Examen - Traitement numérique du signal - 2008

Les documents et calculatrices sont autorisés. La qualité de la rédaction et de la présentation entreront pour une part importante dans l'appréciation des copies.

Exercice 1. Echantillonage

On échantillonne $e_r(t) = cos(2\pi rt)$ à une fréquence $f = 1/\Delta$.

1. Rappeler la signification de Δ.

   Correction : Δ représente le pas de temps entre deux valeurs d'échantillonnées.

2. Montrer que deux signaux e_r et $e_{r'}$ ont le même échantillonnage si $r\Delta - r'\Delta \in \mathbb{Z}$

   Correction : Supposons $r\Delta - r'\Delta = k$, avec $k \in \mathbb{Z}$. Alors :

$$\begin{aligned} e_{r'}(n\Delta) &= cos(2\pi r'(n\Delta)) \\ &= cos(2\pi n r'\Delta) \\ &= cos(2\pi n(r\Delta + k)) \\ &= cos(2\pi n(r\Delta)) \\ &= e_r(n\Delta). \end{aligned}$$

3. Expliquer pourquoi et dans quel sens le théorème de Nyquist-Shannon et ses hypothèses garantissent l'unicité du r.

   Correction : selon les hypothèses du théorème de Shannon, l'échantillonage est effectué de telle sorte que la plage de fréquence du signal à échantilloner soit inférieur à $\frac{1}{2\Delta}$. On ne peut donc considérer dans ce cadre deux fréquences r et r' telles que $|r - r'| \leqslant \frac{1}{2\Delta}$. Il ne peut donc pas y avoir de k tel que celui considéré dans la question précédente.

Exercice 2. Codage de Huffman

On considère le message

"ceciestuncodagedehuffman"

(on a supprimé les espaces et la ponctuation pour simplifier la construction).

1. Construisez un codage de Huffman du message (Il y a plusieurs codages de Huffman possibles). On indiquera à chaque noeud de l'arbre le poid de son "sous arbre".

   Correction : On commence par dresser le tableau du nombre d'occurences des symboles de la source :

symboles	*occurences*
e	*4*
c	*3*
d	*2*
a	*2*
f	*2*
u	*2*
n	*2*
o	*1*
i	*1*
s	*1*
t	*1*
g	*1*
h	*1*
m	*1*

On obtient par exemple cet arbre :

2. Combien de bits seraient nécessaire pour transmettre le message sans codage ? (Plusieurs réponses possibles, précisez bien les hyopthèses.)

   Correction : deux réponses possibles. Soit on considère qu'il faut coder les lettres de l'alphabet latin, et il faut alors au minimum 5 bits pour coder 26 symboles. Soit on considère qu'on ne code que les lettres considérées. Puisqu'il y en a 14 lettres, il suffit alors de 4 bits.
   Dans un cas on obtient 120 bits, dans l'autre 96.

3. Combien de bits sont nécessaires après codage de Huffman ?

   Correction : Dans le cas de l'arbre proposé ci-dessus, on obtient :

$$3.(4+3+2)+4.(2+2+2+2+1)+5.(1+1+1+1+1+1)=93$$

Exercice 3. Codes correcteurs
On considère le code linéaire de correction d'erreur décrit par le tableau ci-dessous :

mots source	mots code
000	?00 ? ?
001	00101
0 ? ?	010 ? ?
011	? ?1 ?1
100	1001 ?
101	101 ?1
110	1100 ?
111	111 ? ?

1. En utilisant l'hypothèse de linéarité, remplacer les « ? » par les éléments binaires manquant (« 0 » ou « 1 »).

   Correction : Dans la colonne de droite, on trouve facilement que l'élément manquant est $[010]$. *Tout le reste de cette question se résoud en utilisant la linéarité du code. Notons* f *l'application linéaire associée.*
   On a obligatoirement $f([000]) = [00000]$. *De plus,* $f([001]) + f([010]) = f([011])$. *On en déduit* $f([010]) = [010?0]$ *et* $f([011]) = [011?1]$.
   De même $f([100]) + f([101]) = f([001])$ *donne par déduction* $f([100]) = [10010]$ *et* $f([101]) = [10111]$.
   De proche en proche, on obtient le tableau :

mots source	*mots code*
000	*00000*
001	*00101*
010	*01010*
011	*01111*
100	*10010*
101	*10111*
110	*11000*
111	*11101*

2. Expliciter la matrice génératrice du code.

   Correction : On déduit la matrice génératrice d'après les images des trois vecteurs de la base canonique $[001]$, $[010]$, $[100]$. *Ce qui donne- en notant les mots sous forme de vecteurs colonnes :*

$$G = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

   Evidement, selon les conventions adoptées, on peut obtenir un ordre diffèrnt des colonnes, ou bien la transposé de cette matrice, avec une permutation des lignes.

3. Quel est l'image d'une séquence $[abc]$, où a, b, c sont des nombres binaires dans $\mathbb{Z}/2\mathbb{Z}$?

   Correction : on trouve par lecture de la matrice :

$$f([abc]) = [a\ b\ c\ a+b\ c]$$

4. En déduire deux relations simples caractérisant les composantes des mots du code.

   Correction : si on note $[x\ y\ z\ u\ v]$ un mot du code, on a, par exemple

$$\left\{\begin{array}{lcl} x+y+u & = & 0 \\ z+v & = & 0 \end{array}\right.$$

5. On note les vecteurs sous forme de colones. Expliquer pourquoi la matrice de contrôle permettant de générer le syndrome est de taille 2.

   Correction : l'espace de codage étant $(\mathbb{Z}/2\mathbb{Z})^5$, la matrice de contrôle a nécessairement 5 colonnes. Puisque le rang du sous espase des mots du code est 3 (puisqu'ils sont l'image par une injection de $(\mathbb{Z}/2\mathbb{Z})^3$), on en déduit qu'il est caractérisé par deux équations. La matrice aura donc deux lignes.

6. Déduire des trois questions précédentes une matrice de contrôle.

   Correction : on peut par exemple considérer

$$H = \begin{pmatrix} 11010 \\ 00101 \end{pmatrix}$$

7. Combien ce code corrige-t-il d'erreurs ?

   Correction : la distance de se code est 3. On en déduit qu'il corrige une erreur.

Exercice 4. Etude et synthèse de filtres numériques RIF

1. Analyse d'un filtre numérique
   On considère un système numérique caractérisé par :

$$H(z) = K.[1 - z^{-10}]; K > 0$$

   (a) Donner l'algorithme de filtrage et la réponse impulsionnelle de ce système.

   Correction : la formule de récurrence déduite de la définition de H donne :

$$y_n = x_n - x_{n-10}.$$

   La réponse impulsionnelle est donc définie par :

$$\left\{\begin{array}{lcl} h_n & = & 1 \text{ si } n = 0, \\ h_n & = & -1 \text{ si } n = 10 \\ h_n & = & 0 \text{ sinon.} \end{array}\right.$$

   (b) Ce filtre est-il stable ? justifier votre réponse.

   Correction : ce filtre est un RIF. Il est donc stable.

   (c) Est-il à phase linéaire ? justifier votre réponse.

   Correction : les coefficient de la réponse impulsionnelle vérifient $h_{10-n} = -h_n$, le filtre est donc à phase linéaire.

(d) Réponse en fréquence du système :

i. Calculer $|H(e^{i\omega})|$ et $j(\omega) := arg\,(H(e^{i\omega}))$.

Correction : On a

$$1 - e^{-i10\omega} = 2i.e^{-i5\omega}.sin(5\omega).$$

Donc : $arg(K(1 - e^{-i10\omega})) = arg(ie^{-i5\omega}) + arg(sin(5\omega))$, *et par conséquent :*

$$j(\omega) = \frac{\pi}{2} - 5\omega.$$

De même :

$$|H(e^{i\omega})| = sin(5\omega).$$

ii. Déterminer les fréquences f_{max} et f_{min} pour lesquelles on obtient les valeurs respectivement maximales (H_{max}) et minimales (H_{min}) de $|H(e^{i\omega})|$.

Correction : On trouve comme valeurs $\omega = \frac{\pi}{10}, \frac{3\pi}{10}, \frac{5\pi}{10}, \frac{7\pi}{10}, \frac{9\pi}{10}$ *ainsi que les opposés de ces valeurs.*

iii. Déterminer la valeur de K qui normalise H_{max} à l'unité.

Correction : On trouve $H_{max} = 2K$, *donc* $K = 1/2$.

iv. Dessiner l'allure de $|H(e^{i\omega})|$ pour cette valeur de K.

Correction : tracer un sinus...

2. Synthèse d'un filtre RIF passe-bas

On désire réaliser un filtre numérique passe-bas dont la fréquence de coupure est $f_e/3$, où f_e est la fréquence d'échantillonnage.

(a) Donner le graphe de la réponse en fréquence $G(f)$ du filtre idéal à phase nulle répondant a ce cahier des charges.

Correction : on cherche à obtenir un créneau, valant 1 avant $f_e/3$ *et 0 après.*

(b) Déterminer la réponse impulsionnelle g_k du filtre idéal non causal à phase nulle par développement en séries de Fourier de $G(f)$.

Correction : d'après les formules du cours (revues au TP2), on trouve

$$g_k = \frac{sin(\frac{2\pi}{3}k)}{k\pi}.$$

(c) On décide de fixer la longueur du filtre à N termes. Montrer que ce choix permet de rendre le RIF causal et donner l'algorithme de filtrage.

Correction : il suffit tronquer puis de décaler le signal. On obtient :

$$\tilde{g}_k = \frac{sin(\frac{2\pi}{3}(k - \frac{N-1}{2}))}{\pi(k - \frac{N-1}{2})}.$$

L'algorithme correspondant s'écrit :

$$y_n = \sum_{k=0}^{N} g_{n-k} x_k.$$