SAPIENZA
UNIVERSITÀ DI ROMA

Communications Theory and Engineering

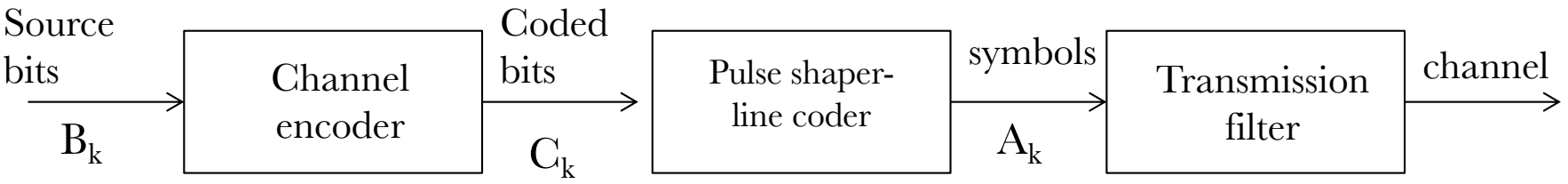
Master's Degree in Electronic Engineering

Sapienza University of Rome

A.A. 2019-2020

Channel Coding

The channel encoder

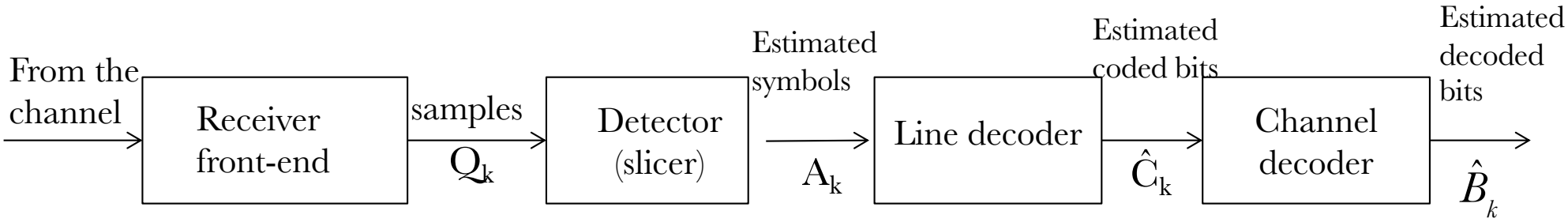


The channel coder is designed for the purpose of

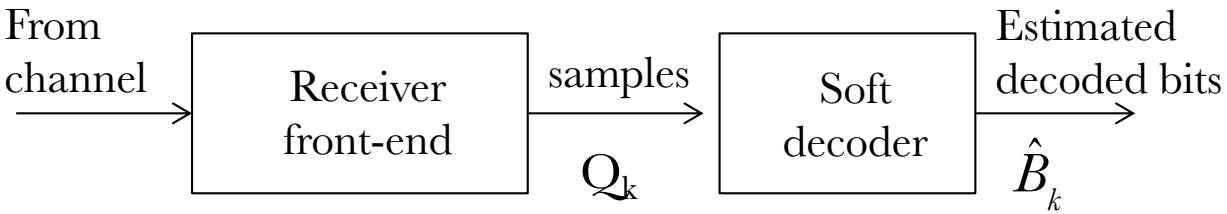
Detect errors and possibly allow retransmission

Correct errors thanks to the introduction of redundancy

Decoding



Hard Decoding



Soft decoding

Error correction codes

Block codes: blocks of k bits are mapped into blocks of n bits, where n>k, in order to introduce (n-k) redundant bits

The block code increases the rate by a factor of n/k

The ratio k/n is called code rate

The code rate represents the fraction of bits corresponding to information bits

Convolutional codes: also in this case the rate increases, but the source bits are not divided into blocks

Performance analysis

Performance analysis consists in comparing the uncoded system against the coded system

> This is done by considering the SNR required at the receiver to achieve a fixed probability of error

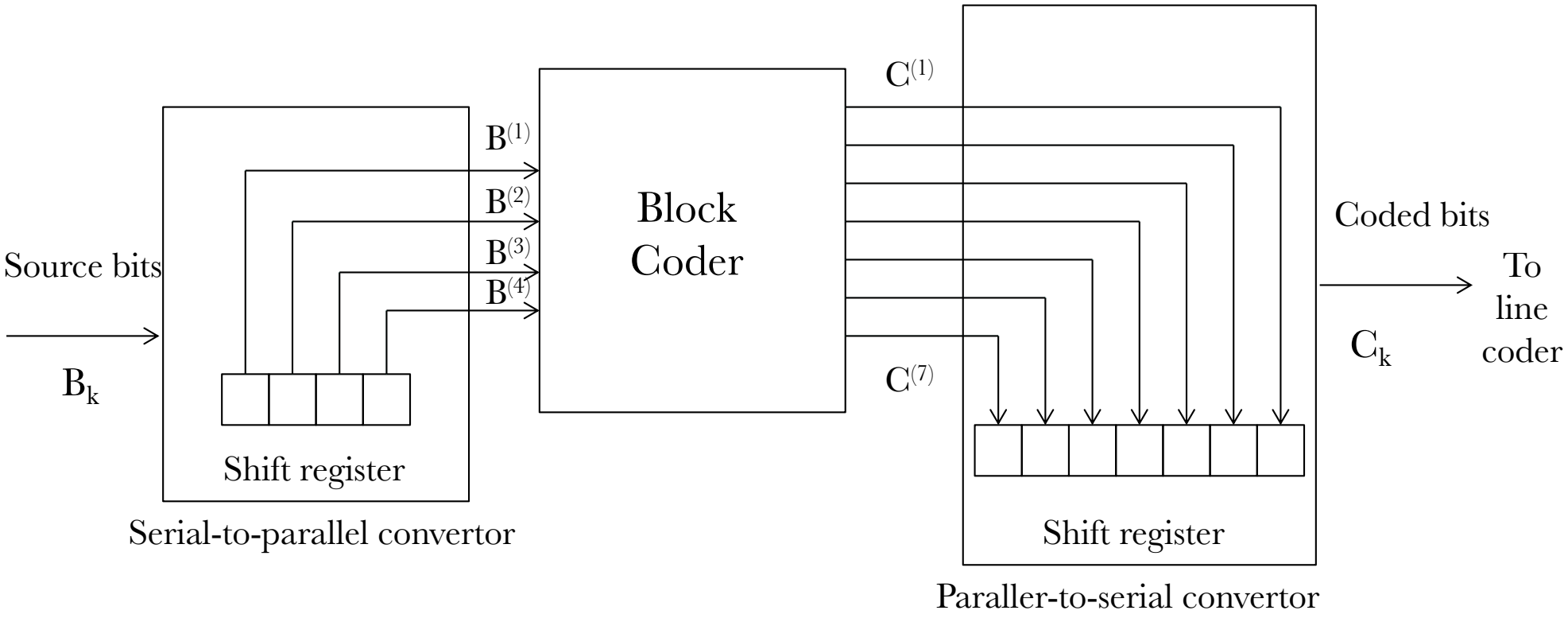
The coded system should be able to tolerate a lower *SNR*!

G code gain

$$SNR_{cod}\Big|_{dB} - SNR_{uncod}\Big|_{dB} = G\Big|_{dB}$$

Block codes

An *(n,k)* block coder maps blocks of *k* source bits into blocks of *n* coded bits.



The block coder consists of modulo 2 adders

Example: parity check code with 1 bit

Parity-check code: the number of "1" in a coded block must be even.

The first k bits of the codeword are the same the source bits

In the codeword, an extra bit is appended to ensure that there are an even number of "*1*"

$$C^{(1)} = B^{(1)}, \cdots, C^{(k)} = B^{(k)}$$
$$C^{(n)} = C^{(k+1)} = B^{(1)} \oplus B^{(2)} \oplus \cdots \oplus B^{(k)}$$

SAPIENZA UNIVERSITÀ DI ROMA

Parity-check codes

In general, in parity-check codes all codewords are modulo-two summations of subsets of the source bits

Representing the input and output bits in row vectors ***b*** and ***c***, then

$$c = bG$$

where all the summations are modulo-two

G is the code generator matrix

A parity-check code has a generator matrix of the form

$$G = \left[I_k \mid P \right]$$

where I_k is the dimension k identity matrix

Codes with G matrices of this type are called systematic

Example: the Hamming code (7,4)

Generator matrix of *(7,4)* Hamming code

$$G = \left[\begin{array}{ccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

Definitions and properties

Given b , $c=bG$ is called a codeword

The set of all the codewords is called code or codebook

Every codeword is a modulo-two summation of rows of the generator matrix

Therefore, the modulo-two sum of any two codewords is a codeword

Parity-check codes form a closed set under the operation of modulo-two summation

Furthermore, all linear codes are equivalent to systematic codes, after reordering columns and elementary operations on the rows of the generation matrix G

Distance and weight of Hamming

Hamming distance

Hamming distance is the number of different elements between two strings

The Hamming distance is therefore the number of variations to be made to convert one string into another

Hamming weight

The weight of Hamming is the Hamming distance from a string consisting of only "*0*"

For linear codes the minimum Hamming distance is equal to the minimum Hamming weight (number of one-bits) among all non-zero codewords

$$d_{H,\min} = \min_{\substack{c \in C \\ c \neq 0}} w_H(c)$$

SAPIENZA
UNIVERSITÀ DI ROMA

Soft vs. hard decoding

In the case of soft decoding the detector chooses the closest codeword in terms of the Euclidean distance measured in symbols

In the case of hard decoding the detector chooses the closest codeword in terms of Hamming distance measured in bits

SAPIENZA
UNIVERSITÀ DI ROMA

Example

Consider a line encoder that maps $\{0,1\}$ into $+a$ and $-a$

The following is true:

$$d_E = 2a\sqrt{d_H}$$

(*) See note for proof

where d_E is the Euclidean distance and d_H is the Hamming distance between a pair of codewords

(*) The square of the Euclidean distance is the square of the distance in one component $(2a)^2$, times the number of the components that differ, which is the Hamming distance. The results follows immediately.

SAPIENZA UNIVERSITÀ DI ROMA

Example

In the case of the parity check code example with *1* bit one has

$$d_{H,\min} = 2 \Rightarrow d_{E,\min} = 2a\sqrt{2}$$

In fact:

The distinguishing feature of all codewords is that they have an even-valued Hamming weight

The smallest Hamming weight among all non-zero codewords is two

The smallest Hamming weight is also the minimum Hamming distance between codewords

Performance of soft decoders

Discrete-time additive Gaussian noise channel

C is a codeword that is transmitted as a vector with binary antipodal components

The noise samples have variance σ_c (subscript indicates "coded")

The received samples Q_k can be collected in a vector q

$$q = a + n$$

where n is a vector of i.i.d. Gaussian random variables

The maximum likelihood detector selects the vector â in Ω_a such that the Euclidean distance between â and the observed vector q is minimum

Let $d_{E,min}$ be the minimum Euclidean distance between the transmitted vector and all other vectors in Ω_a the following lower bound is true:

$$\Pr_{block\,error} \geq Q\left[\frac{d_{E,\min}}{2\sigma_c}\right] \qquad Q[X] = \frac{1}{2}erfc\left\{\frac{X}{\sqrt{2}}\right\}$$

If there are few codewords at distance $d_{E,min}$, then the probability of error is close to this limit

Performance of soft decoders

A crude upper bound on the number of codewords of distance $d_{E,min}$ from any codeword is $2^k - 1$, hence an upper bound on the block error probability is

$$\Pr_{block\,error} \le \left(2^k - 1\right) Q\left[\frac{d_{E,\min}}{2\sigma_c}\right]$$

and therefore

$$Q\left[\frac{d_{E,\min}}{2\sigma_c}\right] \le \Pr_{block\,error} \le \left(2^k - 1\right) Q\left[\frac{d_{E,\min}}{2\sigma_c}\right]$$

SAPIENZA UNIVERSITÀ DI ROMA

Example

In the parity check code with *1* bit we know that

$$d_{H,\min} = 2 \Rightarrow d_{E,\min} = 2a\sqrt{2}$$

and therefore

$$Q\left[\frac{a\sqrt{2}}{\sigma_c}\right] \leq \Pr_{block\,error} \leq \left(2^k - 1\right) Q\left[\frac{a\sqrt{2}}{\sigma_c}\right]$$

$$Q\left[X\right] = \frac{1}{2} erfc\left\{\frac{X}{\sqrt{2}}\right\}$$

SAPIENZA UNIVERSITÀ DI ROMA

Coding Gain

In the uncoded case, we do not refer to the symbol but to the P_e of the information bits

In the case of antipodal transmission +a and −a one has

$$\Pr_{bit\,error,uncoded} = Q\left[\frac{a}{\sigma_u}\right]$$

σ_u is the Gaussian noise variance at the slicer input for the uncoded system

Note: depending on the medium, modulation technique, and bit rate, σ_u and σ_c may not be equal

SAPIENZA
UNIVERSITÀ DI ROMA

Coding Gain

$$\Pr_{bit\,error,\,soft\,decoding} \geq \frac{1}{k}\Pr_{block\,error,\,soft\,decoding}$$

$$\Pr_{bit\,error,\,soft\,decoding} \leq \Pr_{block\,error,\,soft\,decoding}$$

SAPIENZA UNIVERSITÀ DI ROMA

Example

Parity check code with 1 bit

One can estimate the probability of bit errors by setting the arguments of *Q(.)* equal (ignoring the constant multipliers)

$$\frac{a_c\sqrt{2}}{\sigma_c} = \frac{a_u}{\sigma_u}$$

and so

$$SNR_u = 2SNR_c$$

The coding gain is therefore *3 dB*

SAPIENZA
UNIVERSITÀ DI ROMA

Code Gain

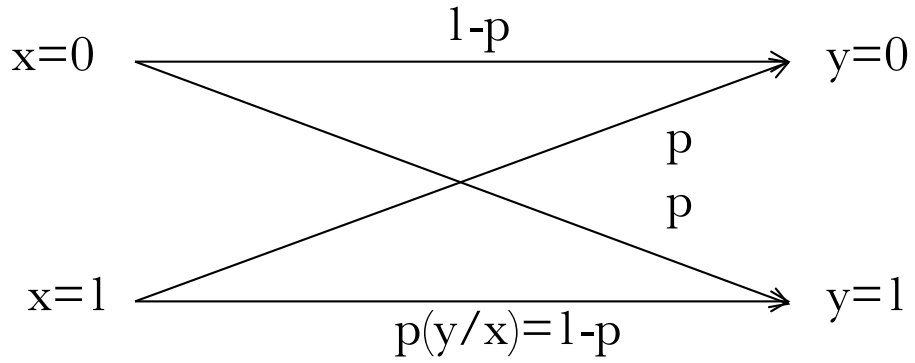
But the coded system has a rate of $n/(n-1)$ so if you consider that you spend an energy $n.a^2$ to send $n-1$ information bits, the energy you spend for one information bit is $a^2 .n/(n-1)$.

So if $n=3$ the power required for the coded system is *1.25 dB* smaller than for the uncoded system

The coding gain is *3 dB* but *1.75 dB* is lost due to the presence of redundancy bits.

Performance for Hard decoders

For a hard decoder, the decoding takes place after the slicer and the equivalent binary channel after the slicer can be usually modeled as a BSC



Let c* denote the bits emerging from the BSC

A decision maker selects the code word ĉ closest to c* in terms of Hamming distance

Example: the Hamming code (7,4)

c=*0000000* (transmitted codeword) c^*=*0001010* (received codeword)

$\hat{c}$=*0001011*

The detected codeword is $\hat{c}$=*0001011*, which is closer in Hamming distance than the all zero codeword.

But the question is: how many bits can be corrected by such a code?

Correction capability

How many errors can be corrected by a hard decoder detector?

If c* has fewer than

$$t = \left\lfloor \left(d_{H,\min} - 1 \right) / 2 \right\rfloor$$ Errors, then those errors can be corrected

For example, for the parity check code with 1 bit, t =0

While for the Hamming code (7,4), t = 1

One can be certain of correcting up to t bit errors, but some error patterns with more bit errors than t may be correctable also, unless the code is a so-called perfect code

SAPIENZA UNIVERSITÀ DI ROMA

Perfect code

A perfect code is such that

All bit patterns of length n are within Hamming distance t from a codeword

No bit pattern of length n is at a Hamming distance less or equal than t from more than 1 codeword

The Hamming code *(7,4)* is a perfect code

The seven-bit patterns are either a codeword or one bit distant from exactly one codeword.

Therefore if one sends c and there are 2 bit errors then c* is at distance 1 from a code $\hat{c} \neq c$ and thus there is error

Perfect codes are optimal on the BSC in the sense that they minimize the probability of error among all codes with the same n and k

SAPIENZA UNIVERSITÀ DI ROMA

Perfect code

In the case of perfect code it is easy to determine the performance of a hard decoder

The probability of m bit errors in a block of n bits is a binomial distribution

$$P(m,n)=\binom{n}{m}p^m(1-p)^{n-m}=\frac{n!}{m!(n-m)!}p^m(1-p)^{n-m}$$

In the case of perfect codes there is a symbol decoding error if more than t bits are incorrect

$$\Pr_{blockerror}=\sum_{m=t+1}^{n}P(m,n)=1-\sum_{m=0}^{t}P(m,n)$$

SAPIENZA UNIVERSITÀ DI ROMA

Example

The Hamming code *(7,4)* is a perfect code

$$\Pr_{block\,error} = 1-(1-p)^7 - 7p(1-p)^6$$

For $p=10^{-2}$ the probability of block error is about $2*10^{-3}$

Coding seems to reduce the probability of error by a factor of 5,
If $p=0.01$, then $Pr_{block\ error}=0.002$

However, the coded system requires a bandwidth *7/4* times larger than the uncoded system and a more accurate evaluation shows that the true gain is very small

SAPIENZA UNIVERSITÀ DI ROMA

For non-perfect codes

In the case of non perfect, some patterns with more than t bit errors can be corrected and an upper bound is:

$$\Pr_{blockerror} \leq \sum_{m=t+1}^{n} P(m,n)$$

In practice, many codes are quasiperfect, meaning that although some error patterns with t+1 bit errors can be corrected, none with t+ 2 or more can be corrected. For these we can get a lower bound

$$\Pr_{blockerror} \geq \sum_{m=t+2}^{n} P(m,n)$$