Lecture 6: Linear Code Blocks

Shannon Basic Ideas

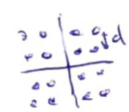
* Rate = R
Channel Capacity } If $R<C \Rightarrow P(E) \sim 0$ but if $R>C \Rightarrow$ Unnaceptable Performances

* To reach good performances:
↳ Dimension of signal vector space $N \rightarrow N$ very big
↳ Use SOFT Decisions at the receiver
1) Shoft Decision ⟶ Real values
2) Hard Decision ⟶ Quantized values ("1" "0")

* Shannon demonstrated the existance of these codes, but he didn't achieve any implementation

Classic Codes

* In the classic modulation systems we want to increase the distance between signals

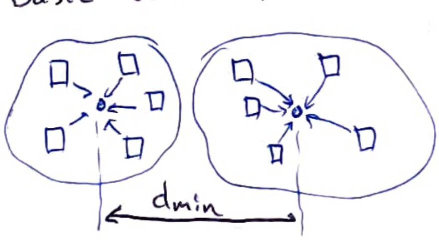


The performance stays far from the Shannon's limits
↳ To increase de distance(d) ⇒ More power or more BW or both
↳ Increasing the number of dimentions N ⇒ The system get very complicated

* To improve P(E) ⇒ concatenation in time domain of many symbols with some rules able to increase d_{min}
↳ The complexity has to be linear with N (to allow receiver demod)

* Classical solutions
↳ Linear Block Codes
↳ Convolutional Codes

* Basic idea of channel coding:

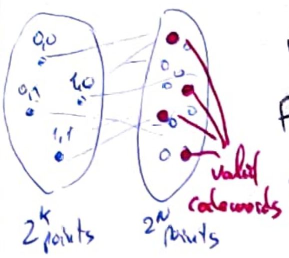


• codewords
□ received sequence with the errors

Each point is related to a signal. If the signal has errors I cannot understand what happened
Instead of using any possibility I use a "codeword" for each set
The □ are possible signals but due to the code they are not allowed. Only codewords • are allowed
Choosing one codeword in a group ⇒ increase d_{min} ⇒ decrease P(E)

Block Codes

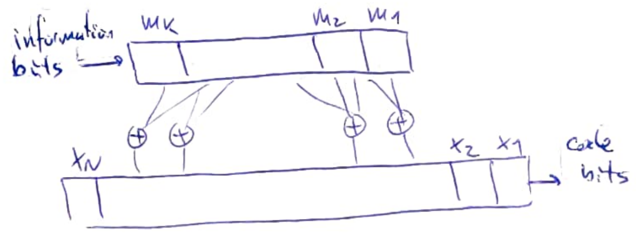
* **Basic Idea:** Starting from a vector with K-elements I will map the vector in other set of N-elements



From 2^k points I can send maximum 2^N elements
From this 2^n set I select a subset to be the valid codewords, if I can distinguish the possibilities ⇒ I can increase the d_{min} ⇒ decrease $P(E)$

* **Code Rate:** $\frac{K}{N} < 1$

* **Shift Register Description**
  - ↳ We have a shift register with K-bits ⇒ My Information
  - ↳ This K-bit vector is transformed in a new vector of N-bits $(N > K)$
  - ↳ The codewords are obtained by a linear combination (boolean: $1+1=0$) of the information bits.



↳ Almost all block codes used today belong to the subset: Linear Block Codes.

* **Mathematical Description**
  - ↳ An element of the codeword X_i is obtained: $\boxed{X_i = \sum_{j=1}^{K} g_{ji} m_j}$
  - ↳ m_i message information
  - ↳ $g_{ji} = \{1, 0\}$, $i = 1, \dots, N$ — g_{ji} are the connections

* **Matrix Description**

$\bar{X} = \bar{m}\, G$

$\bar{X}$ → vector with the elements of the codeword

G → generator matrix, dim → $K \times N$

⇒ Parity Check Codes (or Systematic Code)

↳ Special arrangement of the vector:
At the beginning: information
At the end: parity check control

↳ The Generation matrix:

Example:

$$G = \underbrace{\begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}}_{I_K}\!\!\underbrace{\begin{matrix} 0&1&1 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{matrix}}_{P} = |I_K\ P|$$

when systematic code ⇒ $G = |I_K\ P|$
↳ Parity Matrix
↳ K order unitary matrix

Fundamental Relation

$G = |I_K\, P|$

* We work with a systematic code $\bar{x} = \bar{m}\,G = |\bar{m}, \bar{m}P|$
* We can generate a new matriz: Parity Check Matrix: H^T $H^T = \begin{vmatrix} P \\ I_{N-K} \end{vmatrix} \rightarrow (H^T)^T = [P \vdots I_{N-K}]$

$\Rightarrow$ Fundamental relation: $\boxed{\bar{x}\,H^T = 0}$

* Demonstration:

$$x\,H^T = |\bar{m}, \bar{m}P| \begin{vmatrix} P \\ I_{N-K} \end{vmatrix} = \bar{m}P + \bar{m}P = 0$$

$\uparrow$ boolean

H^T, the parity check matrix is able to describe the rule of the codes
If $\bar{x}$ is a codevector consistent with the rule of the codes, using this relation we can check this consistency
- $\hookrightarrow$ If $x\,H^T \neq 0 \Rightarrow$ We have errors
- $\hookrightarrow$ If $x\,H^T = 0 \Rightarrow$ The word belongs to the subset but we don't if it's the correct one

Statements for Linear Code Blocks

* If K is the dimension of the block, the number of codewords is 2^K so there are 2^K distinct messages
* The set $\{g_i\}$ of vectors are linearly independent $\Rightarrow$ Can be used as a basis in vector space called C, which dimension is K
* The rows ($\{g_i\}$) of the generator matrix are valid codewords

Basic Definitions

* Def: The weight of a codeword c_i denoted by $w(c_i)$ is the number of nonzero elements in the codeword
* Def: The minimum weight of a code W_{min} is the smallest weight of the nonzero codewords in the code
* Theorem: In any linear code: $d_{min} = W_{min}$
* Hamming distance $d_H(a,b)$: Given two codes words a, b is the number of different bits in a and b
* Hamming weight $W_H(a)$ = weight

Decoding Strategy: Maximum Likehood Decoding Strategy (Hard Decision)

The decoding strategy tries to maximize the likehood function. We consider that the symbols show the same probability.

⇒ The maximization of the likehood function is the same as the minimization of the distance between the received codeword and the possible codewords

This strategy will be the optimal strategy

⇒ The minimum distance is the same as the minimum codeword weight and the minimum distance is limited to $d^* \leq N-k+1$

* Correcting power:
A code can correct h errors if and only if $d^* \geq 2h+1$ → correcting power of the code ⇒ $t = \left\lfloor \frac{d^*-1}{2} \right\rfloor$

Syndrome

We use the concept of syndrome will tell us if there are errors in the transmission

$x \rightarrow (\text{channel}) \rightarrow \bar{y} = \bar{x} + \bar{e}$

At receiver:

$$\bar{y} H^T = (\bar{x} + \bar{e}) H^T = \underset{\uparrow\ \bar{x}H^T=0}{} \bar{e} H^T = \begin{cases} \Rightarrow \bar{e}H^T \neq 0 & \text{we have some errors} \\ \Rightarrow \bar{e}H^T = 0 & \text{the received word is a codeword but we don't know if it is correct!} \end{cases}$$

$H^T = \begin{vmatrix} P \\ I_{N-k} \end{vmatrix}$ $\quad \bar{x}H^T = 0$

⇒ The system can detect errors which are not codewords!, when $s=0$ nothing can be assumed

↳ Example

We have min. dist = 3 looking at the codewords

min dist = 3 → correct $d=1$, detect $d=2$

We see all the possible sequences
the seq. with $d=1$ ⇒ I can correct errors
the seq. with $d=2$ ⇒ I can dectect erros

Strategy to use the syndrome: Given one syndrome we have many posibilities of errors related to that syndrome because adding a codeword we obtain the same syndrome

⇒ We assume that it's more probable to have small errors than big errors

Starting from one syndrome we list all the possible errors that could produce this syndrome and select the one with the minimum weight (more probable)

⇒ The correction: To add to the received signal the more probable error

Cyclic Codes

* It is a subset of linear codes.

* If c is a codeword of a cyclic code then if we shift the elements of c what we obtain is also a codeword
  Also the linear comb. of codewords is a codeword.

* The cyclic codes are important because they give us a decoding strategy that is feasible

* Polynomial representation

Being the codeword $\bar{a}$: $\bar{a} = (a_1, a_2, \dots, a_N) \Rightarrow a(D) = a_1 D^{N-1} + a_2 D^{N-2} + \dots + a_{N-1} D + a_N$

↳ word of length $N \Rightarrow$ polynomial degree $N-1$

* Generator polynomial $g(D)$ with the form

↳ Given a cyclic code (N, K) exists a unique polynomial of degree $(N-K)$ that is able to generate all the codewords

$\boxed{g(D) = D^{N-K} + \dots + 1}$

↳ All the other codewords are multiples of $g(D)$

, All the polynomial of degree less or equal to $N-1$ which are divisible by $g(D)$ are codewords

⇒ The generator polynomial of a cyclic code have to be a divisor of (D^N+1)

⇒ Every divisor of (D^N+1) of degree $N-K$ generates a cyclic (N,K)

↳ Generating a code

→ Polynomial associated to the information word $m(D) = m_1 D^{K-1} + \dots + m_K$

→ Generator polynomial $g(D)$ → degree r

→ We obtain the codeword $X(D)$ associated to $m(D)$

$$\boxed{X(D) = g(D) \cdot m(D)}$$

→ the codeword is cyclic if is a divisor of $D^N + 1$

Usually the code is not systematic, We can obtain a systematic one doing:

$$X(D) = m(D) \cdot D^{N-K} + \text{reminder}\left\{\frac{m(D) \cdot D^{N-K}}{g(D)}\right\}$$

* Every codeword is divisible by $g(D)$
  ↳ If not divisible ⇒ identify errors
  ↳ Reminder related to syndrome

Hamming Codes

* It is a one error correcting codes
* They are: $N = 2^{N-k} - 1$ → i.e. (7,4), (15,11),
  ↳ with different generator polynomials: D^3+D+1 ; D^4+D+1 ; D^5+D^2+1
* Minimum distance always equal to 3 $d_{min} = 3$
  ↳ If N big P(E) is not very good
  ↳ If N small performance not so interesting

* BCH codes

An extension of Hamming codes
⇛ They are able to correct any error we would like to correct.
For any positive integers m and t there is a t-error-correcting binary BCH code with

$n = 2^m - 1$

$k \geq n - mt$

In order to correct t errors the minimum Hamming distance is $d_{H,min} \geq 2t+1$

* Reed-Solomon codes: special case of BCH

The symbols instead of being composed by "1" and "0" are blocks of bits:

⇒ Used to correct burst of errors.

Expurgating

A cyclic code with an odd minimum distance can be expurgated by:
multiply generator polynomial by $D+1$

The words in the expurgated code have an even number of ones
The expurgated code is cyclic

Shortening

The first b positions are reset, therefore this data is not transmitted
The new code has: $K' = K - b$, $N' = N - b$
The shortened code is not cyclic

Extending

For Hamming code we can add a general parity check bit with the same K
We obtain a new code $\Rightarrow$ NOT cyclic
$\Rightarrow$ with an even d_{min}, at least same or greater than original

Performance Evaluation

* Soft Decision: (Optimal Decoding)

The optimal receiver implement the soft decision. It works in the signal space using euclidean distance and stimating the likehood function

↳ Minimize the distance $\int_T (r-s_i)^2 dt$

↳ Maximize the correlation

$\int r(t) s_i(t) dt = \langle r, s_i \rangle$

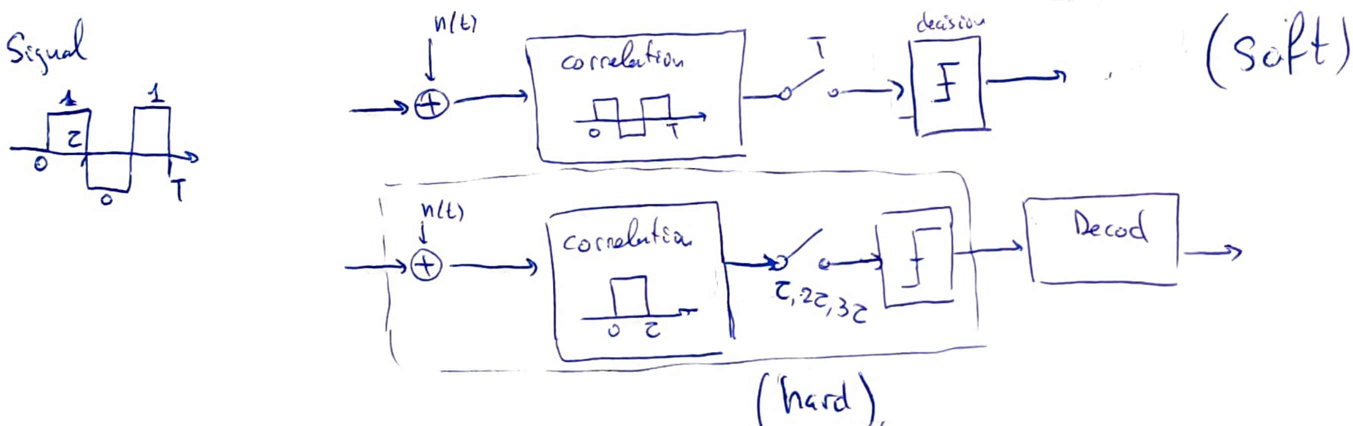
* Hard Decision:

To simplify the receiver architecture. Instead of working in euclidean space we work in Hamming space ("1" and "0").

Using hard decision is an approximation so the hard decision receiver is sub-optimal but much more simple.

We loss only around 2dB of SNR

* Difference in soft/hard procedure



↳ Soft: If we change the code ⇒ we need to change the receiver

↳ Hard: If we chang the code ⇒ change the decodification strategy, usually a software

* Error Probability

The best performance that can be obtained is the soft decision

We introduce the parameter

$$\boxed{\varepsilon = Q\left(\sqrt{\frac{2E_b}{N_0}\frac{K}{N}}\right)}$$

↳ code rate

⇒ In the binary antipolar mod. $P(E) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$

⇒ In case of using codes
- We transmit K information bits
- To transmit the useful information bits we need to use N code bits

⇒ For the soft decision we obtain

$$\boxed{Q\left(\sqrt{\frac{2E_b}{N_0}\frac{K}{N}\cdot d_{min}}\right)}$$

⇒ For hard decision we have

$$P(E) = \sum_{h=t+1}^{N}\binom{N}{h}\varepsilon^h(1-\varepsilon)^{N-h}$$

being t the errors we can correct (that's why the sum starts from $t+1$)

t → error correcting capability

↓ approximation

$$\boxed{P(E) \simeq Q\left(\sqrt{\frac{2E_b}{N_0}\frac{K}{N}(t+1)}\right)}$$

↳ $d_{min} \leq 2t+1$ ≈ we lose ~3dB (divide by 2)

Interleaving

When we have a lot of errors in the transmission that are in group we call them: Burst Errors.

To correct this errors the classical correction strategy is not very efficient.

The idea of interleaving is to break this burst of errors in order to make them manageable for the classic coding strategy.

⇒ Ex

Instead of reading by rows we read by columns so we only have 1 error to solve

⇒ Interleaving introduces a delay