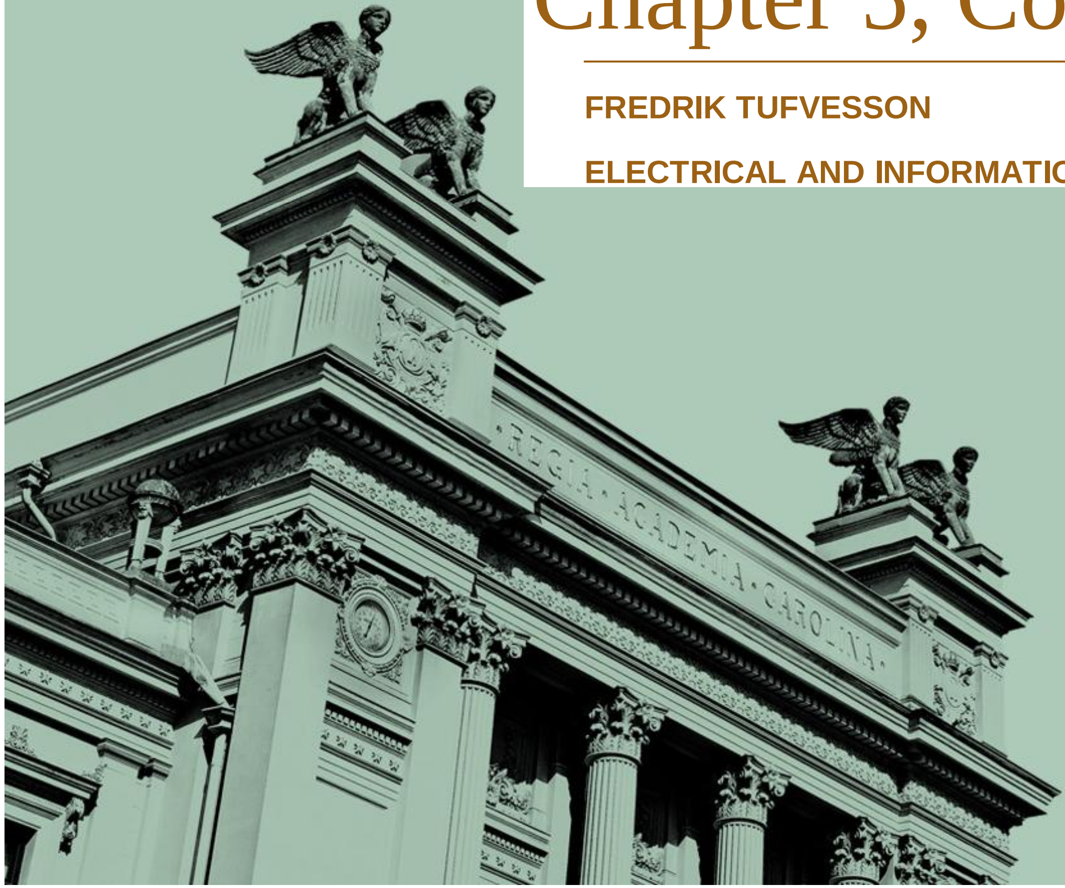
LUND
UNIVERSITY

Information Transmission
Chapter 5, Convolutional codes

FREDRIK TUFVESSON

ELECTRICAL AND INFORMATION TECHNOLOGY



Convolutional codes

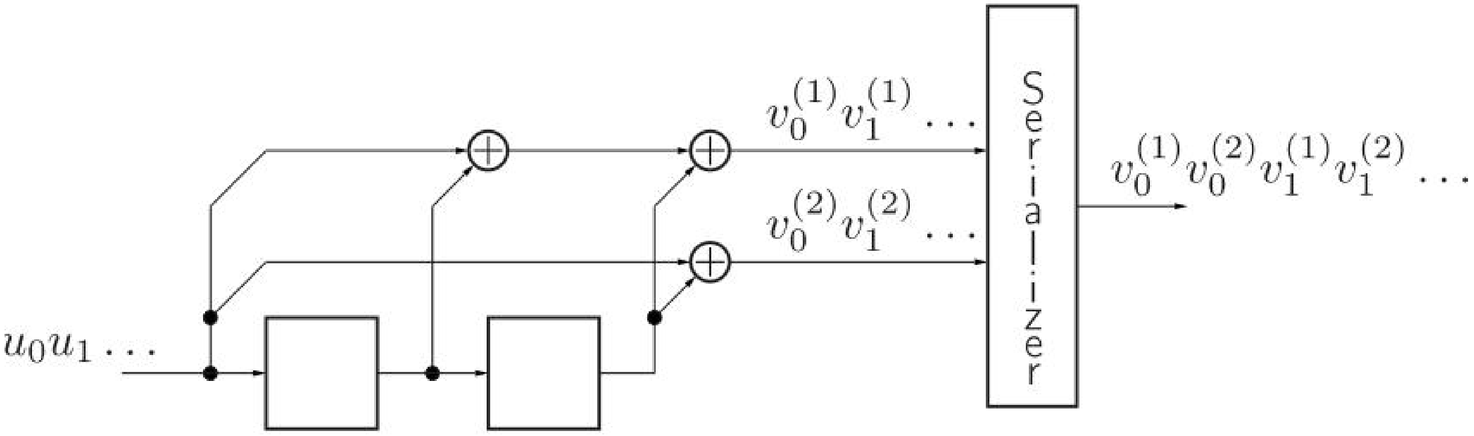
When we study convolutional codes we regard

the *information sequence* $\mathbf{u} = u_0 u_1 \ldots$

and the code sequence $\mathbf{v} = v_0^{(1)} v_0^{(2)} v_1^{(1)} v_1^{(1)} \ldots$

as semi-infinite; they start at time *t*=0 and go on forever.

Consider the convolutional encoder below:


$u_0 u_1 \ldots$
$v_0^{(1)} v_1^{(1)} \ldots$
$v_0^{(2)} v_1^{(2)} \ldots$
Serializer
$v_0^{(1)} v_0^{(2)} v_1^{(1)} v_1^{(2)} \ldots$

The encoder state

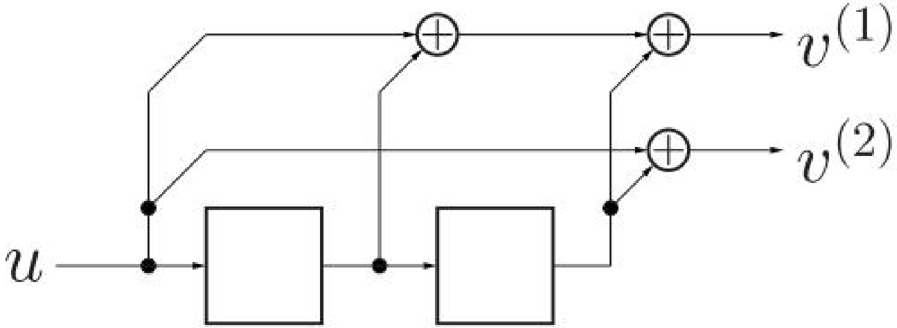
The *state* of a system is a compact description of its past history such that it, together with the present input, suffices to determine the present output and the next state.

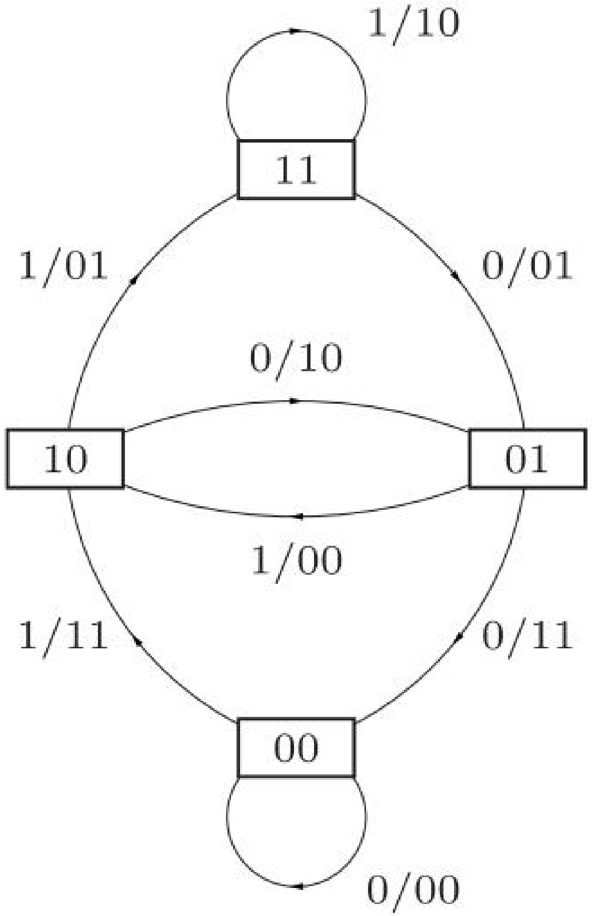
For our convolutional encoder we can simply choose the state to be the contents of its memory element; that is, at time *t* we have the state

$$\boldsymbol{\sigma}_t = u_{t-1}u_{t-2}$$

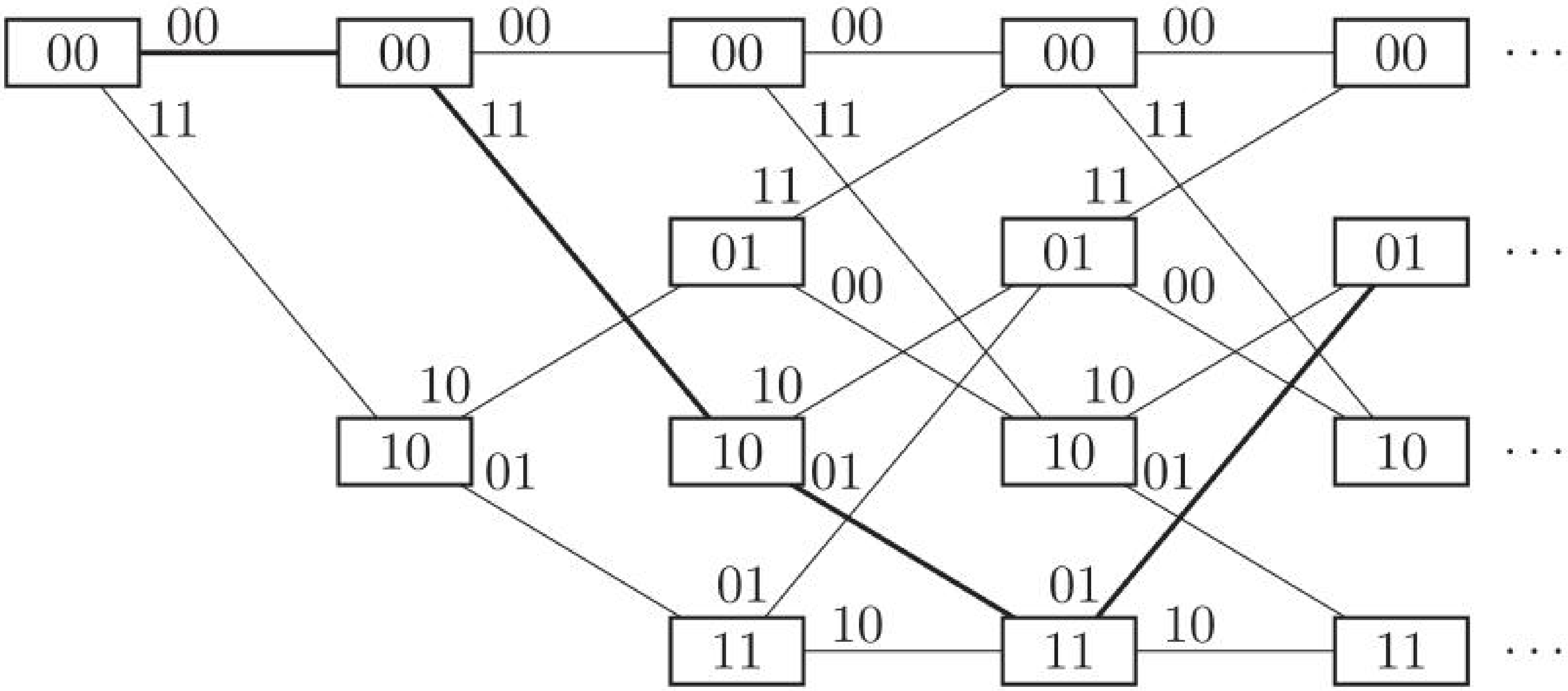
How many stated does our encoder have?

The state transition diagram





The trellis description



The optimal decoder – the Viterbi algorithm

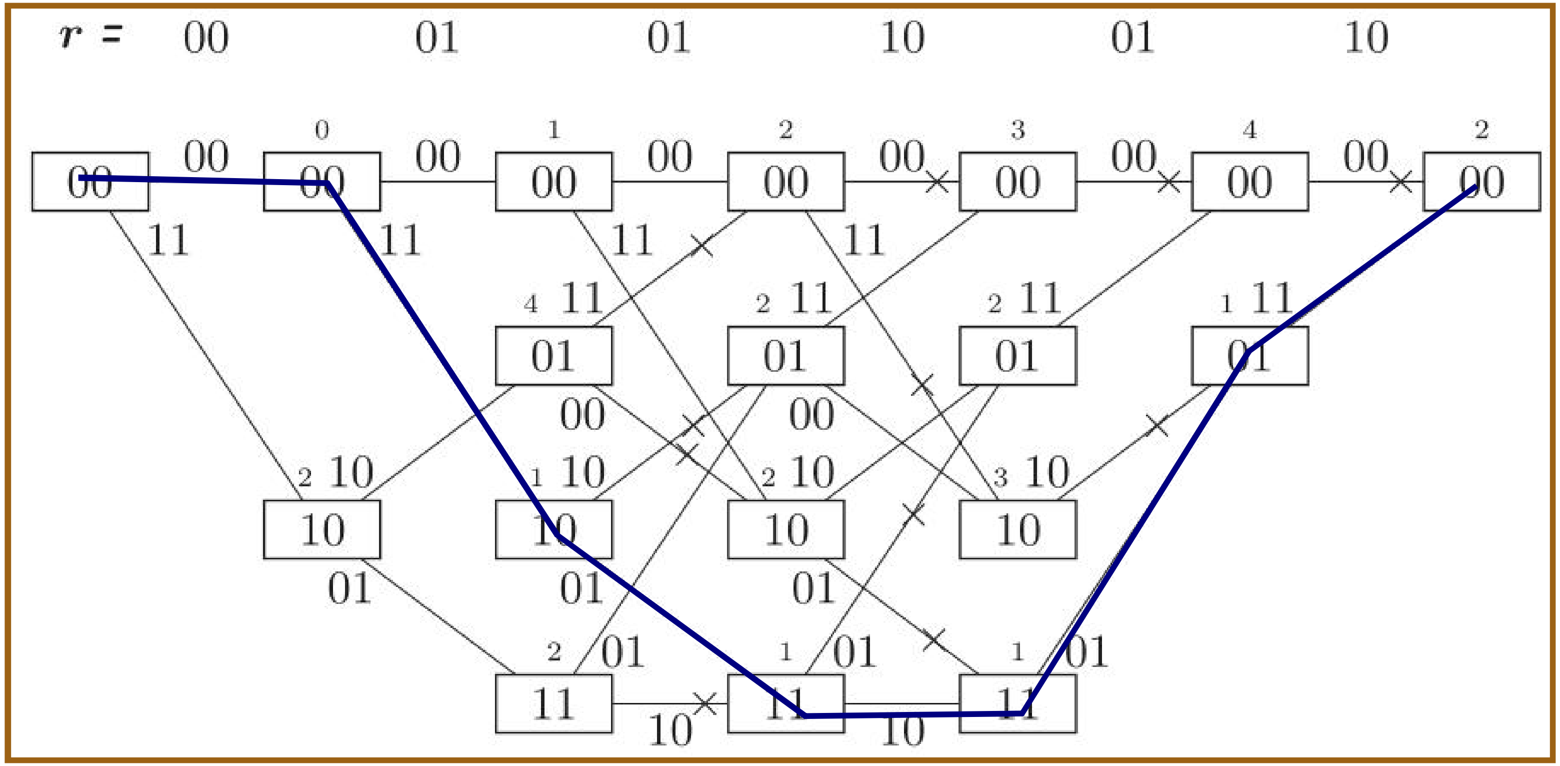
At each state we compare subpaths leading to it and discard the one that is not closest (measured in Hamming distance) to the received sequence.

The discarded path cannot possibly be the initial part of the path $\widehat{\mathbf{v}}$ that minimizes the Hamming distance between the **r** sequence and the codeword **v**.

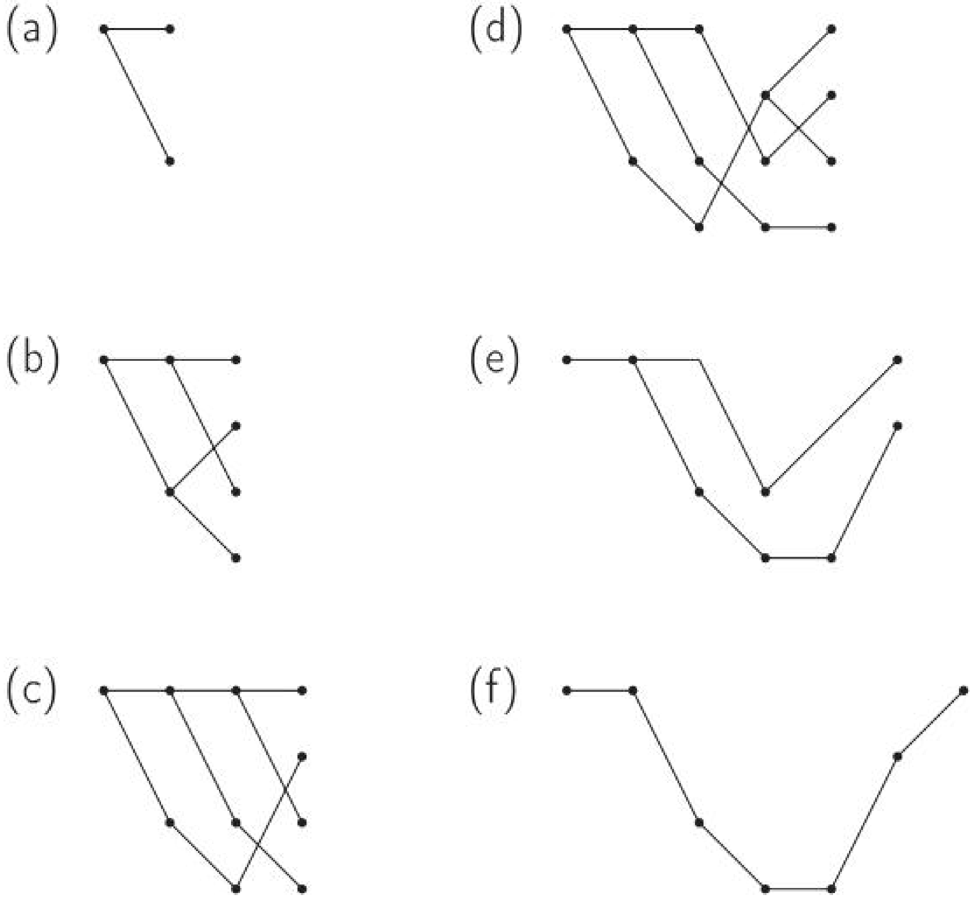
This is the *principle of nonoptimality*.

Let’s consider the previous (7,5) R=1/2 code and a received sequence **r**=00 01 01 10 01 10.

An example


r = 00 01 01 10 01 10
00
0
00
1
00
2
00
3
00
4
00
2
00
00 00 00 00 00 00
11 11 11 11
4 11
01
2 11
01
2 11
01
1 11
01
00 00
2 10
10
1 10
10
2 10
10
3 10
10
01 01 01
2 01
11
1 01
11
1 01
11
10 10

Evolution of sub-paths



Error correction capabilities

We could correct a certain pattern of two errors, namely,

$$\widehat{\mathbf{e}} = \mathbf{r} + \widehat{\mathbf{v}} = (00\ 01\ 01\ 10\ 01\ 10) + (00\ 11\ 01\ 10\ 01\ 11)$$
$$= 00\ 10\ 00\ 00\ 00\ 01$$

where $\widehat{\mathbf{e}}$ is the estimated error pattern.

How many errors can we correct in general?

The answer is related to the minimum Hamming distance between any two codewords in the trellis. Since a convolutional code is linear this value is equal to the least number of ones in any nonzero codeword, this is called *the free distance,* d_{free}

Error correction capabilities

Clearly we can correct all error patterns with$(d_{free} - 1)/2$ or fewer errors.

What about error patterns with more errors?

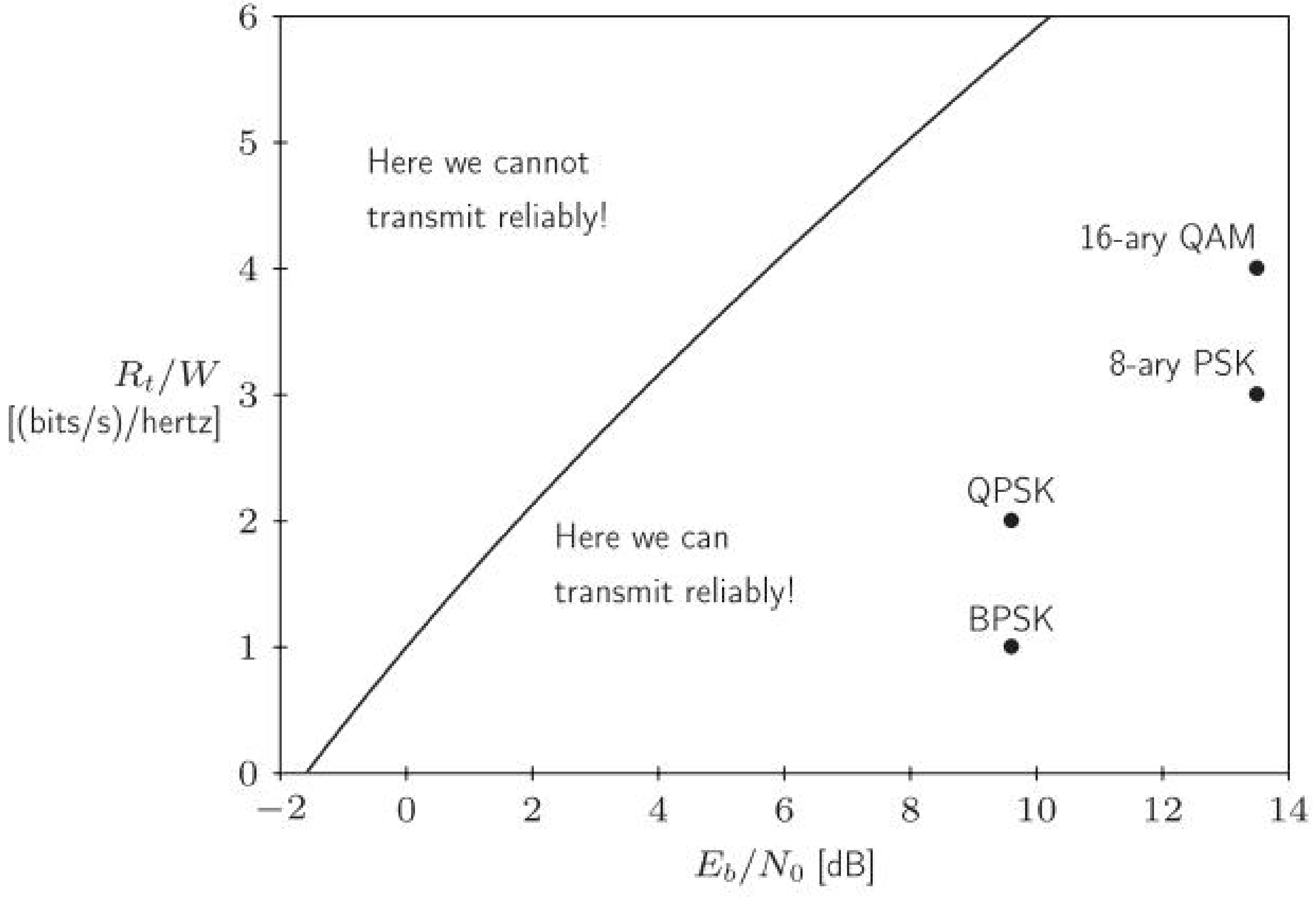
The answer is that it depends; if the errors are sparse enough we can correct many more! That is why convolutional codes are so powerful.

Trellis coded modulation

With trellis coded modulation we combine coding and symbol selection to avoid transmitting symbols that are close to each other in the euclidean space.

Leads to an improved bit error rate performance.

Trellis coded modulation



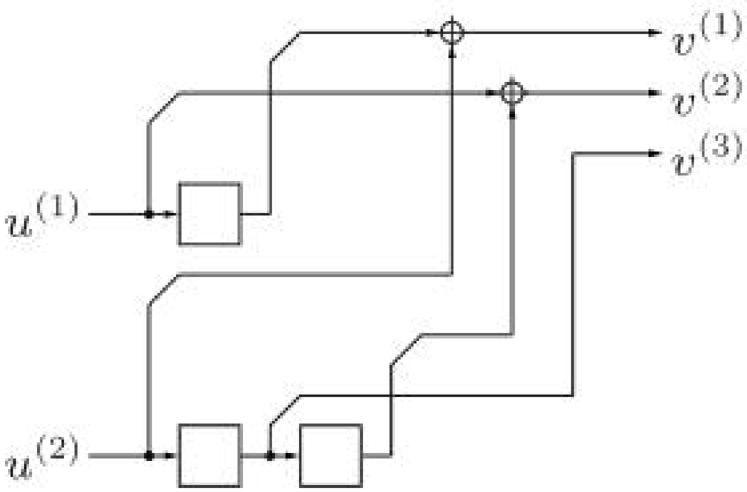
How to maximize error performance

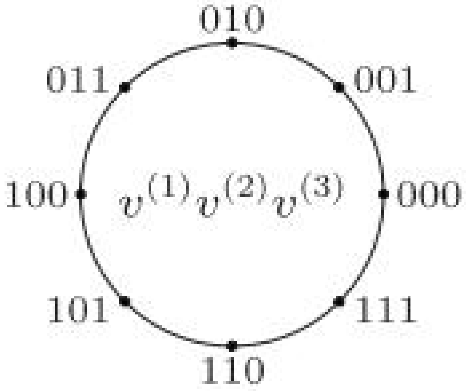
Clearly we can improve the error performance by increasing the signaling energy. Alternatively we can delete some signal points from the constellation and increase the distance between the remaining signal points, but this reduces the data rate.

A more clever approach is, surprisingly enough, to insert new signal points and create a *denser* constellation but at the same time introduce interdependencies between consecutive signal points by coding.

Due to these interdependencies all *sequences* of signal points are not allowed and hence we obtain an increased distance between different sequences of signal points.

Example of a trellis code





Concluding remarks, Chapter 5

Information theory

In this chapter we first gave a brief introduction to Claude E. Shannon's information theory which is the basis for modern communication technology. It provides guidelines for the design of digital communication systems. We then looked at some practical methods of source and channel coding.

Source coding

Shannon modeled sources as discrete stochastic processes and showed that a source is characterized by the uncertainty of its output, *H(U)*, in the sense that the source output sequence can be compressed arbitrarily close to *H(U)* binary digits per source symbol but not further.

The *uncertainty* or *entropy* of a discrete random variable *U* is defined by the quantity

$$H(U) \overset{\text{def}}{=} -\sum_{i=1}^{L} P_U(u_i) \log P_U(u_i)$$

The unit of the uncertainty is called *bit*.

Channel capacity

Shannon's most remarkable result concerns transmission over a noisy channel. It is possible to encode the source at the channel input and decode the possibly corrupted signal at the receiver side such that:

if the source uncertainty is less than the channel capacity, *H(U)<C*, the source sequence can be reconstructed with arbitrary accuracy; This is impossible if the source uncertainty exceeds the channel capacity.

It can be shown that the capacity of a Gaussian channel with energy constraint *E* and noise variance $N_0/2$ is

$$C = \frac{1}{2}\log\left(1+\frac{2E}{N_0}\right) \quad \text{bits/channel use}$$

Huffman coding

Huffman coding is a fix-to-variable length optimal source coding procedure that uses the probability of the source symbols to obtain an encoding for single (or blocks of) source symbols into codewords that consist of variable length strings of binary digits such that the average codeword length is minimum.

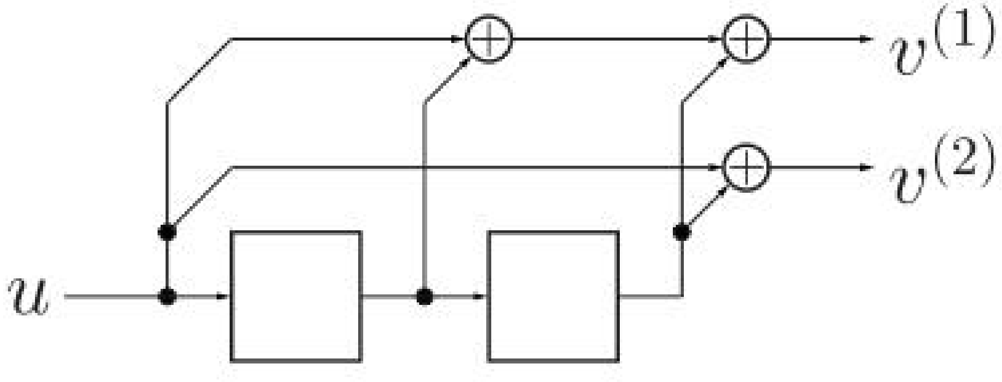
The LZW algorithm

The LZW algorithm is a procedure that does not need to know the source statistics beforehand. It parses the source output sequence, recognizes fragments that have appeared before, and refers to the addresses of these fragments in a dynamical dictionary. This algorithm is asymptotically optimum, easy to implement, and widely used in practice.

Coding methods

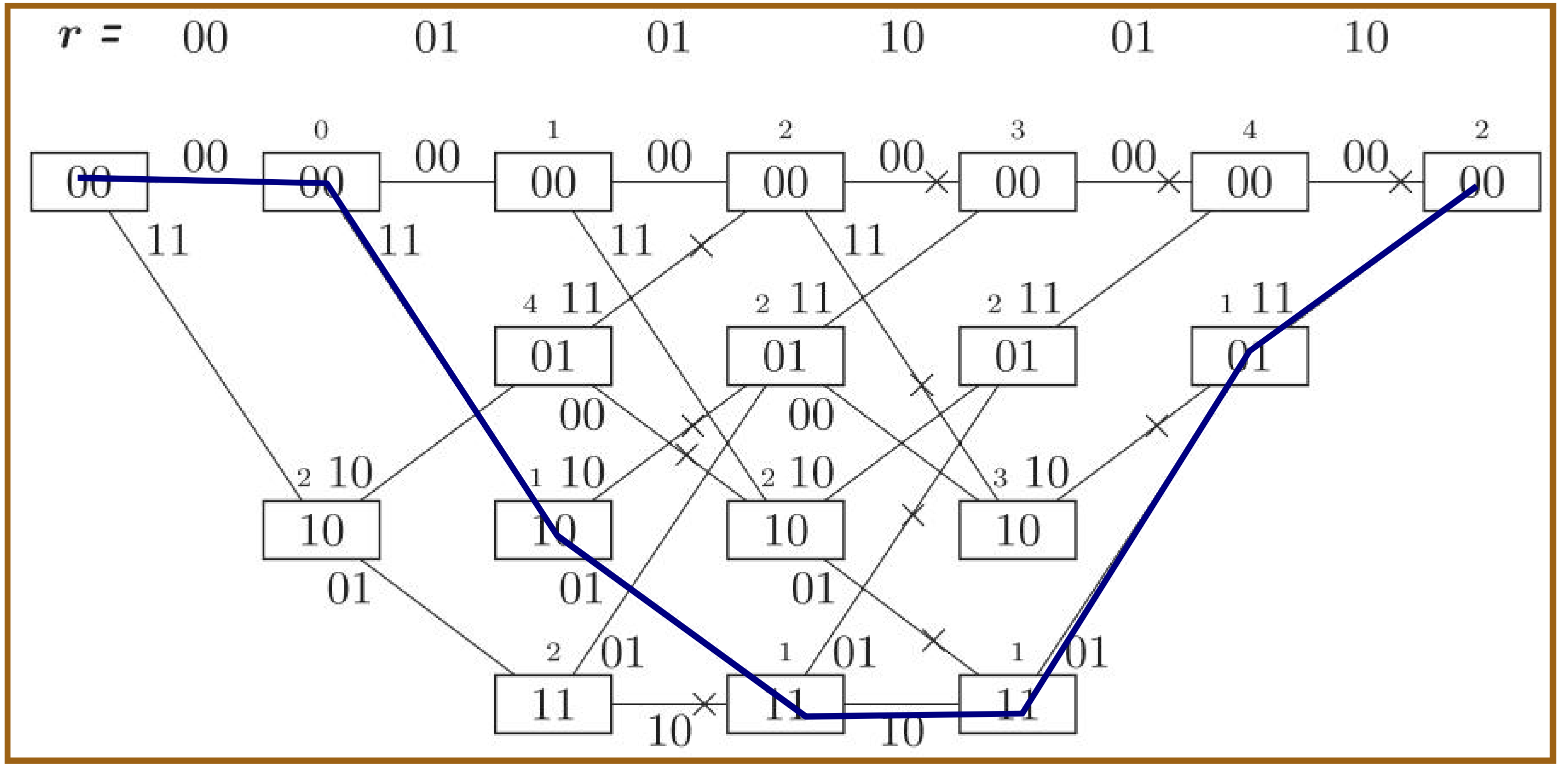
Hamming codes constitute the most celebrated class of block codes. Their minimum distance is d_{min}=3 and, thus, theycorrect all error patterns of single errors.

Convolutional codes are more powerful and often used in practice, either by themselves, as concatenated codes or for parallell codes.



The Viterbi decoder.

Viterbi decoding is a both optimum and practical method for decoding convolutional codes. It is widely used in mobile telephony and high-speed modems.



LUND
UNIVERSITY