INFORMATION THEORY & CODING

Week 13 : Channel Coding

Dr. Rui Wang

Department of Electrical and Electronic Engineering
Southern Univ. of Science and Technology (SUSTech)

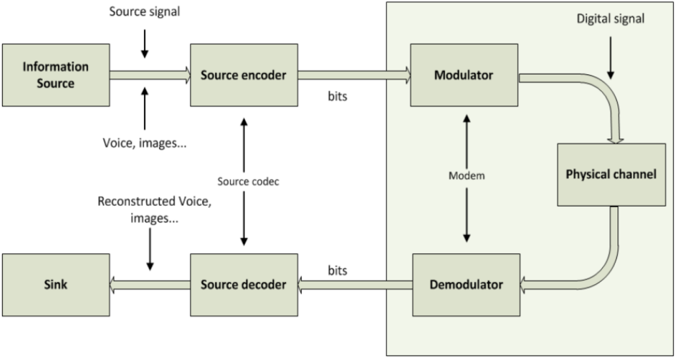
Email: wang.r@sustech.edu.cn

December 7, 2020

Outline

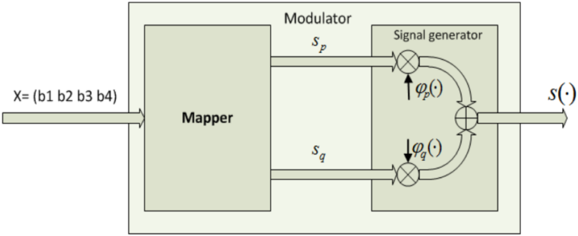
- Communication model: QAM modulation, Constellation
- Block codes: Parity check code, Hamming code, Linear block code
- Continuous codes: Convolutional code

Communication model



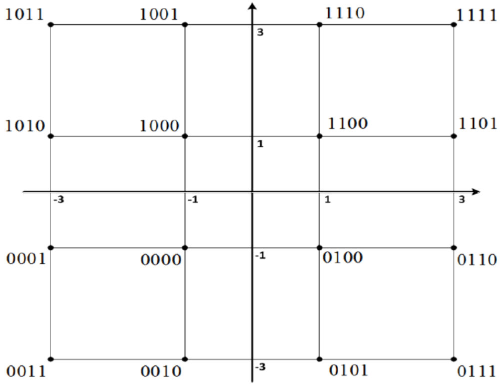
Modem example: QAM

Quadrature Amplitude Modulation (QAM)



The two basis signals $\varphi_p(.)$ and $\varphi_q(.)$ are orthogonal: $\int_0^{T_s} \varphi_p(t)\varphi_q(t)\mathrm{d}t = 0$. For simplicity, let $\varphi_p(.)$ and $\varphi_q(.)$ be normalized: $\int_0^{T_s} \varphi_p^2(t)\mathrm{d}t = 1$ and $\int_0^{T_s} \varphi_q^2(t)\mathrm{d}t = 1$.

Constellation



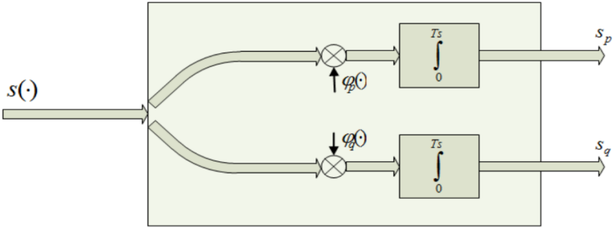
Mapping of 16 QAM modem.

Translate every 4 bits to a constellation point, $b_1b_2b_3b_4 \to (s_p, s_q)$:

- For example, $1011 \to (-3, 3)$, $1111 \to (3, 3)$ and etc.

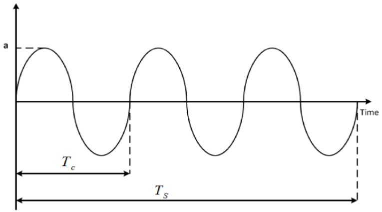
Modem example: QAM

The detector of QAM modem



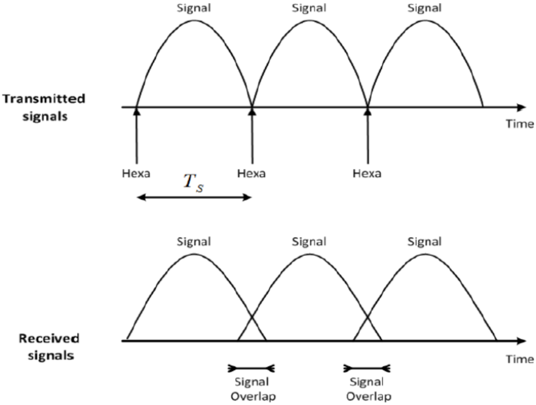
$$\begin{aligned}\int_0^{T_s} s(t)\varphi_p(t)\mathrm{d}t &= \int_0^{T_s} [s_p\varphi_p(t) + s_q\varphi_q(t)]\varphi_p(t)\mathrm{d}t \\ &= \int_0^{T_s} s_p\varphi_p^2(t)\mathrm{d}t + \int_0^{T_s} s_q\varphi_p(t)\varphi_q(t) \\ &= s_p\end{aligned}$$

Orthogonal Carriers



- $\varphi_p(t) = asin\left(\frac{2\pi}{T_c}t\right)$, $\varphi_q(t) = acos\left(\frac{2\pi}{T_c}t\right)$ since $\int_0^{T_s} \sin\left(\frac{2\pi}{T_c}t\right)\cos\left(\frac{2\pi}{T_c}t\right)\mathrm{d}t = \frac{1}{2}\int_0^{T_s}\sin\left(\frac{4\pi}{T_c}\mathrm{d}t\right) = 0$
- To normalize, $a = \sqrt{\frac{2}{T_s}}$.

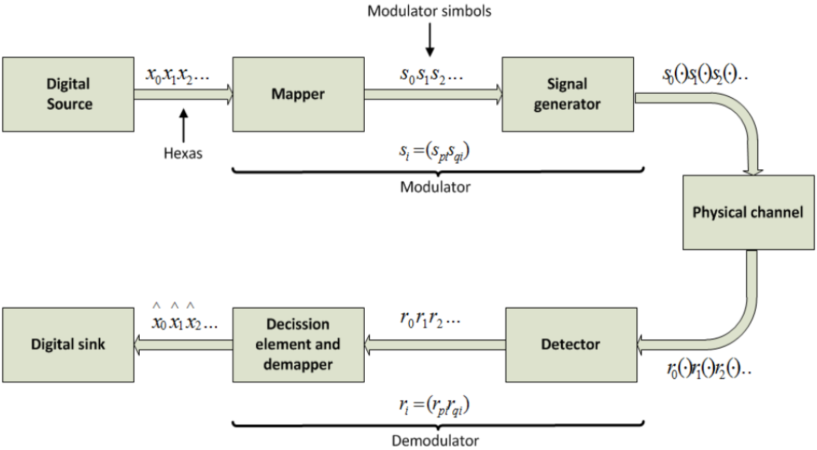
Interference

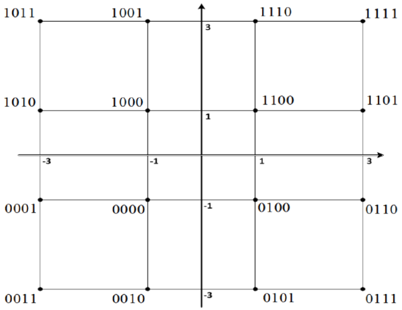


Assumption

- To deal with these interferences is an important topic in Digital Communication course (e.g. adaptive equalization)
- Assume that the interference has been eliminated. The only problem is noise.

Modulation/demodulation process





Mapping of 16 QAM modem.

To transmit the hexa $E = (1110)$, the modem generates with $(1, 3)$ the signal

$$s(t) = \varphi_p(t) + 3\varphi_q(t) = \sqrt{\frac{2}{T_s}} \sin\left(\frac{2\pi}{T_s} t\right) + 3\sqrt{\frac{2}{T_s}} \cos\left(\frac{2\pi}{T_s} t\right)$$

Modulation/demodulation process

To transmit the hexa E, the modem generates with $(1,3)$ the signal

$$s(t) = \varphi_p(t) + 3\varphi_q(t) = \sqrt{\frac{2}{T_s}} \sin\left(\frac{2\pi}{T_s}t\right) + 3\sqrt{\frac{2}{T_s}} \cos\left(\frac{2\pi}{T_s}t\right)$$

Let ϕ be an angle s.t. $\sin\phi = \frac{3}{\sqrt{10}}$, $\cos\phi = \frac{1}{\sqrt{10}}$. Thus,

$$\begin{aligned} s(t) &= \sqrt{10}\sqrt{\frac{2}{T_s}} \sin\left(\frac{2\pi}{T_c}t\right)\cos\phi + \sqrt{10}\sqrt{\frac{2}{T_s}} \cos\left(\frac{2\pi}{T_c}t\right)\sin\phi \\ &= \sqrt{10}\sqrt{\frac{2}{T_s}} \sin\left(\frac{2\pi}{T_c}t + \phi\right) \end{aligned}$$

Modulation/demodulation process

The received signal with noise $n(t)$ can be written as

$$r(t) = s(t) + n(t).$$

The detected symbols

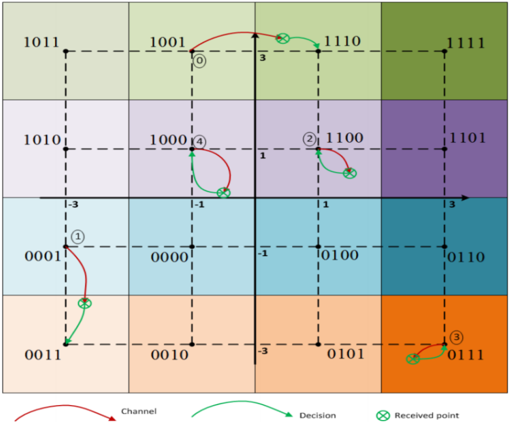
$$r_p = \int_0^{T_s} r(t)\varphi_p(t)dt = 1 + \underbrace{\int_0^{T_s} n(t)\varphi_p(t)dt}_{n_p}.$$

$$r_q = \int_0^{T_s} r(t)\varphi_q(t)dt = 3 + \underbrace{\int_0^{T_s} n(t)\varphi_q(t)dt}_{n_q},$$

where n_p and n_q are both Gaussian R.V.

Modulation/demodulation process

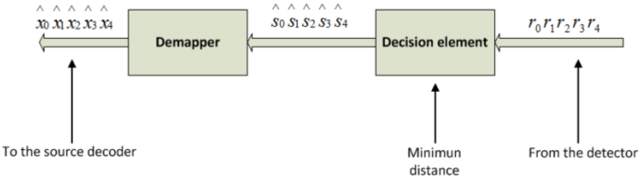
9 1 C 7 8 - 1001 0001 1100 0111 1000 — hexas to be sent
(-1, 3) (-3, -1) (1, 1) (3, -3) (-1, 1) — modem symbols
(0.5, 3.25) (-2.75, -2.13) (1.5, 0:5) (2.5, -3.25) (-0.5, 0.12) (1, 3) (-3,-3)
(1, 1) (3,-3) (-1, 1) — decision symbols



Decision regions for the 16QAM modem

Modulation/demodulation process

9 1 C 7 8 - 1001 0001 1100 0111 1000 — hexas to be sent
(-1, 3) (-3, -1) (1, 1) (3, -3) (-1, 1) — modem symbols
(0.5, 3.25) (-2.75, -2.13) (1.5, 0:5) (2.5, -3.25) (-0.5, 0.12) (1, 3) (-3,-3) (1, 1) (3,-3) (-1, 1) — decision symbols



$\hat{x}_0 = E = (1110)$ $\hat{s}_0 = (1,3)$ $r_0 = (0.5, 3.25)$

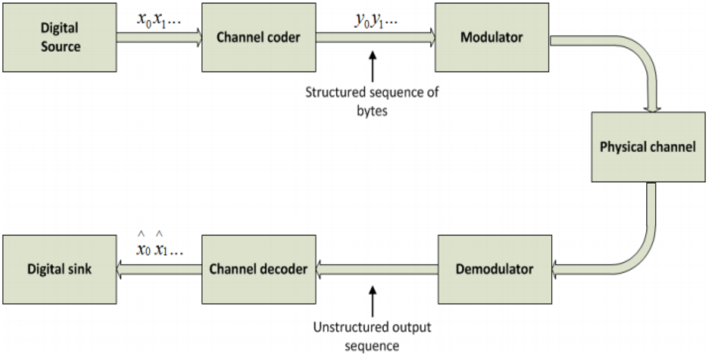
$\hat{x}_1 = 3 = (0011)$ $\hat{s}_1 = (-3,-3)$ $r_1 = (-2.75, -2.13)$

$\hat{x}_2 = C = (1100)$ $\hat{s}_2 = (1,1)$ $r_2 = (1.5, 0.5)$

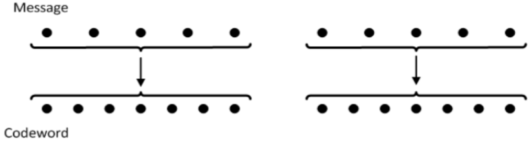
$\hat{x}_3 = 7 = (0111)$ $\hat{s}_3 = (3,-3)$ $r_3 = (2.5, -3.25)$

$\hat{x}_4 = 8 = (1000)$ $\hat{s}_4 = (-1,1)$ $r_4 = (-0.5, 0.12)$

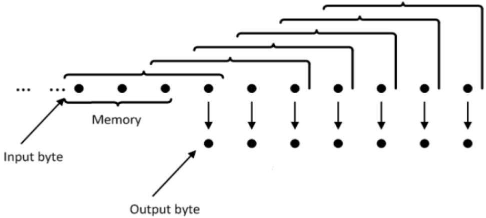
Channel coder & decoder



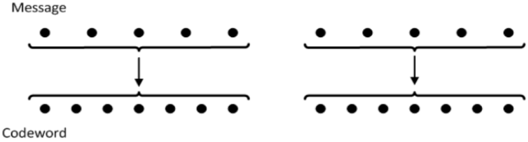
- Block codes
  Map a block of bits onto a codeword



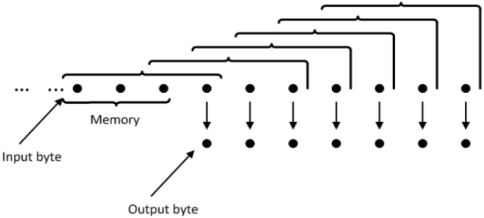
- Continuous (convolutional, trellis) codes
  Each output block depends on some of the past inputs



- Block codes
  Map a block of bits onto a codeword

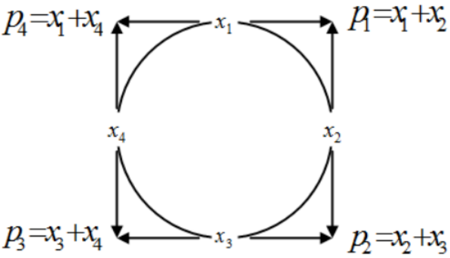


- Continuous (convolutional, trellis) codes
  Each output block depends on some of the past inputs



Block codes

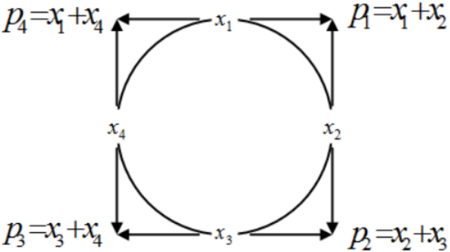
4 bits information with 4 check bits. We are able to correct 1 bit error.



If at most 1 bit is wrong, then

- All parity checks correct — No Error
- One parity check fails — Error in the parity check
- Two parity checks fail — Point to the position of the erroneous bit

4 bits information with 4 check bits. We are able to correct 1 bit error.



If at most 1 bit is wrong, then

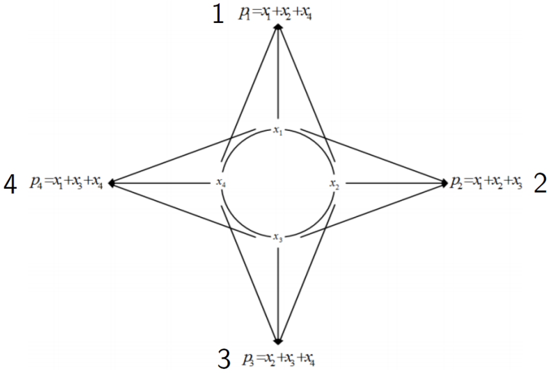
- All parity checks correct — No Error
- One parity check fails — Error in the parity check
- Two parity checks fail — Point to the position of the erroneous bit

Block codes

Two possibilities to do better

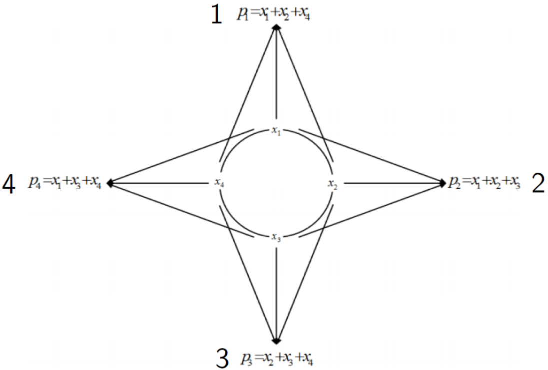
- Find a single error correcting code that requires fewer check bits: a $(7,4)$ code?
- Find a code with the same number of parity bits but enhanced error control capabilities, e.g., a single error correcting code also able to detect two errors?

Block codes



1. x_1 is erroneous, $1, 2, 4$
2. p_1 is erroneous, 1
3. x_1 and p_1 ..., $2, 4$
4. x_1 and p_2 ..., $1, 4$
5. x_1 and p_3 ..., $1, 2, 3, 4$
6. x_1 and x_2 ..., $3, 4$
7. x_1 and x_3 ..., $1, 3$
8. p_1 and p_2 ..., $1, 2$
9. p_1 and p_3 ..., $1, 3$

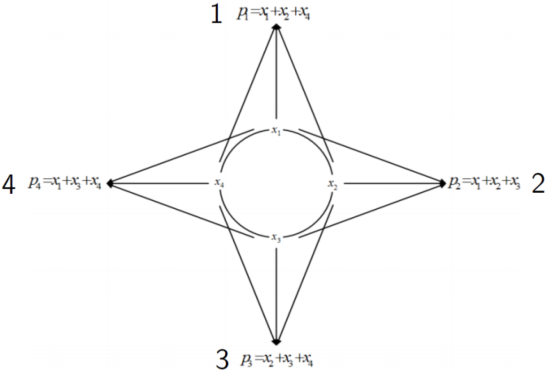
Due to symmetry, only a few cases need to be checked.



1. x_1 is erroneous, $1, 2, 4$
2. p_1 is erroneous, 1
3. x_1 and p_1 ..., $2, 4$
4. x_1 and p_2 ..., $1, 4$
5. x_1 and p_3 ..., $1, 2, 3, 4$
6. x_1 and x_2 ..., $3, 4$
7. x_1 and x_3 ..., $1, 3$
8. p_1 and p_2 ..., $1, 2$
9. p_1 and p_3 ..., $1, 3$

Suppose at most 2 bits are wrong, then

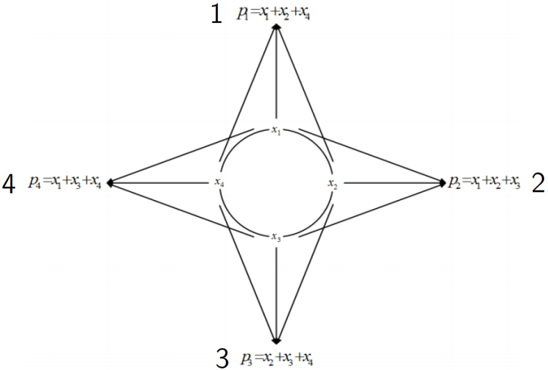
- All parity equations correct — No error
- One equation fails — Error in the parity bit
- An even number of equations fail — Two errors detected
- Three equations fail — One error corrected



If the codeword 10010110 was sent, the receiver got 01001110 with 4 errors. All parity equations are correct, and this is accepted as a legitimate codeword!

undetectable error

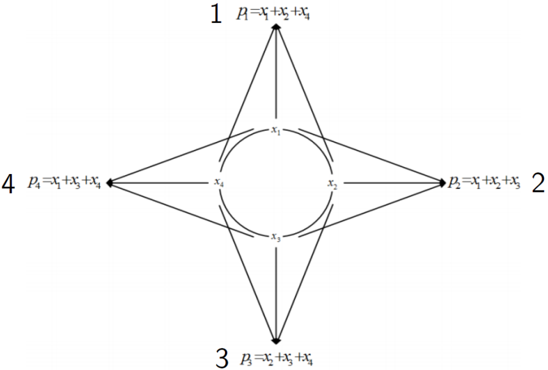
Block codes



If the codeword 10010110 was sent, the receiver got 01001110 with 4 errors. All parity equations are correct, and this is accepted as a legitimate codeword!

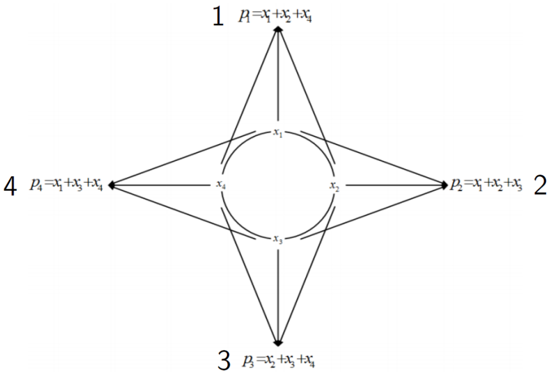
undetectable error

Block codes



If the codeword 10010110 was sent, the receiver got 11001110 with 3 errors. Equations $1, 2, 4$ fail. x_1 was corrected!

erroneous decoding



If the codeword 10010110 was sent, the receiver got 11001110 with 3 errors. Equations $1, 2, 4$ fail. x_1 was corrected!

erroneous decoding

Block codes

— No error
— Undetectable error
— Error detected
— Error corrected
— Erroneous decoding

Block codes

In the previous example, three parity equations are enough! — Hamming code