) of 10^{-5}, the soft decoded rate $1/2$ convolutional code requires an SNR of 4.15 dB, which is a 5.35 dB coding gain compared to uncoded BPSK. How good is this with respect to the Shannon limit?...see next page...

HOW GOOD IS THIS WITH RESPECT TO THE SHANNON LIMIT?

Recall the following Shannon capacity plot from earlier lectures: (using binary antipodal modulation for an AWGN continuous channel, so that bit error probability $p = Q(\sqrt{\frac{2E_b}{N_0}})$)



STRUCTURE OF TURBO CODE:

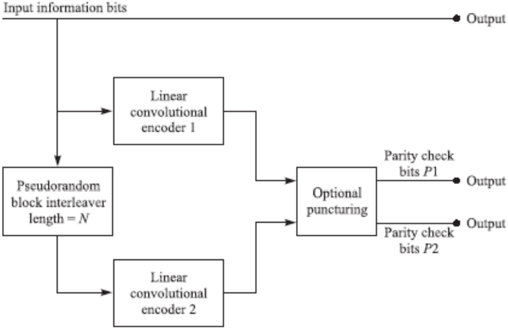


FIGURE 8.9–1
Encoder for parallel concatenated code (turbo code).

PERFORMANCE:

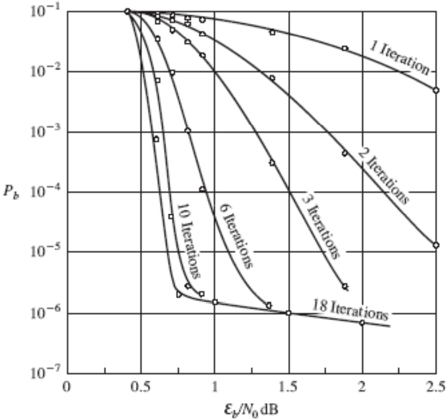


FIGURE 8.9–4
Performance of iterative decoding for turbo codes.

So to achieve a BER of 10^{-5}, the 18 iterations-soft decoded rate $1/2$ turbo code requires an SNR of 0.4 dB, which is only 0.212 dB away from the Shannon limit!

Tutorial Questions on convolutional codes from textbook PS08

- Q 8.1 parts 1 and 3
- Q 8.3 parts 1 and 3