Devoir de filtrage optimal

Élèves de l'option DSMT

Année scolaire 2021-2022

GUY PLANTIER
ESEO

3 décembre 2021

Table des matières

Chapitre 1

Introduction

Vous trouverez ci-après un devoir de filtrage optimal qui servira d'évaluation des connaissances pour le cours de filtrage optimal de l'option DSMT de l'ESEO. Les points suivants seront à respecter :

- Vous devez vous regrouper en binômes
- Le devoir est à rendre pour le **vendredi 28 janvier 2022 à 17h00**. Passé ce délai, je n'accepterai plus vos devoirs, qui ne seront donc pas notés.
- **Je pourrai éventuellement organiser quelques oraux avec quelques étudiants tirés au hasard.**
- Vous devez me fournir
  - un rapport manuscrit proprement présenté avec des figures soigneusement commentées.
  - un ou des programmes rangés dans un seul et même fichier compressé au format ZIP. Ce fichier doit porter un nom du type NOMa_NOMb.zip où NOMa et NOMb désignent les noms des élèves composant le binôme.
- Vous pouvez éventuellement dactylographier votre document, mais ce n'est absolument pas une obligation. Ne perdez pas trop de temps à cela.
- Le fichier NOMa_NOMb.zip devra m'être envoyé par mail.

Pour ce devoir, vous devrez

- Travailler les différents supports de cours que je vous ai distribués et en particulier sur :
  - le filtrage de KALMAN,
  - les corrélations temporelles,
  - la transformation en z (que je vous redonne)
- Faire des recherches par vous-même sur le filtrage de WIENER et les équations de WIENER-HOPF que je vous ai déjà présentées en cours et que nous avons déjà utilisées pour identifier un système (revoir le devoir du printemps 2021 que nous avons corrigé à l'automne 2021). Je pense que cet article WIKIPEDIA est presque suffisant : `https://en.wikipedia.org/wiki/Wiener_filter`
- Faire une recherche sur l'algorithme adaptatif LMS qui est extrêmement simple et très utilisé pour résoudre les équations de WIENER-HOPF de façon adaptative (récursive). Voir en particulier :
  - `https://en.wikipedia.org/wiki/Least_mean_squares_filter`
  - `https://en.wikipedia.org/wiki/Similarities_between_Wiener_and_LMS`
- Utiliser la fonction `lms_c.mexw64` que j'ai écrite en C et qui peut être appelée directement de MATLAB et que je vous donne également.

REMARQUES et CONSEILS

La DLL exécutable depuis MATLAB est `lms_c.mexw64` et est jointe au zip que je vous donne. Vous devez la mettre dans votre répertoire de travail MATLAB . Pour vérifier que cela fonctionne, tapez simplement la commande `lms_c` sous MATLAB et sans argument. Vous aurez un message d'erreur vous disant comment utiliser correctement cette fonction. L'utilisation est décrite également dans le texte que je vous envoie.

Vous avez une quantité très importante de cours sur le lms sur le web

Voir par exemple : `https://en.wikipedia.org/wiki/Least_mean_squares_filter`

Voir aussi

— SIMON HAYKIN : Adaptive Filter Theory, Prentice Hall, 2002, ISBN 0-13-048434-2

— BERNARD WIDROW, SAMUEL D. STEARNS : Adaptive Signal Processing, Prentice Hall, 1985, ISBN 0-13-004029-0

Rappel sur l'équation de WIENER-HOPF et lien avec le devoir maison du printemps 2021

Dans le devoir maison du printemps 2021 que nous avons corrigé à l'automne 2021 et que je vous redonne dans le zip, nous avons établi puis résolu l'équation de WIENER-HOPF que vous avez à la formule 18 de la page 14.

Cette formule met en évidence la matrice d'autocorrélation du signal $x(k)$ que nous avons construite avec la fonction `toeplitz()` de MATLAB après avoir calculé l'autocorrélation du signal $x(k)$ avec la fonction `xcorr()` de MATLAB . Nous avons vu que la solution de cette équation 18 de la page 14 de ce devoir se fait avec l'opérateur \ de MATLAB .

Pour le présent devoir on démontre que l'équation de WIENER-HOPF associée au problème d'égalisation de canal de transmission présenté sur la FIGURE 2.2 de la page 5 de ce présent devoir est donnée simplement par :

Formulation matricielle de l'équation de WIENER-HOPF

$$\underbrace{\begin{pmatrix} \varphi_{xd}(0) \\ \varphi_{xd}(1) \\ \varphi_{xd}(2) \\ \vdots \\ \varphi_{xd}(N-1) \end{pmatrix}}_{\boldsymbol{\varphi_{\mathbf{xd}}}} = \underbrace{\begin{pmatrix} \varphi_{xx}(0) & \varphi_{xx}(1) & \varphi_{xx}(2) & \dots & \varphi_{xx}(N-1) \\ \varphi_{xx}(1) & \varphi_{xx}(0) & \varphi_{xx}(1) & \dots & \varphi_{xx}(N-2) \\ \varphi_{xx}(2) & \varphi_{xx}(1) & \varphi_{xx}(0) & \dots & \varphi_{xx}(N-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi_{xx}(N-1) & \varphi_{xx}(N-2) & \varphi_{xx}(N-3) & \dots & \varphi_{xx}(0) \end{pmatrix}}_{\boldsymbol{\varphi_{\mathbf{xx}}}} \underbrace{\begin{pmatrix} h(0) \\ h(1) \\ h(2) \\ \vdots \\ h(N-1) \end{pmatrix}}_{\mathbf{h}}$$

$$\varphi_{\mathbf{xd}} = \varphi_{\mathbf{xx}} \cdot \mathbf{h}$$

Dans cette équation,

— $\varphi_{\mathbf{xd}}$ est le vecteur d'intercorrélation construit à partir de l'intercorrélation $\varphi_{xd}(k)$ entre les signaux $x(k)$ et $d(k)$,

— $\varphi_{\mathbf{xx}}$ est appelée matrice d'autocorrélation. C'est une matrice de TOEPLITZ car elle présente une symétrie hermitienne : $\varphi_x = \varphi_x^T$ où l'opérateur T désigne l'opérateur de transposition et conjugaison.

— $\mathbf{h}$ est le vecteur colonne des N coefficients de la réponse impulsionnelle du filtre FIR égaliseur.

Chapitre 2

Notions d'égalisation de canal de transmission

2.1 Introduction

Le dispositif que nous nous proposons d'étudier dans cet atelier est présenté sur la FIGURE 2.1. Celle-ci fait apparaître un canal de transmission défini par une fonction de transfert $H_c(q)$ de la forme

$$H_c(q) = \frac{1 + b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} . \tag{2.1}$$

Ce canal déforme le signal $e(k)$ que l'on souhaite émettre et délivre en sortie le signal $x(k)$ que le récepteur doit traiter pour en extraire l'information pertinente. Cependant, si les transmissions se font dans des conditions trop sévères, les déformations apportées par le canal au signal $e(k)$ ne permettent pas un traitement direct du signal $x(k)$. On est alors contraint de placer en entrée du récepteur un filtre destiné à corriger les distorsions apportées par la canal. Ce filtre, appelé *égaliseur*, a une fonction de transfert $H_e(q)$. Si l'égaliseur corrige parfaitement l'effet du canal de transmission, nous pouvons écrire de façon *naïve* et *simplifiée* :

$$H(q) = H_c(q)H_e(q) = 1 \Longleftrightarrow H_e(q) = \frac{1}{H_c(q)} = \frac{1 + a_1 q^{-1} + a_2 q^{-2}}{1 + b_1 q^{-1} + b_2 q^{-2}} . \tag{2.2}$$

On dit aussi qu'un égaliseur est le filtre inverse du canal de transmission ou encore un filtre de déconvolution.

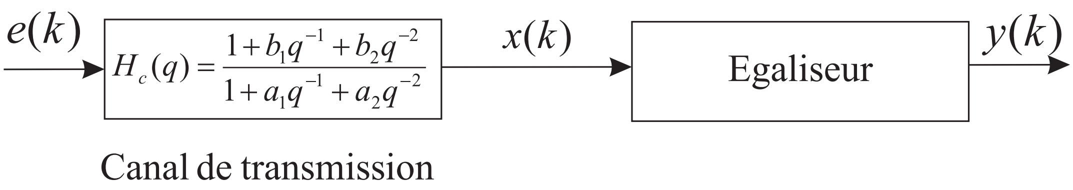

FIGURE 2.1 – **Schéma de principe du canal de transmission étudié et de l'égaliseur utilisé par le récepteur qui permet de reconstruire approximativement le signal d'entrée $e(k)$ à partir du signal sortant du canal de transmission $x(k)$**

Les objectifs de cet atelier sont

1. de montrer que l'équation (2.2) qui donne la fonction de transfert $H_e(q)$ doit être manipulée avec beaucoup de précautions pour déterminer la réponse impulsionnelle du filtre égaliseur,
2. de présenter la notion d'égalisation linéaire d'un canal de transmission,

3. de présenter la méthode d'identification des moindres carrés qui est très employée en pratique,
4. de mettre en œuvre les méthodes de filtrage optimal de WIENER,
5. de programmer et tester deux algorithmes adaptatifs d'égalisation de canal :
   — **Least Mean Square (LMS)**,
   — **Recursive Least Square (RLS) par filtrage de KALMAN**.
   Un schéma général d'égalisation adaptative en temps réel est présenté sur la FIGURE 2.2

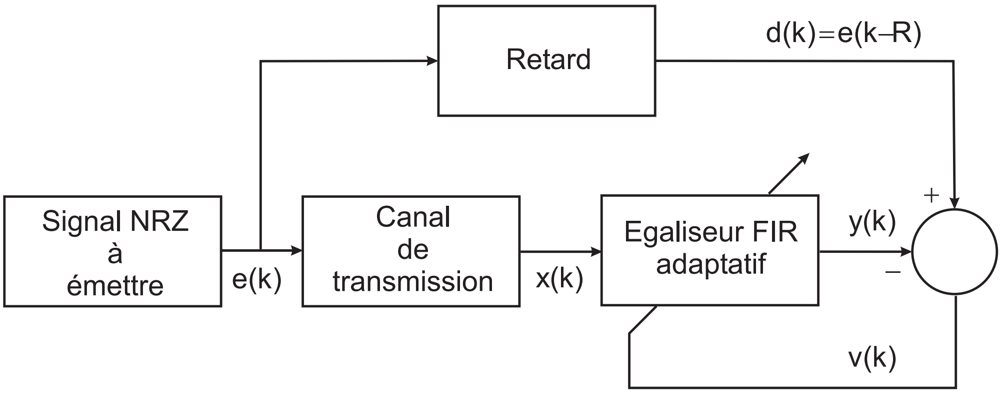

FIGURE 2.2 – **Schéma de principe d'un égaliseur de canal de transmission adaptatif. Le filtre FIR à coefficients ajustables est réglé automatiquement par un des algorithmes LMS ou RLS pour délivrer en sortie une version restaurée (et retardée) du signal émis.**

Le travail que nous proposons ci-après consiste dans un premier temps à faire un étude théorique de la problématique de l'égalisation. Ce travail se fera essentiellement en traitant trois cas particuliers de canaux de transmissions définis par les paramètres présentés dans le tableau ci dessous

	b_1	b_2	a_1	a_2
Cas 1	-1.7119	0.8100	-1.1168	0.9025
Cas 2	-2.0923	1.2100	-1.6042	0.9025
Cas 3	-2.1000	1.0800	-1.7119	0.8100

2.2 Analyse théorique de la problématique de l'égalisation

2.2.1 Étude expérimentale de la transmission d'un code M-aire NRZ sur un canal à bande passante limitée

Nous vous proposons d'effectuer le travail suivant :

1. Écrire un programme MATLAB permettant de présenter la Densité Spectrale de Puissance d'un signal à temps dicret NRZ à 4 états et à valeur moyenne nulle. Pour ce code vous choisirez une fréquence d'échantillonnage Fe et une fréquence symbole F_s telles que :
$$\frac{F_e}{F_s} = 10 \, .$$
2. Représentez la réponse en fréquence du filtre de canal $H_c(q)$ pour les trois cas présentés en introduction. Portez une conclusion sur l'effet du canal sur le signal NRZ à transmettre.
3. Générer et filtrer avec $H_c(q)$ le formant élémentaire servant à définir le code NRZ. Montrer que ces canaux de transmission déforment l'impulsion élémentaire et entraînent de l'Interférence Entre Symboles (IES).

4. Générer une séquence assez longue du signal NRZ $e(k)$. En vous aidant de la fonction `eyediagram()` de MATLAB présentez le diagramme de l'œil associé à ce code.
5. Filtrez $e(k)$ avec les trois canaux de transmission proposés. Montrer à l'aide de la fonction `eyediagram()` que les IES apportées par le canal sont trop importantes pour qu'une décision correcte soit prise. Cette remarque justifie l'utilisation d'un filtre égaliseur.

2.2.2 Détermination théorique de la réponse impulsionnelle d'un filtre égaliseur

Remarquons tout d'abord que la fonction de transfert en z du canal est donnée par :

$$H_c(z) = \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{(1 - z_1 z^{-1})(1 - z_2 z^{-1})}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{2.3}$$

où z_1 et z_2 désignent les zéros de la fonction de transfert $H_c(z)$. La fonction de transfert du filtre inverse est alors donnée par :

$$\begin{aligned} H_e(z) &= \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \\ &= \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{(1 - z_1 z^{-1})(1 - z_2 z^{-1})} \\ &= \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{z_1 - z_2} \left[\frac{z_1}{1 - z_1 z^{-1}} - \frac{z_2}{1 - z_2 z^{-1}} \right] \end{aligned} \tag{2.4}$$

Pour les trois cas de filtres proposés

1. Représenter la carte des pôles et des zéros des filtres $H_c(z)$ et $H_e(z)$. Utiliser pour cela la fonction `zplane()` de MATLAB.
2. Calculer les réponses impulsionnelles des filtres égaliseurs. Montrer que certains de ces filtres ne sont pas causaux. En déduire une condition générale pour que le filtre inverse soit causal et stable.
3. Proposer une solution pour déterminer approximativement la réponse impulsionnelle d'un filtre inverse causal et stable.
4. Écrire un programme MATLAB permettant de présenter les réponses impulsionnelles des filtres égaliseurs.
5. Si on limite la durée de la réponse impulsionnelle $h_e(k)$ à N coefficients, et si on retarde celle-ci de R échantillons, on approxime le filtre égaliseur par un filtre FIR causal de réponse impulsionnelle $h_{e,c}(k)$. En jouant sur les valeurs de N et R présenter l'effet de l'égalisation en filtrant le signal de sortie du canal $x(k)$ à l'aide de $h_{e,c}(k)$. Comparer pour cela le signal de sortie de l'égaliseur $y(k)$ avec le signal désiré $d(k) = e(k - R)$. Utiliser les fonctions `plot()` et `eyediagram()`.
6. Il est aussi intéressant d'exciter le canal de transmission à l'aide d'un signal $e(k)$ quelconque comme un bruit blanc par exemple. Comparer dans ce cas $y(k)$ et $d(k) = e(k - R)$.

2.2.3 Égalisation par filtrage de WIENER causal

1. Reprendre les résultats sur le filtrage de WIENER présentés en cours. Présenter un schéma expliquant comment le filtre optimal de WIENER peut être utilisé pour réaliser un égaliseur.
2. Écrire les équations de WIENER-HOPF pour ce problème. Explicitez chacun des termes de cette équation.

3. En utilisant pour $e(k)$ un bruit blanc, calculer à l'aide de la fonction `xcorr()` une estimation de l'autocorrélation $\varphi_{xx}(k)$ et de l'intercorrélation $\varphi_{xd}(k)$.
4. Utiliser ces derniers résultats pour calculer la solution de l'équation de WIENER-HOPF. Les coefficients du filtre optimal obtenu $\widehat{h}_{e,c}(k)$ ne sont qu'une approximation de la réponse impulsionnelle théorique $h_{e,c}(k)$ puisque les corrélations utilisées ne sont que des estimations des grandeurs théoriques $\varphi_{xx}(k)$ et $\varphi_{xd}(k)$. Comparer $\widehat{h}_{e,c}(k)$ et $h_{e,c}(k)$ en faisant varier N, R et le nombre de points K utilisés pour estimer les corrélations.
5. Évaluer les performances de l'égaliseur optimal de WIENER en visualisant les diagrammes de l'œil de $y(k)$ pour les différents canaux de transmission proposés.

2.3 Égalisation adaptative

2.3.1 Égalisation linéaire adaptative à l'aide de l'algorithme LMS

Nous avons montré en cours que des méthodes itératives de minimisation peuvent être utilisées pour résoudre les équations de WIENER-HOPF. L'algorithme LMS est un des plus utilisé pour effectuer ce travail. Le filtre de WIENER associé à l'algorithme LMS devient alors un *filtre adaptatif*.

Pour permettre de tester cet algorithme sous MATLAB, nous avons écrit une fonction en langage C et créé une DLL associée. Cette fonction, appelée `lms_c()`, est décrite brièvement ci-après

```
Fichier :           lms_c.c

Sujet :             Filtrage adaptatif LMS avec filtre FIR

Auteur :            Guy Plantier

Date :              12-04-94
                    22/06/99 (Modifications 1)

Utilisation :       [sortie,erreur,coefs] = lms_c(entree,desire,coefs_init,mu,nb_coefs_enregistres);

Arguments :         entree                  --> Vecteur d'entrée du signal à filtrer
                    desire                  --> Signal désiré
                    coefs_init              --> Coefficients initiaux du filtre FIR
                    sortie                  --> Vecteur résultat du filtrage adaptatif
                    erreur                  --> Vecteur d'erreur du filtrage e(k) = d(k) -y(k)
                    coefs                   --> Coefficients du filtre FIR après filtrage
                    mu                      --> Coefficient d'adaptation du LMS
                    nb_coefs_enregistres    --> Nombre de vecteurs de coefficients à enregistrer

Modifications :     1)  La fonction enregistre les valeurs des nb_coefs_enregistres
                        dernières évaluations du vecteur des coeficients calculées
```

Cette fonction est obtenue en écrivant un programme en C avec un format particulier permettant d'interfacer l'exécutable `lms_c.dll` avec MATLAB. Nous fournissons les DLL compatibles avec MATLAB 5.2 et MATLAB 6.1. Cependant, les versions pour MATLAB 5.2 peuvent ne pas fonctionner sur tous les PC à cause d'un problème logiciel de MATLAB 5.2. Pour cette raison nous vous proposons le travail suivant :

1. Utiliser les DLL (quand elles fonctionnent !) pour tester vos programmes, mais vous devrez écrire une fonction MATLAB appelée `lms.m` qui reprend la même syntaxe que `lms_c.dll`.
2. Utiliser la fonction `lms.m` pour calculer la réponse impulsionnelle du filtre égaliseur.
3. Utiliser différents signaux $e(k)$ (Code M-aire, bruit blanc, etc.)
4. Comparer les coefficients obtenus par l'algorithme LMS avec les résultats théoriques et les solutions directes des équations de WIENER-HOPF.

2.3.2 Égalisation linéaire adaptative à l'aide de l'algorithme RLS ou un filtre de KALMAN

Une autre stratégie très utilisée pour calculer en ligne, en temps réel et de façon adaptative les coefficients du filtre optimal de WIENER consiste à utiliser un filtre de KALMAN. Cet algorithme est souvent appelé Recursive Least Square (RLS).

Les états du problème sont les coefficients du filtre adaptatif FIR que l'on modélise comme des marches aléatoires. Nous avons alors une pseudo équation de convolution reliant l'entrée et la sortie d'un filtre variant dans le temps

$$y(k) = \sum_{l=0}^{K-1} h_l(k)x(k-l) = h_0(k)x(k) + h_1(k)x(k-1) + \cdots + h_{K-1}(k)x(k-K+1) \tag{2.5}$$

où les $h_l(k)$ représentent la l^evaleur de la réponse impulsionnelle du filtre estimée à l'instant k. Nous avons alors les K équations d'état scalaires suivantes

$$h_l(k+1) = h_l(k) + w_l(k) \quad \text{avec } l = 0, 1, \cdots, K-1 \tag{2.6}$$

où les signaux $w_l(k)$ représentent des bruits d'état fictifs permettant de régler la vitesse de convergence du filtre de KALMAN. En regroupant ces K états dans un vecteur $H(k)$ défini par

$$H(k) = [h_0(k), h_1(k), \cdots, h_{K-1}(k)]^T \tag{2.7}$$

on peut écrire l'équation d'état matricielle suivante :

$$H(k+1) = IH(k) + W(k) \tag{2.8}$$

où $W(k) = [w_0(k), w_1(k), \cdots, w_{K-1}(k)]^T$ désigne le vecteur des perturbations d'état. Ce vecteur $W(k)$ possède une matrice d'autocovariance notée Q.

L'équation d'observation associée à ce problème d'égalisation s'obtient en écrivant que le signal en sortie du filtre égaliseur est égal au signal désiré $d(k) = e(k-R)$ à une erreur près que l'on note $v(k)$

$$\begin{aligned} d(k) &= e(k-R) \\ &= y(k) + v(k) \\ &= h_0(k)x(k) + h_1(k)x(k-1) + \cdots + h_{K-1}(k)x(k-K+1) + v(k) \\ &= \underbrace{[x(k), x(k-1), \cdots, x(k-K+1)]}_{C(k)} H(k) + v(k) \end{aligned} \tag{2.9}$$

La variance de $v(k)$ que l'on ne connaît pas à priori et que l'on notera $\tilde{\sigma}_v^2$ est indispensable pour l'algorithme de KALMAN. On la considère comme un deuxième degré de liberté permettant de régler la vitesse de convergence du filtre.

En vous aidant de votre cours,

1. Mettez en œuvre l'algorithme RLS pour ce problème d'égalisation.
2. En jouant sur les variances des bruits $W(k)$ et $v(k)$, réglez la vitesse de convergence de l'algorithme.
3. Évaluez la qualité de l'égalisation à l'aide d'un diagramme de l'œil.

2.3.3 Comparaison des algorithmes LMS et RLS

Une méthode pratique pour juger de la qualité de l'égalisation consiste à représenter l'erreur d'estimation réelle $v(k) = d(k) - y(k)$. Il est même plus intéressant d'effectuer un grand nombre d'essais (de tirages) de l'ordre de la centaine environ et de représenter la moyenne sur tous ces tirages de la puissance instantanée de $v(k)$ (on fait une moyenne sur des réalisations mais pas une moyenne temporelle). On obtient alors une estimation de la variance de l'erreur d'égalisation à l'instant k notée $\sigma_v^2(k)$. Cette méthode d'évaluation des qualités d'un estimateur porte le nom de méthode de MONTE-CARLO.

1. En reprenant la théorie du filtrage de WIENER, estimez numériquement la puissance J_{min} de l'erreur d'estimation et représentez ces résultats en fonction des paramètres N et R.
2. Représentez sur une figure deux ou trois évolutions $\sigma_v^2(k)$ dans le cas de l'algorithme LMS pour différentes valeur du coefficient d'adaptation μ.
3. Représentez sur une figure deux ou trois évolutions $\sigma_v^2(k)$ dans le cas de l'algorithme RLS pour différentes valeur des variances des bruits $W(k)$ et $v(k)$.
4. En représentant sur une même figure l'évolution temporelle de $\sigma_v^2(k)$ pour les algorithmes LMS et RLS, listez les qualités et les défauts de ces deux méthodes.