Prob1

April 22, 2014 2:43 PM

A. 16

vecteurs d'erreur peuvent etre corrigés

	bits de message	001	010	011	100	101	110	111
1	***0000000***	***1101001***	***1011010***	***0110011***	***0111100***	***1010101***	***1100110***	***0001111***
2	***1000000***	0101001	0011010	1110011	1111100	0010101	0100110	1001111
3	***0100000***	1001001	1111010	0010011	0011100	1110101	1000110	0101111
4	***0010000***	1111001	1001010	0100011	0101100	1000101	1110110	0011111
5	***0001000***	1100001	1010010	0111011	0110100	1011101	1101110	0000111
6	***0000100***	1101101	1011110	0110111	0111000	1010001	1100010	0001011
7	***0000010***	1101011	1011000	0110001	0111110	1010111	1100100	0001101
8	***0000001***	1101000	1011011	0110010	0111101	1010100	1100111	0001110
9	***1100000***	0001001	0111010	1010011	1011100	0110101	0000110	1101111
10	***1010000***	0111001	0001010	1100011	1101100	0000101	0110110	1011111
11	***0110000***	1011001	1101010	0000011	0001100	1100101	1010110	0111111
12	***1001000***	0100001	0010010	1111011	1110100	0011101	0101110	1000111
13	***0101000***	1000001	1110010	0011011	0010100	1111101	1001110	0100111
14	***0011000***	1110001	1000010	0101011	0100100	1001101	1111110	0010111
15	***1000100***	0101101	0011110	1110111	1111000	0010001	0100010	1001011
16	***1110000***	0011001	0101010	1000011	1001100	0100101	0010110	1111111

B. 1

Tous les vecteurs avec 1 erreur de bit

seule 7 de $\binom{7}{2} = \frac{7!}{2!5!} = \frac{7 \cdot 6}{2} = 21$ vecteurs avec 2 bits, donc incomplet

bits de message	001	010	011	100	101	110	111
0000000	***1101001***	***1011010***	***0110011***	***0111100***	***1010101***	***1100110***	***0001111***
1000000	0101001	0011010	1110011	1111100	0010101	0100110	1001111
0100000	1001001	1111010	0010011	0011100	1110101	1000110	0101111
0010000	1111001	1001010	0100011	0101100	1000101	1110110	0011111
0001000	1100001	1010010	0111011	0110100	1011101	1101110	0000111
0000100	1101101	1011110	0110111	0111000	1010001	1100010	0001011
0000010	1101011	1011000	0110001	0111110	1010111	1100100	0001101
0000001	1101000	1011011	0110010	0111101	1010100	1100111	0001110
1100000	0001001	0111010	1010011	1011100	0110101	0000110	1101111
1010000	0111001	0001010	1100011	1101100	0000101	0110110	1011111
0110000	1011001	1101010	0000011	0001100	1100101	1010110	0111111
1001000	0100001	0010010	1111011	1110100	0011101	0101110	1000111
0101000	1000001	1110010	0011011	0010100	1111101	1001110	0100111
0011000	1110001	1000010	0101011	0100100	1001101	1111110	0010111
1000100	0101101	0011110	1110111	1111000	0010001	0100010	1001011
1110000	0011001	0101010	1000011	1001100	0100101	0010110	1111111

c) OUI

on peut lire les bits transmis directement

bits de message	001	010	011	100	101	110	111
0000000	***1101001***	***1011010***	***0110011***	***0111100***	***1010101***	***1100110***	***0001111***
1000000	0101001	0011010	1110011	1111100	0010101	0100110	1001111
0100000	1001001	1111010	0010011	0011100	1110101	1000110	0101111
0010000	1111001	1001010	0100011	0101100	1000101	1110110	0011111
0001000	1100001	1010010	0111011	0110100	1011101	1101110	0000111
0000100	1101101	1011110	0110111	0111000	1010001	1100010	0001011
0000010	1101011	1011000	0110001	0111110	1010111	1100100	0001101
0000001	1101000	1011011	0110010	0111101	1010100	1100111	0001110
1100000	0001001	0111010	1010011	1011100	0110101	0000110	1101111
1010000	0111001	0001010	1100011	1101100	0000101	0110110	1011111
0110000	1011001	1101010	0000011	0001100	1100101	1010110	0111111
1001000	0100001	0010010	1111011	1110100	0011101	0101110	1000111
0101000	1000001	1110010	0011011	0010100	1111101	1001110	0100111
0011000	1110001	1000010	0101011	0100100	1001101	1111110	0010111
1000100	0101101	0011110	1110111	1111000	0010001	0100010	1001011
1110000	0011001	0101010	1000011	1001100	0100101	0010110	1111111

D)

	4	4	4	4	4	4	4
0	***1101001***	***1011010***	***0110011***	***0111100***	***1010101***	***1100110***	***0001111***

D)

4	4	4	4	4	4	4	
0	1101001	1011010	0110011	0111100	1010101	1100110	0001111

poids du mot de code ; valeur minimale = 4 = dist minimal

$4 = d_{min}$

E)

$k = 3$ bits

$n = 7$ bits

bits de message	001	010	011	100	101	110	111
0000000	1101001	1011010	0110011	0111100	1010101	1100110	0001111

taux = $3/7$

F) En cherchant le vecteur reçu dans le tableau standard, nous pouvons lire le mot de code correct dans la première rangé de la colonne avec le vecteur reçu. Par exemple, la séquence 0011011 n'est pas un mot de code donc il y avait une erreur de transmission. En cherchant 0011011 dans la table nous trouvons que le mot de code valide était 011011, c.-à-d., que le troisième bit a été inversé.

G)

bits de message	001	010	011	100	101	110	111
0000000	1101001	1011010	0110011	0111100	1010101	1100110	0001111
1000000	0101001	0011010	1110011	1111100	0010101	0100110	1001111
0100000	1001001	1111010	0010011	0011100	1110101	1000110	0101111
0010000	1111001	1001010	0100011	0101100	1000101	1110110	0011111
0001000	1100001	1010010	0111011	0110100	1011101	1101110	0000111
0000100	1101101	1011110	0110111	0111000	1010001	1100010	0001011
0000010	1101011	1011000	0110001	0111110	1010111	1100100	0001101
0000001	1101000	1011011	0110010	0111101	1010100	1100111	0001110
1100000	0001001	0111010	1010011	1011100	0110101	0000110	1101111
1010000	0111001	0001010	1100011	1101100	0000101	0110110	1011111
0110000	1011001	1101010	0000011	0001100	1100101	1010110	0111111
1001000	0100001	0010010	1111011	1110100	0011101	0101110	1000111
0101000	1000001	1110010	0011011	0010100	1111101	1001110	0100111
0011000	1110001	1000010	0101011	0100100	1001101	1111110	0010111
1000100	0101101	0011110	1110111	1111000	0010001	0100010	1001011
1110000	0011001	0101010	1000011	1001100	0100101	0010110	1111111

mots de code n'incluent pas

0011011 donc un erreur

chercher dans la table

le bon message est en tête de la colonne avec le vecteur reçu

message : 011

14)

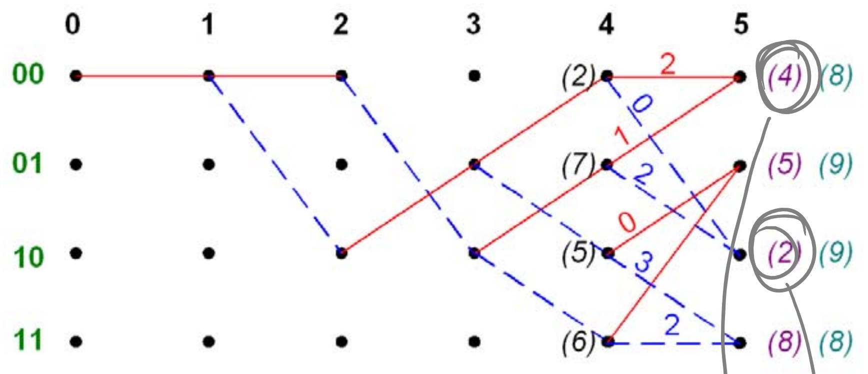
bits de message	001	010	011	100	101	110	111
0000000	***1101001***	***1011010***	***0110011***	***0111100***	***1010101***	***1100110***	***0001111***
1000000	0101001	0011010	1110011	1111100	0010101	0100110	1001111
0100000	1001001	1111010	0010011	0011100	1110101	1000110	0101111
0010000	1111001	1001010	0100011	0101100	1000101	1110110	0011111
0001000	1100001	1010010	0111011	0110100	1011101	1101110	0000111
0000100	1101101	1011110	0110111	0111000	1010001	1100010	0001011
0000010	1101011	1011000	0110001	0111110	1010111	1100100	0001101
0000001	1101000	1011011	0110010	0111101	1010100	1100111	0001110
1100000	0001001	0111010	1010011	1011100	0110101	0000110	1101111
1010000	0111001	0001010	1100011	1101100	0000101	0110110	1011111
0110000	1011001	1101010	0000011	0001100	1100101	1010110	0111111
1001000	0100001	0010010	1111011	1110100	0011101	0101110	1000111
0101000	1000001	1110010	0011011	0010100	1111101	1001110	0100111
0011000	1110001	1000010	0101011	0100100	1001101	1111110	0010111
1000100	0101101	0011110	1110111	1111000	0010001	0100010	1001011
1110000	0011001	0101010	1000011	1001100	0100101	0010110	1111111

Le vecteur
011 0011
est dans la liste
de mots de codes
⇒ pas d'erreur

message : 011

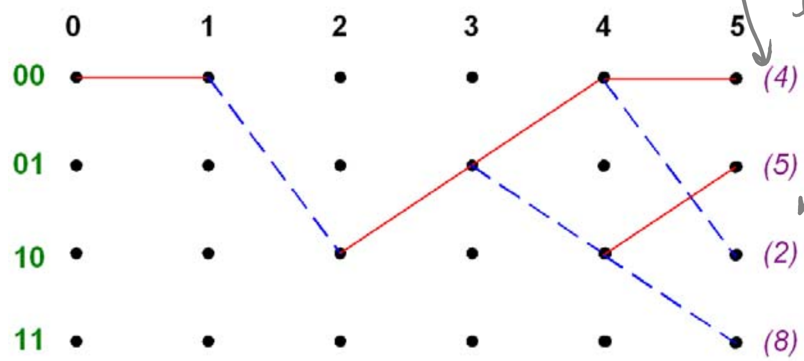
Prob 2

April 22, 2014 3:03 PM

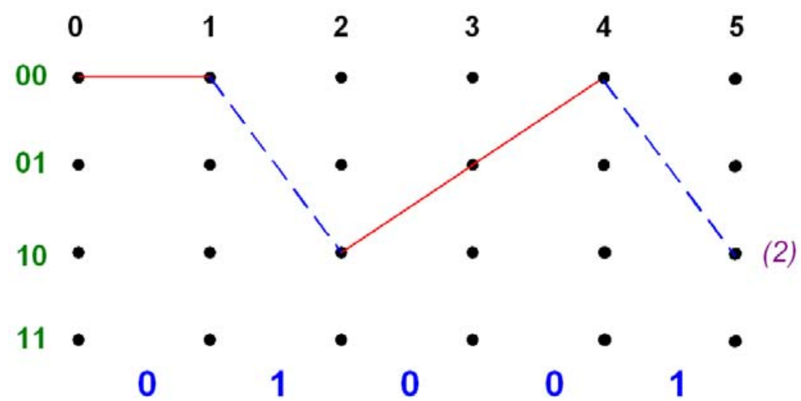


Calculer les deux métrique de chemin (en haut + en bas) pour les 4 états

etc.



Choisir le chemin avec la distance plus bas + éliminer les autres chemins.



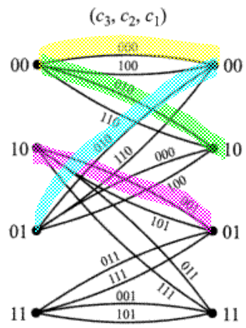
Le décodeur choisera le chemin le plus court ⇒ celui avec métrique 2.

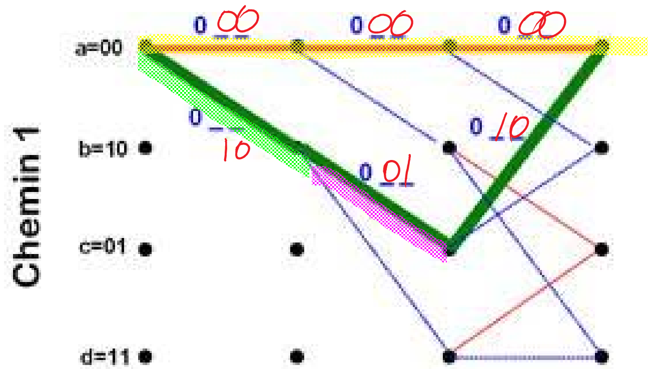
ligne rouge = 0
ligne bleu = 1

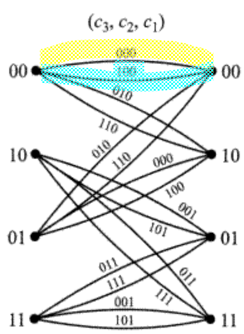
La distance n'est pas nulle, donc il y a eu des erreurs. La distance est 2, donc deux erreurs.

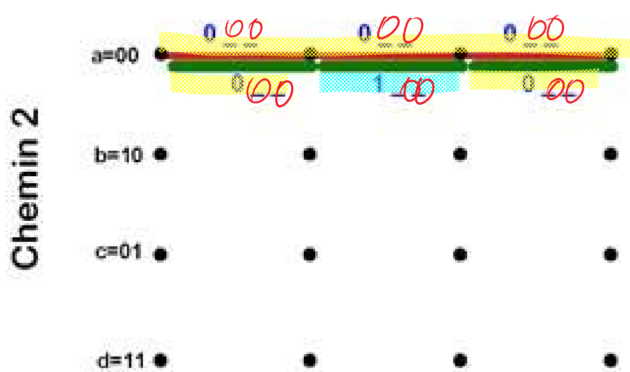
Prob 3

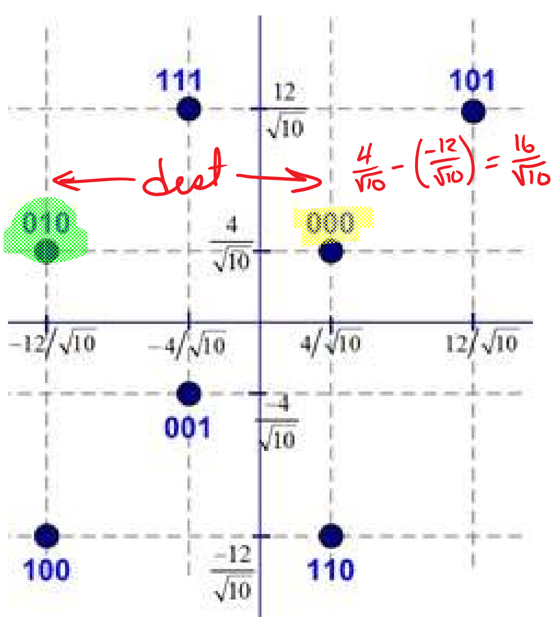
April 22, 2014 3:09 PM

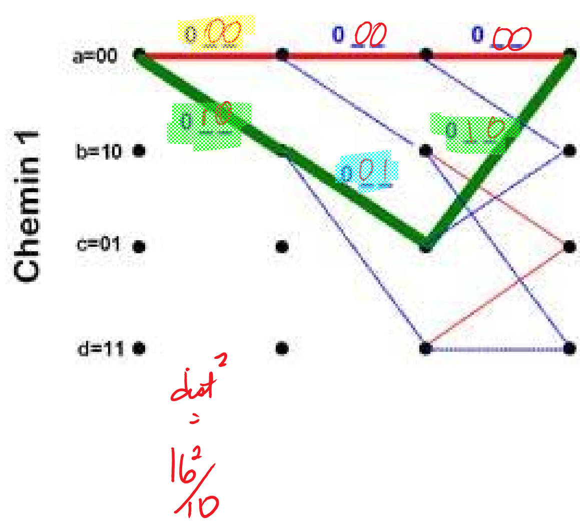


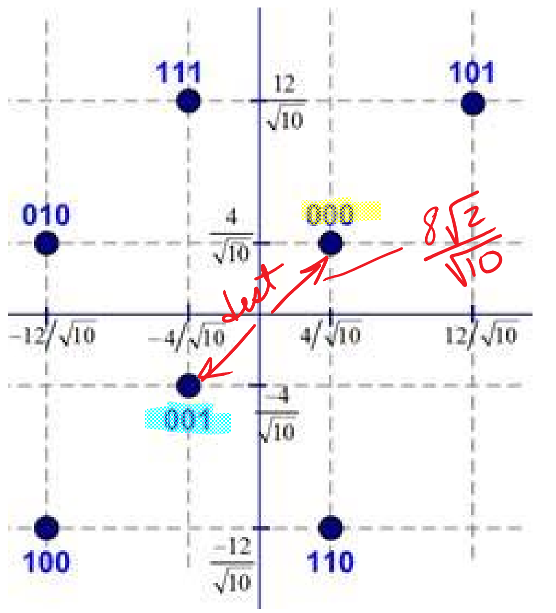


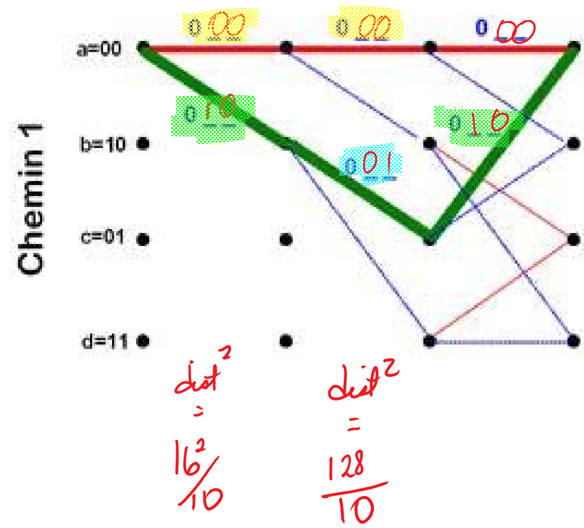


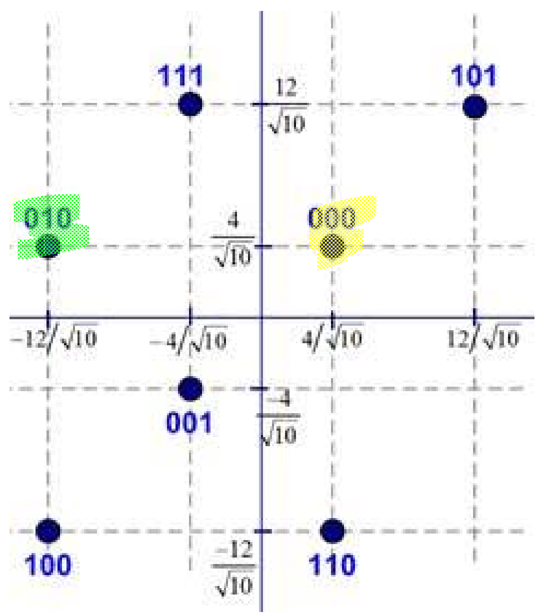


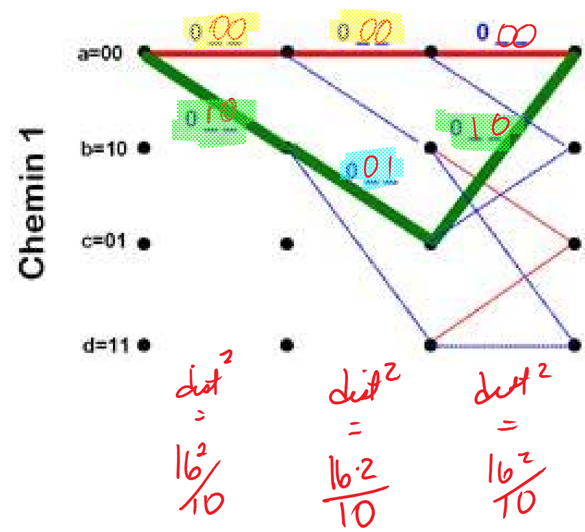






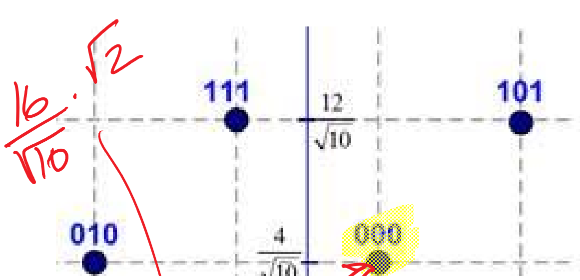


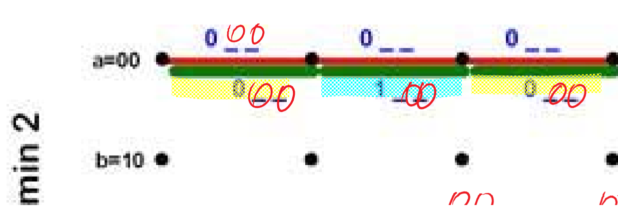


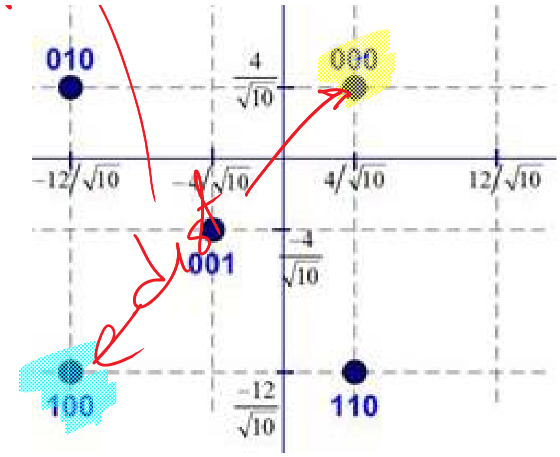


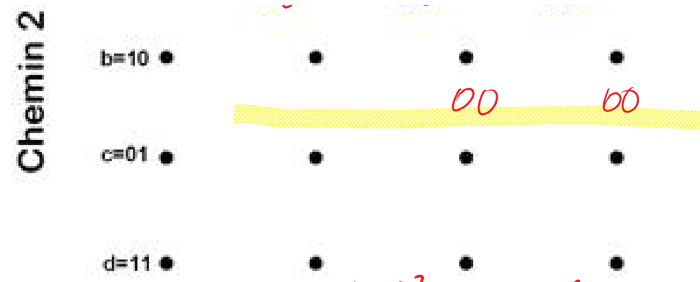
$$dist^2 = \frac{16^2}{10} + \frac{128}{10} + \frac{16^2}{10} = \frac{16^2}{5} + \frac{64}{5}$$

$$= 64 \cdot \left[\frac{4}{5} + \frac{1}{5}\right) = 64 \checkmark$$









$dist^2 = 0$

$dist^2 = \frac{16^2 \cdot 2}{10}$

$dist^2 = 0$

$$dist^2 = 0 + \frac{16^2}{5} + 0 = \frac{16^2}{5} = \frac{256}{5} = 51.2$$

plus petit $\Rightarrow$ dist minimal $= \frac{16^2}{5} = \frac{256}{5}$

Prob 4

April 22, 2014 3:21 PM

Nous limitons l'expansion en largeur de bande à 50% pour le code correcteur d'erreur.

largeur de bande initial = x

" " " après codage = $x + \frac{x}{2} = \frac{3}{2}x$

taux de code = $\frac{x}{\frac{3}{2}x} = 2/3$

Temps de garde

- Codage moyenne (2/3 ou 3/4)
  - » Temps de garde = 4 fois délai

$\Rightarrow T_g = 4 \times$ délai $= 4 \times 10\mu sec = 40\mu sec$

Nous limitons la perte en SNR à 1 dB pour l'exploitation d'un temps de garde.

$$SNR_{perte} = 10\log_{10}\frac{\text{temps util}}{\text{temps totale}} = 10\log_{10}\frac{\text{temps util}}{\text{temps d'un symbole}} = 5/6$$

$6\,temp_u = 5\,tem_u + T_G \cdot 5$

$t_u = 5 \cdot T_G$

$= 200\mu sec$

$T_{sym} = 240\mu sec$

$$4:1 \quad 10\log_{10}\frac{4}{5} = 1\text{dB}$$

$5:1 \quad 10\log_{10} 5/6 = .8\,dB$

$\Rightarrow \frac{\text{temps util}}{\text{temp util} + T_g} = \frac{4}{5} \Rightarrow$ temps util $= 4T_g$

$= 4 \times 40\mu sec = 160\mu sec$

Temp symbol $= 160\mu sec + T_g = 200\mu sec$

$1/\text{temp util}$ = espacement entre porteuse

$= 1/160\mu sec = 6.25kHz \qquad \frac{1}{200\mu s} = 5kHz$

$\#$ porteuses $= \frac{500kHz}{6.25kHz} = 80 \qquad \frac{500}{5} = 100$

$\overline{6.25\,kHz}$ = 80   5 = 100

Nombre de porteuses

- Largeur de bande totale

$$BW = \#\text{ de porteuses} \times \frac{1}{\text{temps util}} = \frac{\#\text{ de porteuses}}{\text{temps d'un symbole-temps de garde}}$$

- Larguer de bande d'un « sous canal »

Taux binaire

$$\text{taux sans codage} = \frac{\#\text{ bits sans codage dans un "symbole" OFDM}}{\text{temps util+temps de garde}}$$

32QAM ⇒ 5 bits

$$\begin{matrix}\#\text{ bits sans codage} \\ \text{dans un "symbole" OFDM}\end{matrix} = \#\text{ de porteuses} \times \frac{1\text{ symbole codé}}{\text{porteuse}} \times \frac{\#\text{ bits codés}}{\#\text{ symboles codés}} \times \frac{\#\text{ bits non-codés}}{\#\text{ bits codés}}$$

modulation   codage

100 $\cancel{80} \times 1 \times 5 \times \frac{2}{3}$

$= \cancel{800}/3$ 1000 $= 266.7$ 333.33

taux binaire $= \frac{266.7\text{ bits}}{200\,\mu\text{sec}} = \frac{1333.33}{1000\,\mu\text{sec}} = \frac{1.33}{\mu\text{sec}} = 1.33\,\text{Mb/s}$

$\frac{333}{240} = 1.388$

B) Bande de garde $= 10\% \cdot 500 = 50\,kHz =$

$\Rightarrow 450\,kHz$ disponible $\Rightarrow \frac{450}{6.25} = \cancel{72}$ 88 porteuses

100 − 10% 100 = 90

(80 − 10% 80 = 72)

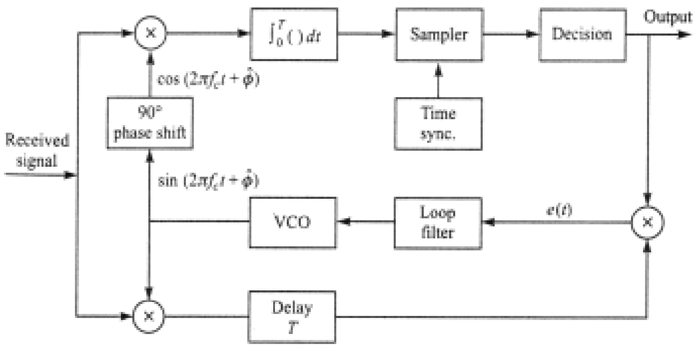
2 pour tonalités ⇒ 70 pour les données

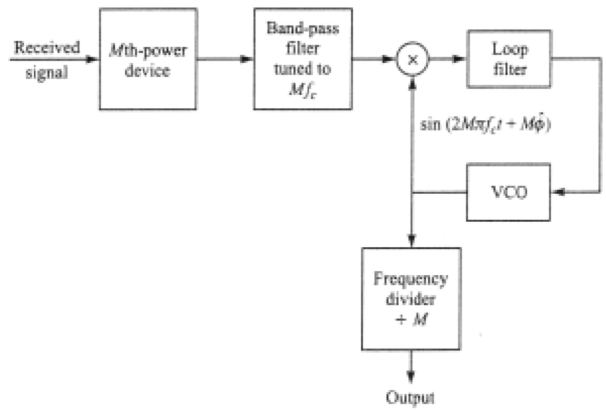
$$\Rightarrow \frac{70 \times 5 \times \frac{2}{3}}{200\,u\,sec} = \frac{14}{4 \cdot 3} = \frac{14}{12} = \frac{7}{6} = 1.167\ Mb/s$$

$$\frac{88 \times 5 \times \frac{2}{3}}{240} = 1.22\ mb/s$$

Prob. 5

April 22, 2014 3:49 PM

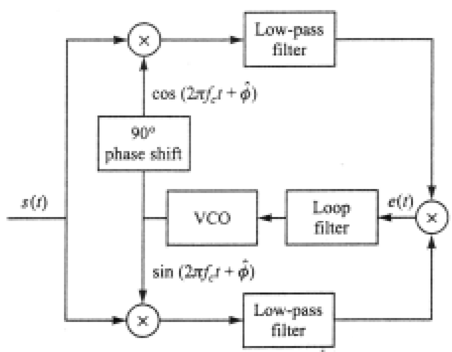


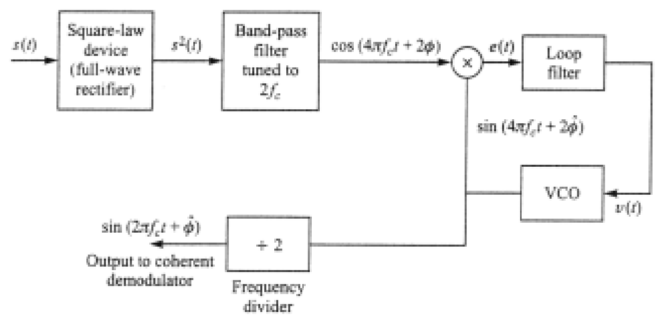


PLL 1. B) re-modulation

Nous voyons que le détecteur sort la décision et fait entrer les données dans le boucle pour enlever la modulation de la référence. La désavantage de cette approche est le délai.

PLL 2. C) mettre signal reçu au carré (puissance quatre, etc.)





PLL 4. B) re-modulation

Dans une boucle de Costas la décision est générée dans la boucle, donc en exploitant la re-modulation des données. Il n'y a pas de délai introduit.

PLL 3. C) mettre signal reçu au carré (puissance quatre, etc.)

B)

$$\frac{d\theta}{dt} = \omega = K_{VCO} \cdot \text{entrée} \qquad \omega \text{ proche de } \omega_0$$

change l'estimé

via

signal de contrôle

change l'estimé de la phase via signal de ...

signal de contrôle

- Entrée de VCO positive

$$\theta-\hat{\theta}>0 \quad \Rightarrow \quad \frac{d\hat{\theta}}{dt}>0 \quad \Rightarrow \quad \hat{\theta}\nearrow$$

$$\theta-\hat{\theta}\to 0$$

- Entrée de VCO négative

$$\theta-\hat{\theta}<0 \quad \Rightarrow \quad \frac{d\hat{\theta}}{dt}<0 \quad \Rightarrow \quad \hat{\theta}\searrow$$

$$\theta-\hat{\theta}\to 0$$