

Christian JUTTEN

**FILTRAGE
LINEAIRE
OPTIMAL**

Université Grenoble Alpes - Polytech' Grenoble

Département *Informatique et Electronique des Systèmes Embarqués*

IESE5, option *Images et Signaux et Automatique*

Juillet 2018

Table des matières

1	Introduction	5
1.1	Problèmes de filtrage	5
1.2	Plan du cours	5
1.3	Références	6
2	Processus stochastiques discrets	7
2.1	Position du problème	7
2.2	Matrices d'auto-corrélation	7
2.2.1	Matrice d'auto-corrélation	7
2.2.2	Propriétés de la matrice d'auto-corrélation	8
2.3	Modèles stochastiques	10
2.3.1	Modèles auto-régressifs (AR)	10
2.3.2	Modèle MA	12
2.3.3	Modèle ARMA	12
2.3.4	Discussion	12
2.4	Caractérisation d'un modèle AR (cas de signaux réels)	13
2.4.1	Equations de Yule-Walker	13
2.4.2	Résolution des équations de Yule-Walker (YW)	14
2.5	Processus AR d'ordre 2	14
2.5.1	Stationnarité asymptotique	14
2.5.2	Fonction d'auto-corrélation	15
2.6	Sélection de l'ordre du modèle	17
2.6.1	Critère AIC	17
2.6.2	Minimum description length	18
3	Filtrage de Wiener	19
3.1	Enoncé du problème	19
3.2	Equation de Wiener-Hopf	22
3.2.1	Expression générale	22
3.3	Représentation des performances	24
3.3.1	Erreur quadratique : forme générale	24
3.3.2	Erreur quadratique minimale	24
3.3.3	Forme canonique	25

4	Prédiction linéaire	27
4.1	Prédiction avant (<i>forward prediction</i>)	27
4.1.1	Modèle	27
4.1.2	Erreur de prédiction à un pas	28
4.1.3	Comparaison entre le filtre de Wiener et le filtre prédicteur	29
4.2	Prédiction arrière ou rétrograde à un pas	30
4.2.1	Le problème	30
4.2.2	Optimisation	30
4.2.3	Relation entre prédiction avant et arrière	31
4.2.4	Erreur de prédiction	32
5	Algorithmes adaptatifs	33
5.1	Introduction	33
5.1.1	Objectifs	33
5.1.2	Dérivation par rapport à un vecteur	33
5.2	Algorithme de descente ou <i>steepest descent algorithm</i>	35
5.2.1	Rappel sur les notations	35
5.2.2	Algorithme de descente	35
5.2.3	Stabilité de l'algorithme	36
5.2.4	Vitesse de convergence	37
5.2.5	Erreur quadratique moyenne	38
5.3	Algorithme du gradient stochastique	39
5.3.1	Conception de l'algorithme du gradient stochastique	39
5.3.2	Analyse de la stabilité du LMS	40
5.3.3	Comparaison LMS/Descente du gradient	43
5.3.4	Algorithme LMS normalisé	43
5.4	Algorithme des moindres carrés	43
5.4.1	Equations préliminaires	44
5.4.2	Forme récursive pour le calcul de $\Phi_{\mathbf{u}\mathbf{u}}(n)$ et $\theta(n)$	45
5.4.3	Lemme d'inversion matricielle	45
5.4.4	Initialisation de l'algorithme des moindres carrés récursifs	46
6	Filtre de Kalman discret	47
6.1	Introduction	47
6.1.1	Histoire	47
6.1.2	Equations de base	47
6.2	Erreurs <i>a priori</i> et <i>a posteriori</i>	48
6.2.1	Erreur <i>a priori</i>	48
6.2.2	Equation de mise à jour et erreur <i>a posteriori</i>	48
6.2.3	Optimisation du gain \mathbf{K}_k	49
6.2.4	Matrice de covariance de l'erreur <i>a posteriori</i> optimale	49
6.3	Algorithmes	50
6.3.1	Algorithme 1	50
6.3.2	Forme alternative	50
6.4	Innovation	51
6.4.1	Prédiction de $\hat{\mathbf{z}}_{k/(k-1)}$	52
6.4.2	Innovation	52

6.4.3	Interprétation de la mise à jour	53
6.4.4	Stabilité du filtre de Kalman	54
6.5	Un filtre de Kalman pour éliminer le 50 Hz	54
6.5.1	Position du problème	54
6.5.2	Modèle de génération du signal	54
6.5.3	Algorithme de Kalman	56
6.5.4	Réponse du filtre	56
6.5.5	Analyse fonctionnelle du filtre Notch du second ordre	58
6.5.6	Programmes et résultats sur des ECG	58

Chapitre 1

Introduction

1.1 Problèmes de filtrage

Le filtrage est une opération fondamentale en traitement du signal et en automatique. Par filtrage, on entend généralement plusieurs types de traitements, qu'il convient de préciser :

- le filtrage (*filtering*) proprement dit, qui consiste à extraire de l'information à l'instant t à partir de données disponibles jusqu'à l'instant t inclus ;
- le lissage (*smoothing*) qui consiste à extraire de l'information à l'instant t , à partir de données observées au-delà de l'instant t ; le lissage exploitant plus d'information que le filtrage conduit à des résultats plus performants ;
- la prédiction (*prediction*) qui consiste à prédire la valeur d'une quantité d'intérêt à l'instant $t + \tau$, connaissant cette quantité jusqu'à l'instant t inclus.

Dans ce cours, on se restreindra à des filtres linéaires, c'est-à-dire dont l'information extraite est une combinaison linéaire des observations exploitées. Ceci entraîne en particulier des modèles de filtres très simples, avec des propriétés séduisantes, et permet d'utiliser l'arsenal puissant de l'algèbre linéaire pour formaliser ces modèles.

Le terme *optimal* doit se référer à un critère. Dans ce cadre du filtrage linéaire optimal, le critère usuel est l'erreur quadratique moyenne. Par conséquent, le filtrage linéaire optimal a pour objectif de rechercher le "meilleur" filtre linéaire, c'est-à-dire celui qui fournit une approximation telle que l'erreur quadratique moyenne est minimale.

Enfin, le filtrage linéaire optimal peut être considéré dans un cadre de signaux continus ou discrets. Dans ce cours de durée limitée, et dans le cadre du traitement numérique du signal (le signal est numérisé aussitôt après le capteur et l'étage de conditionnement), nous n'étudierons que la formulation dans le cas de signaux en temps discret.

1.2 Plan du cours

Le plan du cours est le suivant :

1. Introduction
2. Processus stochastiques discrets
3. Filtrage de Wiener

4. Prédiction linéaire
5. Algorithmes adaptatifs
6. Filtre de Kalman

1.3 Références

Le lecteur curieux trouvera facilement des articles de synthèse ou des chapitres de cours ou d'ouvrages sur le Web à partir des mots-clés de cette introduction. Nous lui recommandons les deux ouvrages suivants, en anglais :

1. S. HAYKIN, Adaptive filter theory, Prentice Hall, 1991
2. B. ANDERSON, J. MOORE, Optimal Filtering, Prentice Hall, 1979

Chapitre 2

Processus stochastiques discrets

2.1 Position du problème

On s'intéresse dans ce chapitre aux outils de description des processus stochastiques discrets, échantillonnés de façon uniforme dans le temps. Dans ce cadre, on considère une séquence :

$$u(n), u(n-1), \dots, u(n-M) \quad (2.1)$$

comme une réalisation particulière de $M+1$ échantillons d'un processus aléatoire, où n représente le temps discret. Bien sûr, en théorie, un processus stochastique n'est pas associé à une seule réalisation mais à un nombre infini de réalisations sur lesquelles on peut faire des calculs statistiques, pourvu que l'on connaisse la distribution de probabilité.

En pratique, on supposera fréquemment que le processus est stationnaire, au moins au sens large (à l'ordre 2). Les propriétés (et grandeurs) statistiques sont alors indépendantes de n .

On supposera aussi fréquemment que le processus est ergodique, ce qui permet d'approcher les moyennes statistiques par des moyennes temporelles.

2.2 Matrices d'auto-corrélation

Soit $u(n)$ le terme général (réel ou complexe) d'une série temporelle stationnaire au second ordre. On note $\mathbf{u}(n)$ le vecteur de taille $M \times 1$:

$$\mathbf{u}(n) = (u(n), u(n-1), \dots, u(n-M+1))^T \quad (2.2)$$

où T représente la transposition.

2.2.1 Matrice d'auto-corrélation

Si les valeurs de la série $u(n)$ sont à valeurs complexes, on définit la matrice d'auto-corrélation comme :

$$\mathbf{R}_{\mathbf{u}\mathbf{u}} = E[\mathbf{u}(n)\mathbf{u}(n)^H] = \begin{pmatrix} R_{uu}(0) & R_{uu}(1) & \dots & R_{uu}(M-1) \\ R_{uu}(-1) & R_{uu}(0) & & \vdots \\ \vdots & & & \\ R_{uu}(-M+1) & & & R_{uu}(0) \end{pmatrix} \quad (2.3)$$

où $*$ représente l'opérateur de conjugaison, H représente la transposée conjuguée, $R_{uu}(k) = E[u(n)u^*(n-k)]$. Cette matrice possède la symétrie hermitienne, c'est-à-dire qu'elle vérifie :

$$\mathbf{R}_{\mathbf{u}\mathbf{u}} = \mathbf{R}_{\mathbf{u}\mathbf{u}}^H = \begin{pmatrix} R_{uu}(0) & R_{uu}(1) & \dots & R_{uu}(M-1) \\ R_{uu}^*(1) & R_{uu}(0) & & \vdots \\ \vdots & & & \\ R_{uu}^*(M-1) & & & R_{uu}(0) \end{pmatrix} \quad (2.4)$$

Pour montrer la propriété de symétrie hermitienne, il suffit de montrer que $R_{uu}(k) = R_{uu}^*(-k)$:

$$\begin{aligned} R_{uu}(k) &= E[u(n)u^*(n-k)] \\ &= E[u^*(n)u(n-k)]^* \\ &= E[u(n-k)u^*(n)]^* \\ &= R_{uu}^*(-k). \end{aligned} \quad (2.5)$$

Dans le cas de signaux réels, la matrice est définie par $\mathbf{R}_{\mathbf{u}\mathbf{u}} = E[\mathbf{u}(n)\mathbf{u}(n)^T]$ qui possède la symétrie simple, c'est-à-dire $\mathbf{R}_{\mathbf{u}\mathbf{u}} = \mathbf{R}_{\mathbf{u}\mathbf{u}}^T$:

$$\mathbf{R}_{\mathbf{u}\mathbf{u}} = E[\mathbf{u}(n)\mathbf{u}(n)^T] = \begin{pmatrix} R_{uu}(0) & R_{uu}(1) & \dots & R_{uu}(M-1) \\ R_{uu}(1) & R_{uu}(0) & & \vdots \\ \vdots & & & \\ R_{uu}(M-1) & & & R_{uu}(0) \end{pmatrix} \quad (2.6)$$

2.2.2 Propriétés de la matrice d'auto-corrélation

Propriété 2.2.1 *La matrice d'auto-corrélation d'un processus stochastique discret stationnaire est une matrice de Toeplitz, c'est-à-dire dont les diagonales sont constituées d'éléments identiques.*

Propriété 2.2.2 *La matrice d'auto-corrélation est hermitienne.*

Pour des signaux à valeurs complexes, on a donc les relations suivantes :

$$\begin{aligned} \mathbf{R}_{\mathbf{u}\mathbf{u}} &= \mathbf{R}_{\mathbf{u}\mathbf{u}}^H \\ R_{uu}(-k) &= R_{uu}^*(k). \end{aligned} \quad (2.7)$$

Pour des signaux à valeurs réelles, les relations précédentes deviennent simplement :

$$\begin{aligned} \mathbf{R}_{\mathbf{u}\mathbf{u}} &= \mathbf{R}_{\mathbf{u}\mathbf{u}}^T \\ R_{uu}(-k) &= R_{uu}(k). \end{aligned} \quad (2.8)$$

Propriété 2.2.3 *La matrice d'auto-corrélation d'un processus stochastique discret stationnaire est toujours définie, non négative, et presque toujours définie positive.*

En effet, soit \mathbf{x} un vecteur quelconque de taille $M \times 1$, on note :

$$y = \mathbf{x}^H \mathbf{u}(n) = \mathbf{u}^T(n) \mathbf{x}^* \quad (2.9)$$

et

$$y^* = \mathbf{u}^H(n)\mathbf{x}. \quad (2.10)$$

On a alors :

$$\begin{aligned} E[yy^*] &= E[|y|^2] \\ &= E[\mathbf{x}^H \mathbf{u}(n) \mathbf{u}^H(n) \mathbf{x}] \\ &= \mathbf{x}^H E[\mathbf{u}(n) \mathbf{u}^H(n)] \mathbf{x} \\ &= \mathbf{x}^H \mathbf{R}_{\mathbf{uu}} \mathbf{x}. \end{aligned} \quad (2.11)$$

Puisque $E[|y|^2] \geq 0$, la forme hermitienne $\mathbf{x}^H \mathbf{R}_{\mathbf{uu}} \mathbf{x} \geq 0$, c'est-à-dire qu'elle est définie non négative.

De plus, si pour $\mathbf{x} \neq 0$, la forme satisfait $\mathbf{x}^H \mathbf{R}_{\mathbf{uu}} \mathbf{x} > 0$, alors la matrice $\mathbf{R}_{\mathbf{uu}}$ est définie positive. Cette condition est généralement remplie, à moins qu'il n'y ait des dépendances linéaires entre les $u(n), \dots, u(n - M + 1)$.

Propriété 2.2.4 Si on ordonne les composantes du vecteur $\mathbf{u}(n)$ de façon inverse, c'est-à-dire si on note $\mathbf{u}^B(n) = (u(n - M + 1), \dots, u(n - 1), u(n))^T$, la matrice d'auto-corrélation de \mathbf{u}^B vérifie $\mathbf{R}_{\mathbf{u}^B \mathbf{u}^B} = \mathbf{R}_{\mathbf{uu}}^T$.

En effet, $\mathbf{R}_{\mathbf{u}^B \mathbf{u}^B}$ s'écrit par définition :

$$\mathbf{R}_{\mathbf{u}^B \mathbf{u}^B} = E[\mathbf{u}^B(n) \mathbf{u}^B(n)^H] = \begin{pmatrix} R_{uu}(0) & R_{uu}^*(1) & \dots & R_{uu}^*(M-1) \\ R_{uu}(1) & R_{uu}(0) & & \vdots \\ \vdots & & & \\ R_{uu}(M-1) & & & R_{uu}(0) \end{pmatrix} \quad (2.12)$$

d'où $\mathbf{R}_{\mathbf{u}^B \mathbf{u}^B} = \mathbf{R}_{\mathbf{uu}}^T$.

Dans le cas de signaux réels, compte tenu de la symétrie de $\mathbf{R}_{\mathbf{uu}}$, on a simplement $\mathbf{R}_{\mathbf{u}^B \mathbf{u}^B} = \mathbf{R}_{\mathbf{uu}}$

Propriété 2.2.5 Les matrices d'auto-corrélation de $\mathbf{u}(n)$ de dimensions M et $M + 1$, notées respectivement \mathbf{R}_M et \mathbf{R}_{M+1} (en omettant l'indice \mathbf{uu} pour simplifier l'écriture) sont liées par la relation :

$$\mathbf{R}_{M+1} = \begin{pmatrix} R_{uu}(0) & \vdots & \mathbf{r}^H \\ \dots & \vdots & \dots \\ \mathbf{r} & \vdots & \mathbf{R}_M \end{pmatrix} \quad (2.13)$$

ou bien encore :

$$\mathbf{R}_{M+1} = \begin{pmatrix} \mathbf{R}_M & \vdots & \mathbf{r}^{B*} \\ \dots & \vdots & \dots \\ \mathbf{r}^{BT} & \vdots & R_{uu}(0) \end{pmatrix} \quad (2.14)$$

avec $\mathbf{r} = E[\mathbf{u}(n)u^*(n+1)]$, $\mathbf{r}^H = (R_{uu}(1)R_{uu}(2) \dots R_{uu}(M))$ et $\mathbf{r}^{BT} = (R_{uu}(-M)R_{uu}(-M+1) \dots R_{uu}(-1))$.

2.3 Modèles stochastiques

Dans ce paragraphe, nous nous intéresserons à trois types de modèles de filtres :

- des filtres auto-régressifs (AR),
- des filtres à moyenne mobile (MA, pour *moving average*),
- des filtres auto-régressifs à moyenne mobile (ARMA).

2.3.1 Modèles auto-régressifs (AR)

Définitions

Un signal $\{u(n)\}$ est la réalisation d'un processus auto-régressif d'ordre M si $u(n)$ s'écrit :

$$u(n) = \sum_{k=1}^M w_k u(n-k) + v(n) \quad (2.15)$$

où les $w_k, k = 1, \dots, M$ sont les coefficients du modèle et $v(n)$ est un bruit blanc.

On peut aussi écrire :

$$u(n) - \sum_{k=1}^M w_k u(n-k) = v(n) \quad (2.16)$$

ou bien, en posant $a_0 = 1$ et $a_k = -w_k, k = 1, \dots, M$:

$$\sum_{k=0}^M a_k u(n-k) = v(n). \quad (2.17)$$

Enfin, en utilisant l'opérateur de retard z^{-1} , on peut écrire :

$$\begin{aligned} u(n) &= \sum_{k=1}^M w_k z^{-k} u(n) + v(n) \\ v(n) &= u(n) [1 - \sum_k w_k z^{-k}] \end{aligned} \quad (2.18)$$

soit en utilisant les notations a_k :

$$\begin{aligned} u(n) [1 - \sum_k w_k z^{-k}] &= v(n) \\ H_A(z) u(n) &= v(n) \end{aligned} \quad (2.19)$$

avec $H_A(z) = \sum_{k=0}^M a_k z^{-k}$.

Cette équation donne la relation entre un échantillon $v(n)$ et $u(n)$. Pour les séquences $\{v(n)\}$ et $\{u(n)\}$, en notant :

$$U(z) = \sum_{n=0}^{+\infty} u(n) z^{-n} \text{ et } V(z) = \sum_{n=0}^{+\infty} v(n) z^{-n}, \quad (2.20)$$

on a :

$$H_A(z) U(z) = V(z). \quad (2.21)$$

On peut interpréter ce résultat de deux façons :

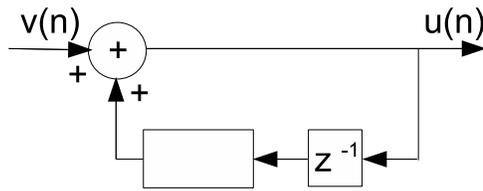


FIGURE 2.1 – Filtre auto-régressif (AR)

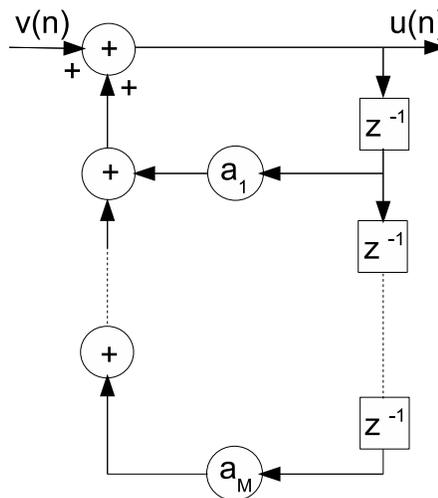


FIGURE 2.2 – Filtre auto-régressif (AR) d'ordre M

1. Etant donné le bruit blanc $\{v(n)\}$, on peut générer le signal auto-régressif $\{u(n)\}$ par la relation :

$$U(z) = V(z)/H_A(z) = H_G(z)V(z) \quad (2.22)$$

où $H_G(z)$, inverse du filtre RIF $H_A(z)$, est donc un filtre “générateur” à réponse impulsionnelle infinie (RII ou IIR en Anglais);

2. Etant donné le processus auto-régressif $\{u(n)\}$, on peut utiliser le filtre $H_A(z)$ pour produire un bruit blanc $\{v(n)\}$: $H_A(z)$ est alors un filtre “analyseur” à réponse impulsionnelle finie (RIF ou FIR en Anglais),

Un filtre AR peut être représenté globalement par le schéma de la figure 2.1 ; plus précisément pour un filtre AR d'ordre M , on a la structure de la figure 2.2.

Zéros, pôles et stabilité

Le filtre $H_A(z) = \sum_{k=0}^M a_k z^{-k}$ est un filtre tout-zéro (*all-zero filter*), car il ne possède que des zéros (exactement M), et il est complètement défini à partir de la position de ses zéros. Ce filtre est implicitement stable, c'est-à-dire que la réponse à une entrée d'énergie finie est à énergie finie.

Au contraire, le filtre $H_G(z) = 1/H_A(z)$ est un filtre tout-pôle (*all-pole filter*). Les pôles de $H_G(z)$ sont bien évidemment les zéros de $H_A(z)$. Ce filtre est stable si et seulement si tous les pôles de $H_G(z)$ sont situés à l'intérieur du cercle unité dans le plan z .

2.3.2 Modèle MA

Le filtre générateur d'un signal MA, $\{u(n)\}$ est un filtre tout-zéro attaqué par un bruit blanc. Le signal à l'instant discret n vaut alors :

$$u(n) = v(n) + b_1v(n-1) + \dots + b_Kv(n-K) = \sum_{k=0}^K b_kv(n-k), \quad (2.23)$$

où $v(n)$ est un bruit blanc de moyenne nulle et de variance σ_v^2 et où les b_k sont les coefficients du filtre. On peut aussi écrire avec les mêmes notations que précédemment :

$$U(z) = H_{MA}(z)V(z). \quad (2.24)$$

De façon inverse, on peut chercher à blanchir un signal MA $u(n)$ selon la relation :

$$V(z) = \frac{1}{H_{MA}}U(z) = H_B(z)U(z), \quad (2.25)$$

où le filtre $H_B(z)$ est un filtre tout-pôle.

2.3.3 Modèle ARMA

C'est la combinaison des parties AR et MA, vues précédemment. Un processus ARMA $\{u(n)\}$ satisfait donc l'équation :

$$u(n) + a_1u(n-1) + \dots + a_Mu(n-M) = v(n) + b_1v(n-1) + \dots + b_Kv(n-K), \quad (2.26)$$

où les a_m et les b_k sont les coefficients des parties AR et MA, respectivement, et où $v(n)$ est un bruit. En factorisant dans z , on a :

$$H_{AR}(z)U(z) = H_{MA}(z)V(z), \quad (2.27)$$

d'où :

$$\frac{U(z)}{V(z)} = \frac{H_{MA}(z)}{H_{AR}(z)}. \quad (2.28)$$

Dans le filtre ARMA, les zéros sont donnés par $H_{MA}(z)$ et les pôles par $H_{AR}(z)$.

2.3.4 Discussion

Le modèle AR présente un avantage calculatoire sur les modèles MA et ARMA dans la mesure où les coefficients AR sont solutions d'un système linéaire d'équations appelé équations de Yule-Walker. Au contraire, le calcul des coefficients MA et ARMA requiert la résolution d'équations non linéaires. Cependant, le modèle ARMA est plus riche que le modèle AR et peut s'avérer indispensable dans de nombreuses situations.

2.4 Caractérisation d'un modèle AR (cas de signaux réels)

Un modèle (ou un processus) AR est complètement caractérisé par :

- la variance du bruit blanc $v(n)$ qui excite le filtre, notée σ_v^2 ,
- les coefficients AR : a_1, a_2, \dots, a_M (rappelons que $a_0 = 1$).

On a alors l'équation :

$$u(n) \left(1 + \sum_{k=1}^M a_k z^{-k} \right) = v(n) \quad (2.29)$$

ou bien sous la forme développée :

$$u(n) + a_1 u(n-1) + \dots + a_M u(n-M) = v(n), \quad (2.30)$$

où $v(n)$ est un bruit blanc de variance σ_v^2 .

2.4.1 Equations de Yule-Walker

Multiplions l'équation précédente par $u(n-p)$, $p > 0$, et calculons l'espérance :

$$E \left[\sum_{k=0}^M a_k u(n-k) u(n-p) \right] = E[v(n)u(n-p)], \quad p = 1, \dots, M. \quad (2.31)$$

On voit apparaître les auto-corrélations de u , et les inter-corrélations entre v et u . Puisque, pour $p > 0$, $u(n-p)$ ne dépend pas de $v(n)$, les intercorrélations sont nulles sauf en $p = 0$. On a donc :

$$E[v(n)u(n-p)] = \begin{cases} \sigma_v^2 & p = 0 \\ 0 & p > 0 \end{cases} \quad (2.32)$$

En écrivant l'équation pour $p = 1, \dots, M$, on obtient le système de M équations :

$$\sum_{k=0}^M a_k E[u(n-k)u(n-p)] = 0 \quad p = 1, 2, \dots, M \quad (2.33)$$

En posant $w_k = -a_k$, on peut mettre cette équation sous la forme :

$$\sum_{k=1}^M w_k E[u(n-k)u(n-p)] = E[u(n)u(n-p)] \quad p = 1, 2, \dots, M. \quad (2.34)$$

En posant maintenant :

$$\begin{aligned} E[u(n-k)u(n-p)] &= R_{uu}(p-k) = R(p-k) \\ E[u(n)u(n-p)] &= R_{uu}(p) = R(p) \end{aligned} \quad (2.35)$$

le système d'équations peut s'écrire :

$$\begin{pmatrix} R(0) & R(1) & \dots & R(M-1) \\ R(1) & R(0) & & R(M-2) \\ \vdots & \dots & & \vdots \\ R(M-1) & R(M-2) & \dots & R(0) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{pmatrix} = \begin{pmatrix} R(1) \\ R(2) \\ \vdots \\ R(M) \end{pmatrix} \quad (2.36)$$

que l'on peut mettre sous la forme matricielle :

$$\mathbf{R}\mathbf{w} = \mathbf{r}, \quad (2.37)$$

en posant $\mathbf{w} = (w_1 \dots w_M)^T$, et $\mathbf{r} = (R(1) \dots R(M))^T$. Cette équation porte le nom d'équation de Yule-Walker.

2.4.2 Résolution des équations de Yule-Walker (YW)

Calcul des coefficients w_k . Si la matrice \mathbf{R} est régulière, on calcule simplement :

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{r}. \quad (2.38)$$

On remarque que l'équation de YW requiert la connaissance de $R(0), \dots, R(M)$, c'est-à-dire de $M + 1$ termes d'auto-corrélation. L'équation (2.38) permet de calculer les M coefficients w_1, \dots, w_M . Il reste à calculer la variance σ_v^2 .

Calcul de la variance. Pour $p = 0$, en tenant compte de l'équation du modèle AR (2.30), l'équation (2.31) devient :

$$E\left[\sum_{k=0}^M a_k u(n-k)u(n)\right] = E[v(n)u(n)] = \sigma_v^2. \quad (2.39)$$

En utilisant la propriété de linéarité de l'espérance, on arrive à :

$$\sum_{k=0}^M a_k E[u(n-k)u(n)] = \sigma_v^2, \quad (2.40)$$

soit, avec les notations $R(k) = E[u(n-k)u(n)]$:

$$\sum_{k=0}^M a_k R(k) = \sigma_v^2. \quad (2.41)$$

Cette $(M + 1)$ -ème équation permet donc de calculer la variance, à partir de $R(0), \dots, R(M)$ et des coefficients a_1, \dots, a_M . Le filtre AR est ainsi complètement spécifié.

2.5 Processus AR d'ordre 2

Soit le processus AR d'ordre 2 défini par :

$$u(n) + a_1 u(n-1) + a_2 u(n-2) = v(n), \quad (2.42)$$

où $v(n)$ est un bruit blanc de moyenne nulle et de variance σ_v^2 .

2.5.1 Stationnarité asymptotique

Ce calcul a été fait en cours de traitement numérique du signal en 3i4.

Le processus AR est stationnaire asymptotiquement si les racines de l'équation caractéristique :

$$1 + a_1 z^{-1} + a_2 z^{-2} = 0 \quad (2.43)$$

sont situées à l'intérieur du cercle unité.

Réolvons l'équation polynomiale :

$$z^2 + a_1 z^1 + a_2 = 0 \quad (2.44)$$

Le discriminant vaut $\Delta = a_1^2 - 4a_2$, et l'on a les deux racines :

$$\rho_{1,2} = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2}}{2}. \quad (2.45)$$

On doit donc vérifier :

$$|\rho_1| < 1 \text{ et } |\rho_2| < 1. \quad (2.46)$$

Si $\Delta = a_1^2 - 4a_2 < 0$, on a deux racines complexes conjuguées, telles que :

$$|\rho_1|^2 = |\rho_2|^2 = \frac{a_1^2 - (a_1^2 - 4a_2)}{4} = a_2, \quad (2.47)$$

donc la condition pour la stationarité asymptotique est $a_2 < 1$.

Si $\Delta = a_1^2 - 4a_2 > 0$, on a deux racines réelles. Si $a_1 > 0$, on a $|\rho_1| < |\rho_2|$. La condition de stationarité asymptotique est alors :

$$|\rho_2| < 1 \Leftrightarrow \left| \frac{-a_1 - \sqrt{a_1^2 - 4a_2}}{2} \right| < 1 \quad (2.48)$$

La résolution de la dernière inégalité donne :

$$\begin{aligned} \frac{a_1 + \sqrt{a_1^2 - 4a_2}}{2} &< 1 \\ \sqrt{a_1^2 - 4a_2} &< 2 - a_1 \\ a_1^2 - 4a_2 &< 4 - 4a_1 + a_1^2 \\ a_1 - 1 &< a_2 \end{aligned} \quad (2.49)$$

Si $a_1 < 0$, on a $\rho_1 > \rho_2$ et $\rho_1 > 0$. La condition de stationarité asymptotique, $|\rho_1| < 1$, s'écrit :

$$\begin{aligned} \frac{-a_1 + \sqrt{a_1^2 - 4a_2}}{2} &< 1 \\ \sqrt{a_1^2 - 4a_2} &< 1 + a_1 \\ a_1^2 - 4a_2 &< 4 + 4a_1 + a_1^2 \\ -a_1 - 1 &< a_2. \end{aligned} \quad (2.50)$$

On peut résumer les trois conditions par le schéma de la figure 2.3. Dans cette figure, la limite entre les régions correspondant aux racines réelles et complexes est la parabole d'équation $a_1^2 - 4a_2 = 0$, la région "complexe" étant située au-dessus.

2.5.2 Fonction d'auto-corrélation

Soit l'équation du modèle (2.42) : $u(n) + a_1 u(n-1) + a_2 u(n-2) = v(n)$. En multipliant par $u(n-m)$, $m > 0$ et en prenant l'espérance, on obtient :

$$R(m) + a_1 R(m-1) + a_2 R(m-2) = 0. \quad (2.51)$$

Pour $m = 0$, on a simplement :

$$R(0) + a_1 R(-1) + a_2 R(-2) = \sigma_v^2. \quad (2.52)$$

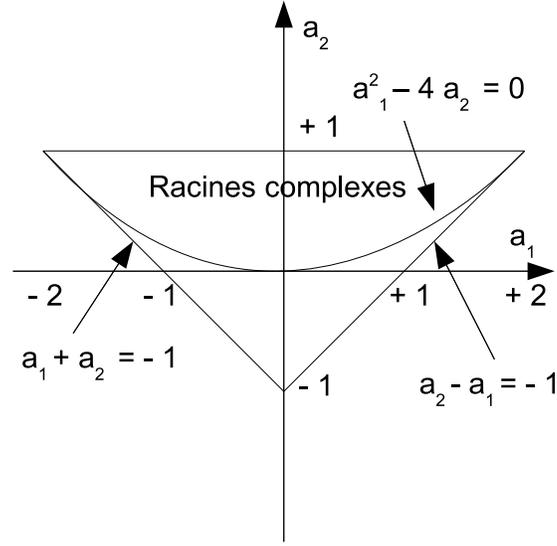


FIGURE 2.3 – Conditions de stationnarité d'une filtre auto-régressif (AR) d'ordre 2

Les équations de Yule-Walker s'écrivent :

$$\begin{pmatrix} R(0) & R(1) \\ R(1) & R(0) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} R(1) \\ R(2) \end{pmatrix} \quad (2.53)$$

avec $w_k = -a_k$.

On en déduit la solution :

$$\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \frac{1}{R(0)^2 - R(1)^2} \begin{pmatrix} R(0) & -R(1) \\ -R(1) & R(0) \end{pmatrix} \begin{pmatrix} R(1) \\ R(2) \end{pmatrix} \quad (2.54)$$

c'est-à-dire :

$$\begin{cases} w_1 = \frac{R(1)(R(0) - R(2))}{R(0)^2 - R(1)^2} \\ w_2 = \frac{R(2)R(0) - R(1)^2}{R(0)^2 - R(1)^2} \end{cases} \quad (2.55)$$

A partir des équations de YW, on peut aussi exprimer $R(1)$ et $R(2)$ en fonction de $a_1 = -w_1$, $a_2 = -w_2$ et de $R(0) = \sigma_u^2$:

$$\begin{aligned} R(1) &= -R(0)a_1 - R(1)a_2 \\ R(1) &= -\sigma_u^2 a_1 - R(1)a_2 \\ R(1)(1 - a_1) &= -a_1 \sigma_u^2 \end{aligned} \quad (2.56)$$

d'où :

$$R(1) = \frac{-a_1}{1 + a_2} \sigma_u^2. \quad (2.57)$$

On peut maintenant calculer $R(2)$ de façon similaire :

$$\begin{aligned} R(2) &= -R(1)a_1 - R(0)a_2 \\ R(2) &= -R(1)a_1 - \sigma_u^2 a_2 \\ R(2)(1 - a_1) &= \frac{a_1^2}{1 + a_2} \sigma_u^2 - a_2 \sigma_u^2 \end{aligned} \quad (2.58)$$

d'où :

$$R(2) = \left(\frac{a_1^2}{1+a_2} - a_2 \right) \sigma_u^2. \quad (2.59)$$

Enfin, on termine en calculant la variance du bruit σ_v^2 , en prenant l'équation (2.52) : $R(0) + a_1R(-1) + a_2R(-2) = \sigma_v^2$, obtenue pour $m = 0$.

$$\begin{aligned} R(0) + a_1R(-1) + a_2R(-2) &= \sigma_v^2 \\ \sigma_u^2 + a_1R(1) + a_2R(2) &= \sigma_v^2 \\ \sigma_u^2 \left(1 - \frac{a_1^2}{1+a_2} + \frac{a_1a_2}{1+a_2} - a_2^2 \right) &= \sigma_v^2. \end{aligned} \quad (2.60)$$

Après des calculs simples, on arrive à :

$$\sigma_u^2 = \left(\frac{1+a_2}{1-a_2} \right) \frac{\sigma_v^2}{(1+a_2)^2 - a_1^2} \quad (2.61)$$

2.6 Sélection de l'ordre du modèle

Dans les paragraphes précédents, l'ordre du modèle AR a été fixé à M (et à $M = 2$ dans le paragraphe 2.5). En général, l'ordre n'est pas connu et on pourrait être tenté d'augmenter l'ordre du filtre pour améliorer la précision. En réalité, il y a un compromis entre l'ordre du filtre (complexité) et la qualité de l'estimation (variance du terme σ_v^2).

Ce problème a été beaucoup étudié au cours des 40 dernières années. Dans ce paragraphe, on présente simplement les principes des critères AIC et BIC d'Akaike, et du *Minimum Description Length* de Rissanen.

2.6.1 Critère AIC

Soit $u_i = u(i)$, $i = 1, \dots, N$, les observations d'un processus stationnaire discret, de densité de probabilité (ddp) $g(u_i)$. On modélise le processus par un modèle AR d'ordre m de paramètre $\hat{\theta}_m$:

$$\hat{\theta}_m = (\hat{\theta}_1, \dots, \hat{\theta}_m)^T. \quad (2.62)$$

Soit $f(u_i/\hat{\theta}_m)$ la ddp de u_i sachant le vecteur des paramètres $\hat{\theta}_m$. Akaike a proposé le critère informationnel suivant :

$$AIC(m) = -2 \max \sum_{i=1}^N \ln f(u_i/\hat{\theta}_m) + 2m. \quad (2.63)$$

On utilise le critère $AIC(m)$ en choisissant la valeur m_0 qui minimise ce critère, c'est-à-dire :

$$m_0 = \operatorname{argmax}_m AIC(m). \quad (2.64)$$

En pratique, le premier terme de (2.63) décroît rapidement avec m , le second croît linéairement. L'allure typique de la fonction $AIC(m)$ est donnée à la figure 2.4.

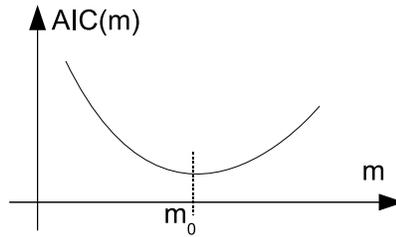


FIGURE 2.4 – Allure typique de la variation du critère AIC en fonction du nombre de paramètres m

2.6.2 Minimum description length

L'idée de Rissanen repose sur la question suivante : combien faut-il de bits (donc quelle est la longueur du mot binaire) pour encoder l'ensemble d'échantillons $u(1), \dots, u(N)$?

Bien sûr, si la suite est aléatoire, ce nombre est grand ; en revanche, si la suite est générée par un modèle simple, il suffit d'avoir les paramètres du modèle, donc très peu de bits sont nécessaires. D'où le nom de Minimum Description Length (MDL).

Asymptotiquement, c'est-à-dire pour $N \rightarrow \infty$, on montre que :

$$MDL(m) = - \max \sum_{i=1}^N \ln f(u_i / \hat{\theta}_m) + \frac{1}{2} m \ln N. \quad (2.65)$$

On remarque que les deux termes, comme dans le critère $AIC(m)$, varient en sens contraire lorsque m augmente. Ici, l'ordre du modèle est pondéré par le nombre d'échantillons : ceci indique que si N est petit, il est illusoire de prendre m trop grand, car il sera impossible d'estimer correctement m paramètres. Souvent de façon empirique, pour estimer m paramètres, on considère qu'il faut environ $N \approx 10m$ échantillons.

Chapitre 3

Filtrage de Wiener

Dans ce chapitre, on considère des signaux à temps discrets et à valeurs complexes. Le choix de travailler avec des nombres complexes est utile pour les applications en télécommunication. Pour des signaux réels, les relations se simplifieront et nous donneront fréquemment ces relations simplifiées. Le filtrage de Wiener doit son nom au scientifique Norbert Wiener (figure 3.1), né au USA le 26 novembre 1894 et décédé à Stockholm, en Suède, le 18 mars 1964.

3.1 Enoncé du problème

On dispose d'une série d'observations à valeurs complexes :

$$u(0), u(1), \dots \quad (3.1)$$

et d'un filtre linéaire discret dont les coefficients, à valeurs complexes, sont :

$$w_0, w_1, \dots \quad (3.2)$$

et qui fournit un signal de sortie $y(n)$ à valeur complexe :

$$y(n) = w_0^* u(n) + w_1^* u(n-1) + \dots \quad (3.3)$$

où w_k^* est le complexe conjugué de w_k . L'objectif est d'estimer les paramètres w_k de sorte que $y(n)$ soit une bonne estimation d'une sortie désirée $d(n)$. On définit $e(n)$, l'erreur d'estimation (à valeur complexe) à l'instant n :

$$e(n) = d(n) - y(n). \quad (3.4)$$

La solution est recherchée avec les contraintes suivantes :

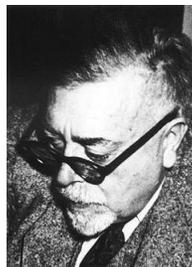


FIGURE 3.1 – Norbert Wiener, inventeur du filtrage optimal et du filtre de Wiener

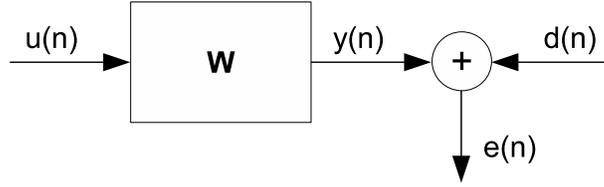


FIGURE 3.2 – Représentation schématique d'un filtre de Wiener

- le filtre W est linéaire, à temps discret,
- le critère est le minimum de l'erreur quadratique moyenne (MMSE pour *Minimum Mean Square Error*), c'est-à-dire le minimum de $E[|e(n)|^2]$.

Ce problème peut être associé au schéma de la figure 3.2.

Equations de base

Selon la figure 3.2, avec les hypothèses de linéarité et de temps discret de $u(n)$, on peut écrire :

$$y(n) = \sum_{k=0}^{+\infty} w_k^* u(n-k), \quad n = 0, 1, \dots \quad (3.5)$$

On peut voir l'équation (3.5) comme le produit scalaire dans l'espace de Hilbert de dimension infinie :

$$\langle \mathbf{u}(n), \mathbf{w} \rangle = \sum_{k=0}^{+\infty} u(n-k) w_k^*, \quad (3.6)$$

avec $\mathbf{u}(n) = (u(n) u(n-1) \dots)^T$ et $\mathbf{w} = (w_0 w_1 \dots)^T$.

L'erreur d'estimation à l'instant n est égale à $e(n) = d(n) - y(n)$. L'erreur quadratique moyenne, J , est la fonction de coût que l'on cherche à minimiser :

$$J = E[e(n)e(n)^*] = E[|e(n)|^2]. \quad (3.7)$$

Optimisation

Compte tenu de la nature complexe des données, on peut écrire :

$$w_k = a_k + j b_k, \quad k = 0, 1, \dots \quad (3.8)$$

et on peut définir un opérateur de gradient complexe :

$$\nabla_k = \frac{\partial}{\partial a_k} + j \frac{\partial}{\partial b_k}, \quad k = 0, 1, \dots \quad (3.9)$$

Appliqué à la fonction de coût J , on a :

$$\nabla_k J = \frac{\partial J}{\partial a_k} + j \frac{\partial J}{\partial b_k}, \quad k = 0, 1, \dots \quad (3.10)$$

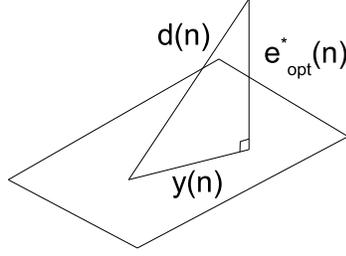


FIGURE 3.3 – Représentation géométrique du théorème de la projection orthogonale

Puisque la fonction est l'espérance d'un produit, $J = E[e(n)e^*(n)]$, en utilisant la propriété de linéarité de l'espérance et les règles de dérivation d'un produit, on a :

$$\nabla_k J = E \left[e^*(n) \frac{\partial e(n)}{\partial a_k} + e(n) \frac{\partial e^*(n)}{\partial a_k} + j e^*(n) \frac{\partial e(n)}{\partial b_k} + j e(n) \frac{\partial e^*(n)}{\partial b_k} \right]. \quad (3.11)$$

En exprimant les termes $e(n)$ et $e^*(n)$ en fonction de a_k et b_k , et en dérivant, on trouve après des calculs algébriques simples :

$$\begin{cases} \frac{\partial e(n)}{\partial a_k} = -u(n-k) \\ \frac{\partial e^*(n)}{\partial a_k} = -u^*(n-k) \\ \frac{\partial e(n)}{\partial b_k} = +j u(n-k) \\ \frac{\partial e^*(n)}{\partial b_k} = -j u^*(n-k). \end{cases} \quad (3.12)$$

En reportant les résultats (3.12) dans l'équation (3.11), on a simplement :

$$\nabla_k J = -2E[u(n-k)e^*(n)], \quad k = 0, 1, \dots \quad (3.13)$$

La solution optimale correspond à une erreur quadratique minimale, notée $e_{opt}(n)$, telle que :

$$\nabla_k J = 0, \quad k = 0, 1, \dots \quad (3.14)$$

ou encore :

$$E[u(n-k)e_{opt}^*(n)] = 0, \quad k = 0, 1, \dots \quad (3.15)$$

On remarque que l'erreur optimale est orthogonale à toutes les entrées $u(n)$, $u(n-1)$, ... On retrouve ainsi le théorème de la projection orthogonale (figure 3.3).

En corrolaire à cette remarque, l'approximation optimale, notée $y_{opt}(n)$, qui s'écrit :

$$y_{opt}(n) = \sum_{k=0}^{+\infty} w_{opt}^* u(n-k), \quad (3.16)$$

est donc une combinaison linéaire de termes $u(n-k)$ qui sont tous orthogonaux à $e_{opt}(n)$. Par conséquent, $y_{opt}(n)$ est elle-même orthogonale à $e_{opt}(n)$, c'est-à-dire que l'on peut écrire :

$$E[y_{opt}(n)e_{opt}^*(n)] = 0. \quad (3.17)$$

Notations

Soit \mathcal{U}_n l'espace engendré par tous les $u(k)$, $k \leq n$, on notera $\hat{d}(n/\mathcal{U}_n)$ l'estimation optimale au sens de l'erreur quadratique moyenne minimale (MMSE), obtenue à partir des données jusqu'à l'instant n inclus. D'après les résultats précédents, on peut donc écrire :

$$y_{opt}(n) = \hat{d}(n/\mathcal{U}_n). \quad (3.18)$$

Calcul de l'erreur quadratique minimale

On rappelle que $e(n) = d(n) - y(n)$ et qu'à l'optimum, on a :

$$e_{opt}(n) = d(n) - y_{opt}(n) = d(n) - \hat{d}(n/\mathcal{U}_n). \quad (3.19)$$

La valeur minimale de l'erreur quadratique moyenne est alors :

$$J_{min} = E[|e_{opt}(n)|^2]. \quad (3.20)$$

D'après (3.19), on peut écrire $d(n) = \hat{d}(n/\mathcal{U}_n) + e_{opt}(n)$. De plus, on sait que $\hat{d}(n/\mathcal{U}_n) = y_{opt}(n) \perp e_{opt}(n)$. On en déduit donc par le théorème de Pythagore l'égalité :

$$E[|d(n)|^2] = E\left[|\hat{d}(n/\mathcal{U}_n)|^2\right] + E[|e_{opt}(n)|^2] \quad (3.21)$$

c'est-à-dire :

$$\sigma_d^2 = \sigma_{\hat{d}}^2 + J_{min}. \quad (3.22)$$

On en déduit finalement :

$$J_{min} = \sigma_d^2 - \sigma_{\hat{d}}^2. \quad (3.23)$$

En pratique, on normalise l'expression en divisant par σ_d^2 , d'où l'erreur quadratique moyenne normalisée minimale :

$$\epsilon = \frac{J_{min}}{\sigma_d^2} = 1 - \left(\frac{\sigma_{\hat{d}}}{\sigma_d}\right)^2. \quad (3.24)$$

On remarque que cette erreur normalisée vérifie $0 \leq \epsilon \leq 1$. Si $\epsilon = 0$, on a alors $\sigma_{\hat{d}}^2 = \sigma_d^2$, ce qui correspond à une adéquation parfaite entre d et \hat{d} .

3.2 Equation de Wiener-Hopf

3.2.1 Expression générale

A partir du principe d'orthogonalité (3.15), de la définition de $e_{opt}(n)$ et de $y(n)$, l'équation (3.15) devient :

$$E[u(n-k)(d^*(n) - \sum_j w_{opt,j} u^*(n-j))] = 0, \quad k = 0, 1, \dots \quad (3.25)$$

En développant l'espérance, on arrive à :

$$E[u(n-k)(d^*(n))] = \sum_j w_{opt,j} E[u(n-k)u^*(n-j)] \quad k = 0, 1, \dots \quad (3.26)$$

Or, pour un signal $u(n)$ stationnaire, le terme $E[u(n-k)u^*(n-j)]$ n'est autre que le terme $R_{uu}(j-k)$ de la matrice d'auto-corrélation. Le terme $E[u(n-k)(d^*(n))]$ est l'inter-corrélation entre l'entrée du filtre et la sortie désirée, que l'on note $R_{ud}(-k)$. L'équation (3.26) peut donc s'écrire :

$$\sum_j w_{opt,j} R_{uu}(j-k) = R_{ud}(-k), \quad k = 0, 1, \dots \quad (3.27)$$

Cet ensemble infini d'équations porte le nom d'équations de Wiener-Hopf.

Solution pour un filtre FIR

Si on suppose que le filtre est à réponse impulsionnelle finie (RIF) de longueur M , on peut écrire les équations (3.27) sous forme d'un système linéaire de M équations à M inconnues :

$$\sum_{j=0}^{M-1} w_{opt,j} R_{uu}(j-k) = R_{ud}(-k), \quad k = 0, 1, \dots, M-1 \quad (3.28)$$

En posant les notations $\mathbf{u}(n) = (u(n)u(n-1) \dots u(n-M+1))^T$ et $\mathbf{R}_{ud} = (R_{ud}(0)R_{ud}(-1) \dots R_{ud}(-M+1))^T$ (on rappelle que $\mathbf{R}_{ud} = E[d^*(n)\mathbf{u}(n)]$) et en remarquant que la matrice d'auto-corrélation s'écrit :

$$\mathbf{R}_{uu} = E[\mathbf{u}(n)\mathbf{u}^H(n)] = \begin{pmatrix} R_{uu}(0) & R_{uu}(1) & \dots & R_{uu}(M-1) \\ R_{uu}^*(1) & R_{uu}(0) & & \vdots \\ \vdots & & & \\ R_{uu}^*(M-1) & R_{uu}^*(M-2) & & R_{uu}(0) \end{pmatrix} \quad (3.29)$$

les équations de Wiener-Hopf s'écrivent sous la forme matricielle :

$$\mathbf{R}_{uu}\mathbf{w}_{opt} = \mathbf{R}_{ud}, \quad (3.30)$$

avec $\mathbf{w}_{opt} = (w_{opt,0}w_{opt,1} \dots w_{opt,M-1})^T$. Si la matrice d'auto-corrélation \mathbf{R}_{uu} est régulière, \mathbf{R}_{uu}^{-1} existe et la solution optimale est donnée par :

$$\mathbf{w}_{opt} = \mathbf{R}_{uu}^{-1}\mathbf{R}_{ud}. \quad (3.31)$$

L'estimation de \mathbf{w}_{opt} requiert donc la connaissance de la matrice d'auto-corrélation (c'est-à-dire des statistiques à l'ordre 2) des entrées du filtre et de l'inter-corrélation entre les entrées du filtre et la sortie désirée. Dans le cas d'un filtre d'ordre M (M coefficients), il faut donc estimer d'abord les M termes pour \mathbf{R}_{uu} (compte tenu de la structure Toeplitz et de la symétrie) et les M coefficients de \mathbf{R}_{ud} .

Cas de signaux réels

Dans le cas de signaux réels, il suffit de remplacer les termes conjugués par les termes simples et H par T :

$$\begin{aligned}
 E[\mathbf{u}(n)\mathbf{u}^H(n)] &\rightarrow E[\mathbf{u}(n)\mathbf{u}^T(n)] \\
 E[\mathbf{u}(n)d^*(n)] &\rightarrow E[\mathbf{u}(n)d(n)] \\
 E[|e(n)|^2] &\rightarrow E[e^2(n)]
 \end{aligned}$$

$$\mathbf{R}_{\mathbf{u}\mathbf{u}} = \begin{pmatrix} R_{uu}(0) & R_{uu}(1) & \dots & R_{uu}(M-1) \\ R_{uu}(1) & R_{uu}(0) & & \vdots \\ \vdots & & & \\ R_{uu}(M-1) & R_{uu}(M-2) & & R_{uu}(0) \end{pmatrix} \quad (3.32)$$

En particulier, les matrices d'auto-corrélation - à symétrie hermitienne dans le cas complexe - sont de simples matrices symétriques dans le cas réel.

3.3 Représentation des performances

3.3.1 Erreur quadratique : forme générale

L'erreur $e(n) = d(n) - y(n)$ s'écrit encore $d(n) - \sum_k w_k^* u(n-k)$ ou bien en notations vectorielles

$$e(n) = d(n) - \mathbf{w}^H \mathbf{u}(n) \quad (3.33)$$

avec $\mathbf{w}^H = (w_0^*, w_1^* \dots)$ et $\mathbf{u}(n) = (u(n)u(n-1) \dots)^T$.

On peut calculer l'erreur quadratique J :

$$\begin{aligned}
 J &= E[e(n)e^*(n)] \\
 &= E[(d(n) - \mathbf{w}^H \mathbf{u}(n))(d^*(n) - \mathbf{w}^T \mathbf{u}^*(n))] \\
 &= E[|d(n)|^2] - E[d(n)\mathbf{w}^T \mathbf{u}^*(n)] - E[\mathbf{w}^H \mathbf{u}(n)d^*(n)] + E[(\mathbf{w}^H \mathbf{u}(n))(\mathbf{w}^T \mathbf{u}^*(n))^*] \\
 &= \sigma_d^2 - E[d(n)\mathbf{u}^H(n)]\mathbf{w} - \mathbf{w}^H E[\mathbf{u}(n)d^*(n)] + \mathbf{w}^H E[\mathbf{u}(n)\mathbf{u}^H(n)]\mathbf{w} \\
 &= \sigma_d^2 - \mathbf{p}^H \mathbf{w} - \mathbf{w}^H \mathbf{p} + \mathbf{w}^H \mathbf{R}_{\mathbf{u}\mathbf{u}} \mathbf{w},
 \end{aligned} \quad (3.34)$$

où $\mathbf{R}_{\mathbf{u}\mathbf{u}}$ est la matrice d'auto-corrélation de $\mathbf{u}(n)$ que l'on a déjà utilisée plusieurs fois, et $\mathbf{p} = \mathbf{R}_{\mathbf{u}d} = (R_{ud}(0)R_{ud}(-1) \dots)^T$.

3.3.2 Erreur quadratique minimale

Pour $\mathbf{w} = \mathbf{w}_{opt}$, l'erreur quadratique moyenne J prend la valeur minimale $J_{min} = \sigma_d^2 - \sigma_{\hat{d}}^2$, montrée en (3.23). Le terme $\sigma_{\hat{d}}^2$ peut s'exprimer :

$$\begin{aligned}
 \sigma_{\hat{d}}^2 &= E[(\mathbf{w}_{opt}^H \mathbf{u}(n))(\mathbf{w}_{opt}^H \mathbf{u}(n))^H] \\
 &= E[\mathbf{w}_{opt}^H \mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}_{opt}] \\
 &= \mathbf{w}_{opt}^H \mathbf{R}_{\mathbf{u}\mathbf{u}} \mathbf{w}_{opt}.
 \end{aligned} \quad (3.35)$$

En utilisant l'équation de Wiener-Hopf :

$$\mathbf{R}_{\mathbf{u}\mathbf{u}} \mathbf{w}_{opt} = \mathbf{p} = E[\mathbf{u}(n)d^*(n)], \quad (3.36)$$

on peut écrire (3.35) sous la forme :

$$\sigma_d^2 = \mathbf{w}_{opt}^H \mathbf{p} = \mathbf{p}^H \mathbf{w}_{opt}. \quad (3.37)$$

La seconde égalité est due au fait que σ_d^2 est un scalaire réel, qui est donc égal à son transposé conjugué. L'équation $J_{min} = \sigma_d^2 - \sigma_d^2$ devient alors :

$$J_{min} = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_{opt} = \sigma_d^2 - \mathbf{p}^H \mathbf{R}_{uu}^{-1} \mathbf{p}. \quad (3.38)$$

3.3.3 Forme canonique

En faisant la différence (3.34) et (3.38), on a :

$$\begin{aligned} J - J_{min} &= -\mathbf{p}^H \mathbf{w} - \mathbf{w}^H \mathbf{p} + \mathbf{w}^H \mathbf{R}_{uu} \mathbf{w} + \mathbf{p}^H \mathbf{w}_{opt} \\ &= -\mathbf{p}^H (\mathbf{w} - \mathbf{w}_{opt}) - \mathbf{w}^H \mathbf{R}_{uu} (\mathbf{w}_{opt} - \mathbf{w}) \\ &= -\mathbf{w}_{opt}^H \mathbf{R}_{uu} (\mathbf{w} - \mathbf{w}_{opt}) - \mathbf{w}^H \mathbf{R}_{uu} (\mathbf{w}_{opt} - \mathbf{w}). \end{aligned} \quad (3.39)$$

Or, $\mathbf{R}_{uu} = \mathbf{R}_{uu}^H$ en raison de la symétrie hermitienne, donc on a :

$$J = J_{min} + (\mathbf{w} - \mathbf{w}_{opt})^H \mathbf{R}_{uu} (\mathbf{w} - \mathbf{w}_{opt}). \quad (3.40)$$

Cette expression montre l'unicité de la solution optimale, puisque la forme hermitienne (ou quadratique dans le cas de signaux réels) ne s'annule que pour $\mathbf{w} = \mathbf{w}_{opt}$.

Par ailleurs, puisque \mathbf{R}_{uu} est une matrice de corrélation à symétrie hermitienne, on peut la décomposer dans sa base de vecteurs propres et l'écrire sous la forme :

$$\mathbf{R}_{uu} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H, \quad (3.41)$$

où $\mathbf{\Lambda}$ est une matrice diagonale (des valeurs propres) et \mathbf{Q} est la matrice des vecteurs propres. L'équation (3.40) devient alors :

$$J = J_{min} + (\mathbf{w} - \mathbf{w}_{opt})^H \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H (\mathbf{w} - \mathbf{w}_{opt}). \quad (3.42)$$

En posant $\mathbf{v}^H = (\mathbf{w} - \mathbf{w}_{opt})^H \mathbf{Q}$ (ce qui correspond à un changement d'origine et d'échelle), on a :

$$J = J_{min} + \mathbf{v}^H \mathbf{\Lambda} \mathbf{v}. \quad (3.43)$$

Puisque $\mathbf{\Lambda}$ est une matrice diagonale constituée d'éléments λ_i , on a finalement :

$$J = J_{min} + \sum_i \lambda_i v_i v_i^* = J_{min} + \sum_i \lambda_i |v_i|^2. \quad (3.44)$$

Cette dernière expression met en évidence les axes principaux \mathbf{v} .

Chapitre 4

Prédiction linéaire

La prédiction linéaire est un outil très puissant utilisé pour l'analyse de séries temporelles, par exemple de signaux de parole, de consommation électrique, d'indices boursiers. On peut aussi utiliser la prédiction linéaire pour modéliser des signaux, par exemple des signaux de parole, en vue de compresser le signal. L'idée est alors de remplacer la transmission ou le stockage des échantillons par :

- la transmission des paramètres de prédiction (une fois pour toutes),
- la transmission (ou le stockage) à chaque instant n , de l'erreur de prédiction seulement, ce qui peut être codé par un nombre restreint de bits.

Dans ce chapitre, on considèrera (i) la prédiction directe (ou avant, *forward* en anglais) qui consiste, connaissant $u(n-1), u(n-2), \dots$ à prédire $u(n)$ voire $u(n+1), \dots$, et (ii) la prédiction rétrograde (ou arrière, *backward* en anglais) qui consiste, connaissant $u(n), u(n-1), \dots u(n-M+1)$ à prédire $u(n-M)$. . .

Pour simplifier, dans ce chapitre, on considèrera uniquement des signaux réels.

4.1 Prédiction avant (*forward prediction*)

4.1.1 Modèle

On considère un filtre linéaire transverse de taille M , caractérisé par le vecteur des paramètres $\mathbf{w} = (w_1, w_2, \dots, w_M)$ qui pondère les entrées $u(n-1), u(n-2), \dots, u(n-M)$ de façon à fournir une prédiction linéaire notée $\hat{u}(n)$:

$$\hat{u}(n) = \sum_{k=1}^M w_k u(n-k). \quad (4.1)$$

La solution, optimale au sens des moindres carrés, est déterminée et notée \mathbf{w}_{opt} , et on peut écrire :

$$\hat{u}(n/\mathcal{U}_{n-1}) = \sum_{k=1}^M w_{opt,k} u(n-k), \quad (4.2)$$

où \mathcal{U}_{n-1} représente l'espace engendré par toutes les observations passées avant l'instant $n-1$ inclus. Autrement dit, $\hat{u}(n/\mathcal{U}_{n-1})$ représente l'approximation de $u(n)$ obtenue en utilisant toute l'information passée.

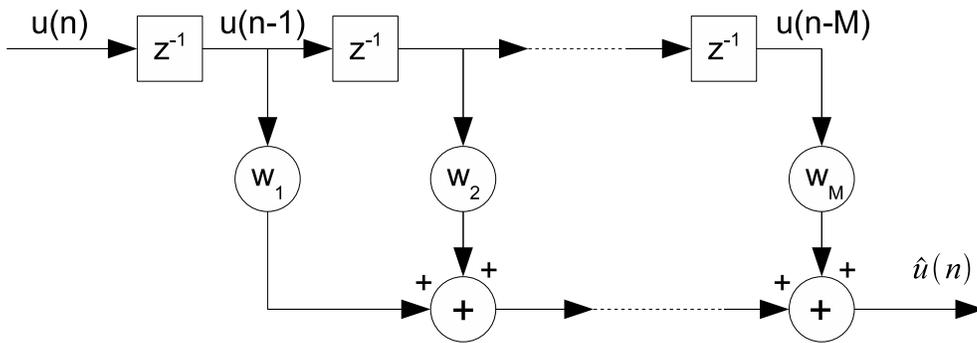


FIGURE 4.1 – Représentation schématique d'un prédicteur à un pas

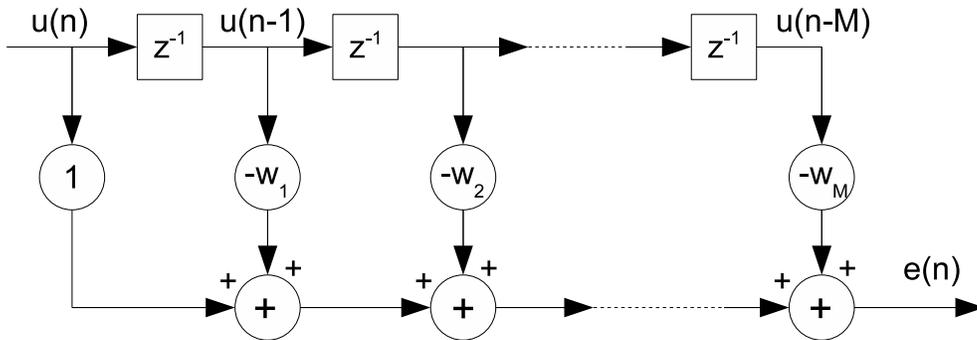


FIGURE 4.2 – Représentation schématique du calcul de l'erreur $e(n)$

On peut représenter schématiquement le filtre prédicteur par le schéma de la figure 4.1.

Le terme d'erreur, $e(n)$, qui intervient dans la minimisation, peut être obtenu par le filtre suivant (figure 4.2) ou de façon plus compacte, en utilisant le filtre prédicteur par le schéma de la figure 4.3.

4.1.2 Erreur de prédiction à un pas

On note l'erreur de prédiction à un pas :

$$f_M(n) = u(n) - \hat{u}(n), \quad (4.3)$$

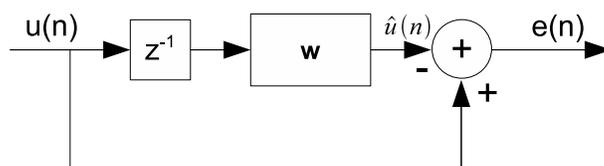


FIGURE 4.3 – Représentation compacte du calcul de l'erreur $e(n)$ à l'aide du filtre prédicteur

où M est l'ordre du prédicteur. A l'optimum, on notera :

$$f_M(n) = u(n) - \hat{u}(n/\mathcal{U}_{n-1}), \quad (4.4)$$

où $\hat{u}(n/\mathcal{U}_{n-1})$ est l'approximation obtenue en tenant compte de tout le passé.

L'erreur quadratique moyenne, mesurant les performances du prédicteur, est donc :

$$P_M = E[f_M^2(n)]. \quad (4.5)$$

Dans le cas de signaux complexes, on aurait $P_M = E[|f_M(n)|^2]$.

Par rapport au filtrage de Wiener, on utilise :

- les entrées (M au total) à partir de $u(n-1)$, c'est-à-dire $u(n-1), \dots, u(n-M)$ et on note $\mathbf{u}(n-1) = (u(n-1), u(n-2), \dots, u(n-M))^T$,
- la sortie désirée est égale à $u(n)$: $d(n) = u(n)$.

On note :

- $\mathbf{R}_{\mathbf{uu}} = E[\mathbf{u}(n-1)\mathbf{u}^T(n-1)]$ la matrice d'auto-corrélation,
- $\mathbf{w}_{opt} = (w_{opt,1}, w_{opt,2}, \dots, w_{opt,M})^T$, le vecteur des paramètres du prédicteur optimal,
- $\mathbf{r} = E[\mathbf{u}(n-1)d(n)] = E[\mathbf{u}(n-1)u(n)]$.

L'équation de Wiener-Hopf du prédicteur à un pas s'écrit alors :

$$\mathbf{R}_{\mathbf{uu}} \mathbf{w}_{opt} = \mathbf{r}, \quad (4.6)$$

et, en utilisant la relation $J_{min} = \sigma_d^2 - \sigma_{\hat{d}}^2$ et (3.37) du chapitre 3, l'erreur de prédiction minimale vaut :

$$(P_M)_{min} = \sigma_d^2 - \sigma_{\hat{d}}^2 = R_{uu}(0) - \mathbf{r}^T \mathbf{w}_{opt}. \quad (4.7)$$

4.1.3 Comparaison entre le filtre de Wiener et le filtre prédicteur

La comparaison des deux filtres est donnée dans le tableau ci-dessous.

Grandeur	filtre de Wiener (dim)	filtre prédicteur (dim)
vecteur d'entrée	$\mathbf{u}(n)$ ($M+1$)	$\mathbf{u}(n-1)$ (M)
sortie désirée	$d(n)$ (1)	$u(n)$ (1)
coef. du filtre	\mathbf{w}_{opt} ($M+1$)	\mathbf{w}_{opt} (M)
erreur d'estimation	$e(n)$ (1)	$f_M(n)$ (1)
matrice d'auto-corr.	$\mathbf{R}_{\mathbf{uu}} = E[\mathbf{u}(n)\mathbf{u}^T(n)]$	$\mathbf{R}_{\mathbf{uu}} = E[\mathbf{u}(n-1)\mathbf{u}^T(n-1)]$
dim. de la matrice	$(M+1) \times (M+1)$	$(M) \times (M)$
vecteur d'intercor.	$\mathbf{p} = E[\mathbf{u}(n)d(n)]$ ($M+1$)	$\mathbf{r} = E[\mathbf{u}(n-1)u(n)]$ (M)
min. de l'erreur quad.	J_{min}	$(P_M)_{min}$
équation de Wiener-Hopf	$\mathbf{R}_{\mathbf{uu}} \mathbf{w}_{opt} = \mathbf{p}$	$\mathbf{R}_{\mathbf{uu}} \mathbf{w}_{opt} = \mathbf{r}$

TABLE 4.1 – Comparaison entre les différentes grandeurs intervenant dans le filtre de Wiener et le filtre prédicteur.

Il est aussi intéressant de comparer les équations de Wiener-Hopf et celles de Yule-Walker caractérisant un filtre auto-régressif. Si le processus est un AR d'ordre M connu, les coefficients du prédicteur linéaire optimal au sens des moindres carrés doivent prendre exactement les mêmes valeurs que les coefficients du processus AR.

Grandeur	Wiener-Hopf	Yule-Walker
équations	$\mathbf{R}_{uu}\mathbf{w}_{opt} = \mathbf{p}$	$\mathbf{R}_{uu}\mathbf{w} = \mathbf{r}$
min. de l'erreur quad.	$P_M = R_{uu}(0) - \mathbf{r}^T \mathbf{w}_{opt}$	$\sigma_v^2 = \sum_{k=0}^M a_k R_{uu}(k)$

TABLE 4.2 – Comparaison entre les équations de Wiener-Hopf et celles de Yule-Walker.

4.2 Prédiction arrière ou rétrograde à un pas

4.2.1 Le problème

Le problème consiste à prédire $u(n-M)$ à partir des M échantillons $u(n), u(n-1), \dots, u(n-M+1)$. Notons \mathcal{U}_n l'espace engendré par $u(n), u(n-1), \dots, u(n-M+1)$, la prédiction arrière est donc :

$$\hat{u}(n-M/\mathcal{U}_n) = \sum_{k=1}^M g_k u(n-k+1), \quad (4.8)$$

où les $g_k, k = 1, \dots, M$ sont les coefficients du prédicteur, que nous estimerons au sens des moindres carrés.

Pour ce problème de prédiction arrière à un pas, la valeur désirée est donc :

$$d(n) = u(n-M), \quad (4.9)$$

et l'erreur de prédiction arrière est :

$$b_M(n) = u(n-M) - \hat{u}(n-M/\mathcal{U}_n). \quad (4.10)$$

L'erreur quadratique qui sera le critère à minimiser s'écrit donc :

$$P_M = E[b_M^2(n)]. \quad (4.11)$$

4.2.2 Optimisation

On note $\mathbf{g} = (g_1, g_2, \dots, g_M)^T$ et $\mathbf{u}(n) = (u(n), u(n-1), \dots, u(n-M+1))^T$, d'où l'on tire $b_M(n) = u(n-M) - \mathbf{g}^T \mathbf{u}(n)$. En appliquant le théorème de la projection, la solution qui minimise $E[b_M^2(n)]$ correspond donc à l'ensemble d'équations :

$$E[b_M(n)u(n-k+1)] = 0, \quad k = 1, \dots, M. \quad (4.12)$$

Or, en développant $b_M(n)$, on a pour tout $k = 1, \dots, M$:

$$\begin{aligned} E[(u(n-M) - \mathbf{g}^T \mathbf{u}(n))u(n-k+1)] &= 0, \\ \sum_{i=1}^M g_i E[u(n-i+1)u(n-k+1)] &= E[(u(n-M)u(n-k+1))]. \end{aligned} \quad (4.13)$$

On peut écrire ces M équations sous la forme matricielle suivante :

$$\begin{pmatrix} E[u^2(n)] & E[u(n-1)u(n)] & \dots & E[u(n-M+1)u(n)] \\ E[u(n)u(n-1)] & E[u^2(n-1)] & & \vdots \\ \vdots & & \ddots & \\ E[u(n)u(n-M+1)] & \dots & & E[u^2(n-M+1)] \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_M \end{pmatrix} = \begin{pmatrix} E[u(n-M)u(n)] \\ E[u(n-M)u(n-1)] \\ \vdots \\ E[u(n-M)u(n-M+1)] \end{pmatrix} \quad (4.14)$$

On peut mettre sous la forme compacte :

$$E[\mathbf{u}(n)\mathbf{u}^T(n)]\mathbf{g} = E[\mathbf{u}(n)u(n-M)], \quad (4.15)$$

que l'on peut encore écrire :

$$\mathbf{R}_{uu}\mathbf{g} = \mathbf{r}^B, \quad (4.16)$$

où $\mathbf{r}^B = (R_{uu}(M), R_{uu}(M-1), \dots, R_{uu}(1))^T$. Cette équation est donc de forme tout à fait similaire à celle du prédicteur avant à un pas. Cependant, les vecteurs mis en jeu sont définis de façon différente.

4.2.3 Relation entre prédiction avant et arrière

Pour la prédiction avant, on a :

$$\begin{aligned} E[\mathbf{u}(n-1)\mathbf{u}^T(n-1)]\mathbf{w} &= E[\mathbf{u}(n-1)u(n)] \\ \text{avec } \mathbf{u}(n-1) &= (u(n-1) \dots u(n-M))^T \end{aligned} \quad (4.17)$$

alors que pour la prédiction arrière, on a :

$$\begin{aligned} E[\mathbf{u}(n)\mathbf{u}^T(n)]\mathbf{g} &= E[\mathbf{u}(n)u(n-M)] \\ \text{avec } \mathbf{u}(n) &= (u(n) \dots u(n-M+1))^T. \end{aligned} \quad (4.18)$$

Pour des processus stationnaires, les termes de gauche sont identiques, et les équations diffèrent par le vecteur du second membre. En explicitant, on a

— pour la prédiction avant :

$$\mathbf{r} = \begin{pmatrix} E[u(n-1)u(n)] = R_{uu}(1) \\ E[u(n-2)u(n)] = R_{uu}(2) \\ \vdots \\ E[u(n-M)u(n)] = R_{uu}(M) \end{pmatrix} \quad (4.19)$$

— pour la prédiction arrière :

$$\mathbf{r}^B = \begin{pmatrix} E[u(n-M)u(n)] = R_{uu}(M) \\ E[u(n-M)u(n-1)] = R_{uu}(M-1) \\ \vdots \\ E[u(n-M)u(n-M+1)] = R_{uu}(1) \end{pmatrix} \quad (4.20)$$

On remarque que l'ordre des composantes est simplement inversé. Ainsi, en ordonnant en sens inverse les coefficients du vecteur \mathbf{g} , c'est-à-dire en introduisant la notation :

$$\mathbf{g}^B = (g_M, g_{M-1} \dots g_1)^T, \quad (4.21)$$

l'équation du prédicteur arrière devient :

$$\mathbf{R}_{uu}^T \mathbf{g}^B = \mathbf{r}. \quad (4.22)$$

Or, dans le cas réel que l'on considère seul ici, la matrice \mathbf{R}_{uu} est symétrique, c'est-à-dire $\mathbf{R}_{uu} = \mathbf{R}_{uu}^T$, et l'équation précédente s'écrit simplement :

$$\mathbf{R}_{uu} \mathbf{g}^B = \mathbf{r}, \quad (4.23)$$

qui est maintenant exactement de la forme de l'équation du prédicteur avant :

$$\mathbf{R}_{uu} \mathbf{w} = \mathbf{r}. \quad (4.24)$$

Il est donc évident que les deux solutions sont identiques, c'est-à-dire que :

$$\mathbf{g}^B = \mathbf{w}_{opt}. \quad (4.25)$$

Les coefficients du filtre prédicteur arrière sont donc exactement les mêmes que ceux du filtre prédicteur avant, mais rangés en sens inverse.

4.2.4 Erreur de prédiction

On peut donc appliquer les résultats du prédicteur avant avec ces dernières notations. Pour le prédicteur arrière, on en déduit l'erreur minimale, à l'optimum :

$$P_M = R_{uu}(0) - \mathbf{r}^T \mathbf{g}^B = R_{uu}(0) - \mathbf{r}^T \mathbf{w}_{opt}. \quad (4.26)$$

On voit ainsi que les erreurs optimales des deux prédicteurs ont exactement les mêmes expressions.

Chapitre 5

Algorithmes adaptatifs

5.1 Introduction

5.1.1 Objectifs

La mise en oeuvre d'un filtre de Wiener demande (i) le calcul des paramètres statistiques : \mathbf{R}_{uu} et $\mathbf{R}_{ud} = \mathbf{p}$, puis (ii) le calcul de la solution : $\mathbf{w}_{opt} = \mathbf{R}_{uu}^{-1}\mathbf{p}$.

En pratique, cette méthode demande donc l'estimation d'une matrice de dimension $M \times M$ et d'un vecteur de dimension M , qui sont réalisés par des moyennes empiriques sur k échantillons, sous hypothèse d'ergodicité :

$$E[u(n)u(n-j)] = \frac{1}{k} \sum_{p=j}^{j+k-1} u(p)u(p-j). \quad (5.1)$$

Une fois l'estimation des coefficients la matrice \mathbf{R}_{uu} effectuée, la solution requiert le calcul de l'inverse de cette matrice.

D'un point de vue complexité, on peut imaginer des approximations plus simples, à la fois pour l'estimation des paramètres et pour le calcul de la solution. Le problème de ces algorithmes simples sera alors d'étudier leur comportement vers la solution optimale : convergence et erreur quadratique.

Dans ce chapitre, comme mise en oeuvre de la solution des équations de Wiener-Hopt, on étudiera un algorithme de descente du gradient, et une approximation très simple : l'algorithme LMS (*least mean square*). Puis nous étudierons l'implantation d'un algorithme des moindres carrés sous sa forme récursive : l'algorithme RLS (*recursive least square*).

5.1.2 Dérivation par rapport à un vecteur

Avant de rentrer dans le vif du sujet, nous introduisons dans ce paragraphe les outils qui nous permettront de concevoir ces algorithmes.

Soient deux vecteurs \mathbf{u} et \mathbf{v} et une matrice \mathbf{R} . On s'intéresse au gradient des quantités scalaires $\mathbf{u}^T \mathbf{v}$ et $\mathbf{u}^T \mathbf{R} \mathbf{v}$ par rapport aux vecteurs \mathbf{u} et \mathbf{v} . Nous nous restreignons encore au cas réel, par conséquent, on peut remarquer que :

$$\mathbf{u}^T \mathbf{v} = \mathbf{v}^T \mathbf{u} \quad (5.2)$$

et

$$\mathbf{u}^T \mathbf{R} \mathbf{v} = \mathbf{v}^T \mathbf{R}^T \mathbf{u}. \quad (5.3)$$

Gradient de $\mathbf{u}^T \mathbf{v}$ par rapport à \mathbf{u} ou à \mathbf{v} .

Calculons le vecteur gradient $\partial \mathbf{u}^T \mathbf{v} / \partial \mathbf{u}$. Pour cela, calculons la composante k de ce vecteur :

$$\frac{\partial \mathbf{u}^T \mathbf{v}}{\partial u_k} = \frac{\partial}{\partial u_k} \sum_k u_k v_k = v_k. \quad (5.4)$$

On en déduit :

$$\frac{\partial \mathbf{u}^T \mathbf{v}}{\partial \mathbf{u}} = \frac{\partial \mathbf{v}^T \mathbf{u}}{\partial \mathbf{u}} = \mathbf{v}, \quad (5.5)$$

et de façon similaire :

$$\frac{\partial \mathbf{u}^T \mathbf{v}}{\partial \mathbf{v}} = \frac{\partial \mathbf{v}^T \mathbf{u}}{\partial \mathbf{v}} = \mathbf{u}. \quad (5.6)$$

Gradient de $\mathbf{u}^T \mathbf{R} \mathbf{v}$ par rapport à \mathbf{u} ou à \mathbf{v} .

Calculons le vecteur gradient $\partial \mathbf{u}^T \mathbf{R} \mathbf{v} / \partial \mathbf{u}$. Pour cela, calculons la composante k de ce vecteur, en posant $\mathbf{R} \mathbf{v} = \mathbf{v}'$ et en appliquant les résultats précédents :

$$\frac{\partial \mathbf{u}^T \mathbf{R} \mathbf{v}}{\partial u_k} = \frac{\partial \mathbf{u}^T \mathbf{v}'}{\partial u_k} = \frac{\partial}{\partial u_k} \sum_k u_k v'_k = v'_k. \quad (5.7)$$

D'où la relation :

$$\frac{\partial \mathbf{u}^T \mathbf{R} \mathbf{v}}{\partial \mathbf{u}} = \mathbf{R} \mathbf{v}. \quad (5.8)$$

De façon similaire, on peut calculer le gradient de cette quantité par rapport à \mathbf{v} . La composante k vaut :

$$\frac{\partial \mathbf{u}^T \mathbf{R} \mathbf{v}}{\partial v_k} = \frac{\partial \mathbf{v}^T \mathbf{R}^T \mathbf{u}}{\partial v_k} \quad (5.9)$$

D'où, en utilisant le résultat précédent, la relation :

$$\frac{\partial \mathbf{u}^T \mathbf{R} \mathbf{v}}{\partial \mathbf{v}} = \mathbf{R}^T \mathbf{u}. \quad (5.10)$$

Enfin, considérons le gradient de la forme quadratique $\mathbf{u}^T \mathbf{R} \mathbf{u}$ par rapport à \mathbf{u} . Ici, l'expression est un produit de la variable de dérivation, et par conséquent le gradient est la somme :

$$\frac{\partial \mathbf{u}^T \mathbf{R} \mathbf{u}}{\partial \mathbf{u}} = \frac{\partial \mathbf{u}^T (\mathbf{R} \mathbf{u})}{\partial \mathbf{u}} + \frac{\partial (\mathbf{u}^T \mathbf{R}) \mathbf{u}}{\partial \mathbf{u}}, \quad (5.11)$$

soit en utilisant les résultats précédents :

$$\frac{\partial \mathbf{u}^T \mathbf{R} \mathbf{u}}{\partial \mathbf{u}} = \mathbf{R} \mathbf{u} + \mathbf{R}^T \mathbf{u}. \quad (5.12)$$

Dans le cas courant où la matrice \mathbf{R} est symétrique (par exemple, si \mathbf{R} est une matrice d'auto-corrélation), on a simplement :

$$\frac{\partial \mathbf{u}^T \mathbf{R} \mathbf{u}}{\partial \mathbf{u}} = 2\mathbf{R} \mathbf{u}. \quad (5.13)$$

5.2 Algorithme de descente ou *steepest descent algorithm*

5.2.1 Rappel sur les notations

On cherche à calculer les paramètres du filtre linéaire optimal. A l'instant n , ces paramètres sont notés $\mathbf{w}(n) = (w_0(n), w_1(n), \dots, w_{M-1}(n))^T$, et les entrées du filtre forment le vecteur $\mathbf{u}(n) = (u(n), u(n-1), \dots, u(n-M+1))^T$. A chaque instant n , la sortie du filtre est un signal $y(n)$ que l'on souhaite aussi proche que possible d'un signal désiré $d(n)$.

On rappelle que la solution optimale, au sens des moindres carrés, est donnée par la solution de l'équation de Wiener-Hopf :

$$\mathbf{R}_{\mathbf{u}\mathbf{u}}\mathbf{w} = \mathbf{p}. \quad (5.14)$$

L'erreur quadratique moyenne (le critère à minimiser), au pas n , s'écrit :

$$J(n) = \sigma_d^2 - \mathbf{w}^H(n)\mathbf{p} - \mathbf{p}^H\mathbf{w}(n) + \mathbf{w}^H(n)\mathbf{R}_{\mathbf{u}\mathbf{u}}\mathbf{w}(n). \quad (5.15)$$

A l'optimum, on note $\mathbf{w} = \mathbf{w}_{opt}$ et on a les égalités :

$$\mathbf{R}_{\mathbf{u}\mathbf{u}}\mathbf{w}_{opt} = \mathbf{p}, \quad (5.16)$$

et

$$J(n) = \sigma_d^2 - \mathbf{p}^H\mathbf{w}_{opt}(n). \quad (5.17)$$

5.2.2 Algorithme de descente

Pour éviter la résolution directe de l'équation de Wiener-Hopf, dont la solution analytique exacte est coûteuse (elle requiert le calcul de l'inverse d'une matrice de dimension M), on propose une implémentation algorithmique par ajustements successifs à chaque nouvelle observation : c'est le principe des algorithmes adaptatifs.

L'algorithme procède en trois étapes, une étape d'initialisation et deux étapes qui sont itérées :

1. Initialisation : au temps $n = 0$, on attribue une valeur initiale $\mathbf{w}(0)$; typiquement, on prend $\mathbf{w}(0) = \mathbf{0}$;
2. A chaque instant n (l'instant courant), on calcule le gradient de $J(n)$ par rapport au vecteur $\mathbf{w}(n)$;
3. On ajuste les paramètres du filtre selon la formule :

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu\nabla_{\mathbf{w}(n)}J(n). \quad (5.18)$$

L'idée de cet algorithme est que les ajustements en sens opposé au gradient (descente du gradient) vont conduire $\mathbf{w}(n)$ vers la valeur \mathbf{w}_{opt} qui est associée à l'erreur minimale J_{min} . Dans l'équation (5.18), le paramètre $\mu/2$ est un petit scalaire positif qui permet d'ajuster la vitesse et la précision des ajustements. Cette idée est illustrée géométriquement à la figure 5.1.

A partir de l'équation (5.15) et des règles de calcul rappelées dans l'introduction de ce chapitre (en tenant compte de la symétrie de $\mathbf{R}_{\mathbf{u}\mathbf{u}}$), on peut facilement calculer le gradient de l'erreur $J(n)$:

$$\nabla_{\mathbf{w}(n)}J(n) = -2\mathbf{p} + 2\mathbf{R}_{\mathbf{u}\mathbf{u}}\mathbf{w}(n), \quad (5.19)$$

si bien que, en reportant dans (5.18), on arrive à la formule d'ajustement du filtre (en rappelant $\mathbf{p} = \mathbf{R}_{\mathbf{u}\mathbf{d}}$) :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{R}_{\mathbf{u}\mathbf{d}} - \mathbf{R}_{\mathbf{u}\mathbf{u}}\mathbf{w}(n)], \quad (5.20)$$

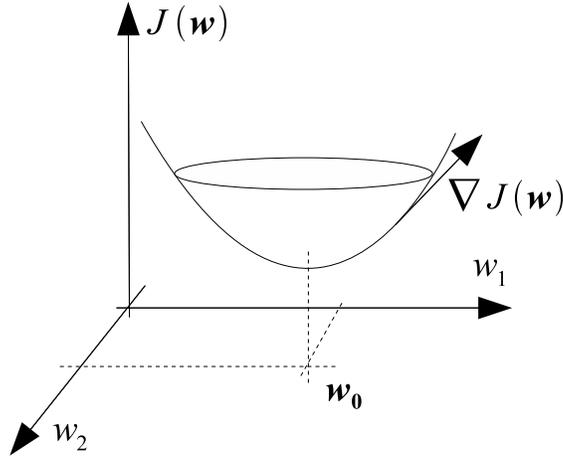


FIGURE 5.1 – Représentation géométrique du principe de la méthode de descente du gradient

où μ est le pas (*stepsize*) du gradient.

On remarque que la correction à l'instant n , $\delta \mathbf{w}(n) = \mu[\mathbf{R}_{ud} - \mathbf{R}_{uu}\mathbf{w}(n)]$, peut s'écrire :

$$\begin{aligned}
 \delta \mathbf{w}(n) &= \mu E[\mathbf{u}(n)d(n) - \mathbf{u}(n)\mathbf{u}^T(n)\mathbf{w}(n)] \\
 &= \mu E[\mathbf{u}(n)(d(n) - \mathbf{u}^T(n)\mathbf{w}(n))] \\
 &= \mu E[\mathbf{u}(n)(d(n) - y(n))] \\
 &= \mu E[\mathbf{u}(n)e(n)].
 \end{aligned} \tag{5.21}$$

On remarque qu'à la convergence, c'est-à-dire lorsque l'ajustement est nul ($\delta \mathbf{w}(n) = 0$), on a $E[\mathbf{u}(n)e(n)] = 0$, ce qui signifie que $e(n)$ est orthogonal à $\mathbf{u}(n-k)$, $\forall k \in \{0, 1, \dots, M-1\}$. On retrouve donc la propriété de la solution optimale des moindres carrés, qui s'énonce par le théorème de la projection orthogonale.

5.2.3 Stabilité de l'algorithme

On conçoit facilement que si le pas μ est trop grand, l'algorithme peut osciller autour de la solution optimale \mathbf{w}_{opt} . On se propose donc d'étudier les conditions de stabilité de l'algorithme, au sens où l'erreur paramétrique

$$\mathbf{c}(n) = \mathbf{w}(n) - \mathbf{w}_{opt} \tag{5.22}$$

est décroissante. Sachant que l'on a l'égalité $\mathbf{R}_{uu}\mathbf{w}_{opt} = \mathbf{R}_{ud}$ et à partir de l'équation de mise à jour des paramètres (5.20), on peut écrire :

$$\begin{aligned}
 \mathbf{c}(n+1) &= \mathbf{w}(n+1) - \mathbf{w}_{opt} \\
 &= \mathbf{w}(n) + \mu[\mathbf{R}_{ud} - \mathbf{R}_{uu}\mathbf{w}(n)] - \mathbf{w}_{opt} \\
 &= \mathbf{c}(n) + \mu[\mathbf{R}_{uu}\mathbf{w}_{opt} - \mathbf{R}_{uu}\mathbf{w}(n)] \\
 &= \mathbf{c}(n) + \mu\mathbf{R}_{uu}[\mathbf{w}_{opt} - \mathbf{w}(n)] \\
 &= (\mathbf{I} - \mu\mathbf{R}_{uu})\mathbf{c}(n).
 \end{aligned} \tag{5.23}$$

La matrice d'auto-corrélation (symétrique dans le cas réel, hermitienne dans le cas complexe) peut être décomposée sur la base des vecteurs propres :

$$\mathbf{R}_{uu} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \tag{5.24}$$

où \mathbf{Q} est une matrice orthogonale (unitaire dans le cas complexe) et $\mathbf{\Lambda}$ est une matrice diagonale dont les éléments diagonaux sont les valeurs propres de $\mathbf{R}_{\mathbf{uu}}$. Notons que, dans le cas complexe, il suffit de remplacer T par H dans l'équation précédente. En utilisant cette décomposition dans la dernière équation de (5.23), et en tenant compte de l'orthogonalité de \mathbf{Q} , c'est-à-dire que $\mathbf{Q}^{-1} = \mathbf{Q}^T$, on a :

$$\begin{aligned} \mathbf{c}(n+1) &= (\mathbf{I} - \mu \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T) \mathbf{c}(n) \\ \mathbf{Q}^T \mathbf{c}(n+1) &= \mathbf{Q}^T (\mathbf{I} - \mu \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T) \mathbf{c}(n) \\ \mathbf{Q}^T \mathbf{c}(n+1) &= \mathbf{I} \mathbf{Q}^T \mathbf{c}(n) - \mu \mathbf{\Lambda} \mathbf{Q}^T \mathbf{c}(n) \\ \mathbf{Q}^T \mathbf{c}(n+1) &= (\mathbf{I} - \mu \mathbf{\Lambda}) \mathbf{Q}^T \mathbf{c}(n). \end{aligned} \quad (5.25)$$

En notant $\mathbf{v}(n) = \mathbf{Q}^T \mathbf{c}(n)$, l'erreur paramétrique dans le nouvel espace de représentation associé à la transformation linéaire \mathbf{Q} , on a simplement :

$$\mathbf{v}(n+1) = (\mathbf{I} - \mu \mathbf{\Lambda}) \mathbf{v}(n). \quad (5.26)$$

Pour le terme d'indice k du vecteur d'erreur (donc lié à l'erreur sur w_k), on peut écrire :

$$\begin{aligned} v_k(n+1) &= (1 - \mu \lambda_k) v_k(n) \\ &= (1 - \mu \lambda_k)^{n+1} v_k(0). \end{aligned} \quad (5.27)$$

Les valeurs propres λ_k de $\mathbf{R}_{\mathbf{uu}}$ étant toutes positives (car $\mathbf{R}_{\mathbf{uu}}$ est définie positive), on voit que $v_k(n+1)$ est une série géométrique de raison $(1 - \mu \lambda_k)$, qui sera convergente si :

$$-1 < 1 - \mu \lambda_k < +1, \quad \forall k \quad (5.28)$$

c'est-à-dire si :

$$0 < \mu < \frac{2}{\lambda_k}, \quad \forall k. \quad (5.29)$$

Pour une convergence globale, c'est-à-dire pour la convergence de tous les paramètres w_k , μ doit donc satisfaire la condition :

$$0 < \mu < \frac{2}{\lambda_{max}}. \quad (5.30)$$

en notant $\lambda_{max} = \max_k \lambda_k$

5.2.4 Vitesse de convergence

On peut illustrer l'évolution des $v_k(n)$ dans l'équation (5.27) par le graphe de la figure 5.2. Compte tenu de la progression géométrique, on peut écrire que l'enveloppe des $v_k(n)$ suit une loi exponentielle décroissante de la forme :

$$v_k(t) = v_k(0) \exp(-t/\tau_k). \quad (5.31)$$

De façon évidente, on a donc :

$$\begin{aligned} v_k(0) &= v_k(0) \\ v_k(1) &= v_k(0) \exp(-1/\tau_k), \end{aligned} \quad (5.32)$$

d'où l'on déduit :

$$(1 - \mu \lambda_k) = \exp(-1/\tau_k). \quad (5.33)$$

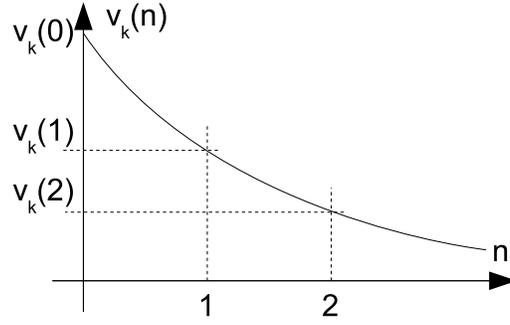


FIGURE 5.2 – Vitesse de convergence de l’algorithme de descente du gradient

Après des calculs algébriques élémentaires, on en déduit donc :

$$\tau_k = \frac{-1}{\ln(1 - \mu\lambda_k)} \approx \frac{+1}{\mu\lambda_k} \quad (5.34)$$

où la dernière approximation, obtenue par un développement au premier ordre en $\mu\lambda_k$, est valide si $\mu\lambda_k \ll 1$.

La constante de temps $\tau_k, \forall k$, doit donc vérifier :

$$\frac{-1}{\ln(1 - \mu\lambda_{max})} \leq \tau_k \leq \frac{-1}{\ln(1 - \mu\lambda_{min})}, \quad (5.35)$$

puisque la fonction \ln est une fonction croissante. Globalement, pour le vecteur \mathbf{w} , c’est-à-dire pour tous les indices $k = 0, \dots, M - 1$, la constante de temps τ vérifie :

$$\frac{-1}{\ln(1 - \mu\lambda_{max})} \leq \tau \leq \frac{-1}{\ln(1 - \mu\lambda_{min})}. \quad (5.36)$$

5.2.5 Erreur quadratique moyenne

L’erreur quadratique moyenne s’exprime (voir chapitre 3, paragraphe 3.3.3) par l’équation (3.44), écrite au pas n :

$$J(n) = J_{min} + \sum_{k=1}^M \lambda_k |v_k(n)|^2. \quad (5.37)$$

On peut encore écrire, à partir de l’équation de récurrence sur $v_k(n + 1)$ (5.27) :

$$J(n) = J_{min} + \sum_{k=1}^M \lambda_k (1 - \mu\lambda_k)^{2n} |v_k(0)|^2. \quad (5.38)$$

Quand μ est choisi de façon à ce que l’erreur paramétrique $\mathbf{c}(n) = \mathbf{w}(n) - \mathbf{w}_{opt}$ diminue, la condition de convergence sur la série géométrique : $|(1 - \mu\lambda_k)| < 1$ entraîne la convergence de la somme dans l’équation (5.38), car les termes $(1 - \mu\lambda_k)^{2n} \rightarrow 0$ et $J(n) \rightarrow J_{min}$ lorsque $n \rightarrow \infty$.

La vitesse de convergence de $J(n)$ vers l’optimum J_{min} est alors caractérisée par la constante de temps :

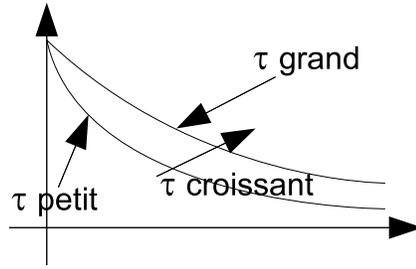


FIGURE 5.3 – Influence de la constante de temps τ sur la vitesse de convergence

$$(\tau_k)_{EQM} = \frac{-1}{2 \ln(1 - \mu \lambda_k)} \approx \frac{1}{2\mu \lambda_k}, \quad (5.39)$$

en raison de l'exposant $2n$ (dans $J(n)$) au lieu de n (dans $v_k(n)$). Notons encore que la dernière approximation est valide si $\mu \lambda_k \ll 1$.

On en déduit donc que (figure 5.3) :

1. plus μ est petit, plus la vitesse de convergence est lente,
2. le paramètre w_k converge d'autant plus vite que λ_k est grande (car alors la constante de temps τ_k est petite et la convergence plus rapide).

5.3 Algorithme du gradient stochastique

La méthode de descente, exposée au paragraphe précédent, a permis de trouver un algorithme qui estime, par itérations successives, une solution qui tend vers la solution optimale de l'équation de Wiener-Hopf, sans inverser la matrice \mathbf{R}_{uu} .

A chaque pas de calcul, cette méthode demande cependant le calcul des M termes (compte tenu des symétries et de la structure Toeplitz) de \mathbf{R}_{uu} et des M termes de \mathbf{R}_{ud} . Chacun de ces $2M$ termes est une espérance estimée par une moyenne empirique, sous hypothèse d'ergodicité des signaux.

Dans ce paragraphe, nous recherchons s'il est possible de simplifier l'algorithme, en évitant d'effectuer le calcul des moyennes à chaque pas.

5.3.1 Conception de l'algorithme du gradient stochastique

Cet algorithme adaptatif est connu sous le nom de Least-Mean-Square (LMS) en anglais. La mise en oeuvre de cet algorithme repose sur la simplification du gradient de $J(n)$, $\nabla J(n) = -2\mathbf{R}_{ud} + 2\mathbf{R}_{uu}\mathbf{w}(n)$, obtenue en remplaçant les estimations des coefficients de \mathbf{R}_{uu} et \mathbf{R}_{ud} fait à l'aide de moyennes empiriques par une estimation instantanée :

$$\begin{aligned} \hat{\mathbf{R}}_{ud} &= d(n)\mathbf{u}(n) \\ \hat{\mathbf{R}}_{uu} &= \mathbf{u}(n)\mathbf{u}^T(n). \end{aligned} \quad (5.40)$$

L'approximation du gradient est alors :

$$\hat{\nabla} J(n) = -2d(n)\mathbf{u}(n) + 2\mathbf{u}(n)\mathbf{u}^T(n)\mathbf{w}(n). \quad (5.41)$$



FIGURE 5.4 – Bernard Widrow, inventeur avec Hoff de l’algorithme LMS

En général, cette estimation simple est biaisée : on remarque en fait qu’il s’agit du gradient de l’erreur quadratique instantanée, et non plus du gradient de l’erreur quadratique moyenne. En effet, l’erreur quadratique instantanée, à l’instant n , s’écrit :

$$e^2(n) = (d(n) - y(n))^2 = (d(n) - \mathbf{w}^T(n)\mathbf{u}(n))^2. \quad (5.42)$$

Calculons maintenant le gradient de $e^2(n)$ par rapport à $\mathbf{w}(n)$:

$$\begin{aligned} \frac{\partial e^2(n)}{\partial \mathbf{w}(n)} &= -2[d(n) - \mathbf{w}^T(n)\mathbf{u}(n)]\mathbf{u}(n) \\ &= -2d(n)\mathbf{u}(n) + 2[\mathbf{w}^T(n)\mathbf{u}(n)]\mathbf{u}(n) \\ &= -2d(n)\mathbf{u}(n) + 2[\mathbf{u}(n)\mathbf{u}^T(n)]\mathbf{w}(n). \end{aligned} \quad (5.43)$$

Bien sûr, minimiser l’erreur quadratique instantanée est moins efficace et moins précis que corriger l’erreur quadratique moyenne. La règle d’ajustement des paramètres devient alors :

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)(d(n) - \mathbf{u}^T(n)\hat{\mathbf{w}}(n)). \quad (5.44)$$

En fait, en exprimant l’erreur $e(n) = d(n) - \mathbf{u}^T(n)\hat{\mathbf{w}}(n)$, la formule devient simplement :

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e(n). \quad (5.45)$$

Cet algorithme très simple a été proposé par Widrow et Hoff en 1960. D’un point de vue de la complexité, chaque itération du LMS demande simplement $2M + 1$ multiplications et $2M + 1$ additions.

5.3.2 Analyse de la stabilité du LMS

Notations et hypothèses

L’algorithme du gradient stochastique (ou LMS) est associé à un système bouclé multivariable, non-linéaire et stochastique. En effet, le système est :

- multivariable en raison des M paramètres à estimer,
- non-linéaire, car la boucle (l’équation de mise à jour) dépend du vecteur d’entrée,
- stochastique, car les itérations successives dépendent de la suite inconnue des observations $u(n)$ et de l’état $\mathbf{w}(n)$ (donc l’évolution, si elle n’est pas déterministe est stochastique).

Pour ces raisons, l’étude de la convergence de l’algorithme est délicate. Compte tenu de la nature stochastique, on ne peut en particulier considérer que la convergence en moyenne quadratique. On rappelle les deux types de convergence :

- convergence en moyenne : $E[\hat{\mathbf{w}}(n) - \mathbf{w}_{opt}] \rightarrow 0$ si $n \rightarrow \infty$,

— convergence en moyenne quadratique : $\lim_{n \rightarrow \infty} J(n) = J_\infty < +\infty$.

Par ailleurs, le calcul n'est possible qu'en introduisant quatre hypothèses simplificatrices supplémentaires :

1. H1 : les vecteurs d'entrée, $\mathbf{u}(1), \dots, \mathbf{u}(n)$ forment une suite de vecteurs statistiquement indépendants ;
2. H2 : à chaque instant n , $\mathbf{u}(n)$ est statistiquement indépendant de toutes les réponses désirées passées $d(1), \dots, d(n-1)$;
3. H3 : à chaque instant n , $d(n)$ est dépendante de $u(n)$ mais indépendante des sorties désirées passées $d(1), \dots, d(n-1)$;
4. H4 : $\mathbf{u}(n)$ et $d(n)$ sont supposés être gaussiens, pour tout n .

A partir de ces hypothèses, on déduit que :

— $\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n)[d(n) - \mathbf{u}^T(n)\hat{\mathbf{w}}(n)]$ est indépendant de $\mathbf{u}(n+1)$ (H1) et de $d(n+1)$ (H2) ;

— pour $k = 1, \dots, n-1$, on a $E[\mathbf{u}(n)\mathbf{u}^T(k)] = 0$ (H1) ;

— pour $k = 1, \dots, n-1$, on a $E[\mathbf{u}(n)d(k)] = 0$ (H2).

Notons l'erreur paramétrique de l'estimateur LMS à l'étape n :

$$\epsilon(n) = \hat{\mathbf{w}}(n) - \mathbf{w}_{opt}, \quad (5.46)$$

pour étudier la convergence en moyenne et en moyenne quadratique, on calcule les espérances $E[\epsilon(n)]$ et $E[\epsilon(n)\epsilon^T(n)]$.

Convergence en moyenne

A partir de l'équation de mise à jour $\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n)[d(n) - \mathbf{u}^T(n)\hat{\mathbf{w}}(n)]$ et de la définition de l'erreur LMS (5.46), en soustrayant \mathbf{w}_{opt} de l'équation de mise à jour, on obtient :

$$\begin{aligned} \hat{\mathbf{w}}(n+1) - \mathbf{w}_{opt} &= \hat{\mathbf{w}}(n) - \mathbf{w}_{opt} + \mu \mathbf{u}(n)[d(n) - \mathbf{u}^T(n)(\epsilon(n) + \mathbf{w}_{opt})] \\ \epsilon(n+1) &= \epsilon(n) - \mu \mathbf{u}(n)\mathbf{u}^T(n)\epsilon(n) + \mu \mathbf{u}(n)[d(n) - \mathbf{u}^T(n)\mathbf{w}_{opt}] \\ \epsilon(n+1) &= (\mathbf{I} - \mu \mathbf{u}(n)\mathbf{u}^T(n))\epsilon(n) + \mu \mathbf{u}(n)e_{opt}(n), \end{aligned} \quad (5.47)$$

où $e_{opt}(n)$ est l'erreur optimale à l'instant n , c'est-à-dire l'erreur obtenue si les paramètres estimés sont optimaux, soit : $\hat{\mathbf{w}}(n) = \mathbf{w}_{opt}$.

Prenons l'espérance de (5.47) :

$$E[\epsilon(n+1)] = E[(\mathbf{I} - \mu \mathbf{u}(n)\mathbf{u}^T(n))\epsilon(n)] + \mu E[\mathbf{u}(n)e_{opt}(n)]. \quad (5.48)$$

Le second terme de droite de cette équation est nul en raison du théorème de la projection qui précise, qu'à l'optimum (pour \mathbf{w}_{opt}) l'erreur et les entrées sont orthogonales (c'est-à-dire statistiquement non corrélées).

Calculons maintenant le premier terme. Puisque $\epsilon(n) = \hat{\mathbf{w}}(n) - \mathbf{w}_{opt}$ est supposé indépendant de $\mathbf{u}(n)$, on peut factoriser le premier terme :

$$\begin{aligned} E[(\mathbf{I} - \mu \mathbf{u}(n)\mathbf{u}^T(n))\epsilon(n)] &= E[\mathbf{I} - \mu \mathbf{u}(n)\mathbf{u}^T(n)]E[\epsilon(n)] \\ &= (\mathbf{I} - \mu \mathbf{R}_{uu})E[\epsilon(n)]. \end{aligned} \quad (5.49)$$

En appliquant sur la moyenne de l'erreur LMS, la même décomposition de \mathbf{R}_{uu} que pour l'algorithme de descente du gradient : $\mathbf{R}_{uu} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, on arrive à la condition :

$$0 < \mu < \frac{2}{\lambda_{max}}. \quad (5.50)$$

Si μ satisfait cette condition, on peut donc affirmer (sous les hypothèses formulées, qui sont assez fortes, et donc pas forcément vérifiées) que l'algorithme LMS converge en moyenne, c'est-à-dire que, pour $n \rightarrow +\infty$, $\hat{\mathbf{w}}(n)$ tend vers la solution optimale \mathbf{w}_{opt} de l'équation de Wiener-Hopf.

Convergence en moyenne quadratique

On calcule la matrice de variance-covariance de l'erreur paramétrique :

$$\mathbf{K}(n) = E[\epsilon(n)\epsilon^T(n)]; \quad (5.51)$$

Les calculs sont un peu complexes, et on donnera uniquement les principaux résultats. L'erreur s'exprime alors par :

$$J(n) = J_{min} + \text{tr}(\mathbf{R}_{uu}\mathbf{K}(n)), \quad (5.52)$$

et on trouve :

$$\mathbf{K}(n+1) = \mathbf{K}(n) - \mu[\mathbf{R}_{uu}\mathbf{K}(n) + \mathbf{K}(n)\mathbf{R}_{uu}] + \mu^2\mathbf{R}_{uu}\text{tr}(\mathbf{R}_{uu}\mathbf{K}(n)) + \mu^2\mathbf{R}_{uu}\mathbf{K}(n)\mathbf{R}_{uu} + \mu^2 J_{min}\mathbf{R}_{uu}. \quad (5.53)$$

Les fluctuations autour de l'optimum J_{min} sont égales à $\text{tr}(\mathbf{R}_{uu}\mathbf{K}(n))$. En utilisant la décomposition $\mathbf{R}_{uu} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ et en posant $\mathbf{Q}^T\mathbf{K}(n)\mathbf{Q} = \mathbf{X}(n)$, on en déduit $\mathbf{K}(n) = \mathbf{Q}\mathbf{X}(n)\mathbf{Q}^T$ et :

$$\begin{aligned} \text{tr}(\mathbf{R}_{uu}\mathbf{K}(n)) &= \text{tr}(\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T\mathbf{K}(n)) \\ &= \text{tr}(\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T\mathbf{Q}\mathbf{X}(n)\mathbf{Q}^T) \\ &= \text{tr}(\mathbf{Q}\mathbf{\Lambda}\mathbf{X}(n)\mathbf{Q}^T). \end{aligned} \quad (5.54)$$

L'opérateur trace acceptant la commutation des matrices, on peut encore écrire :

$$\begin{aligned} \text{tr}(\mathbf{R}_{uu}\mathbf{K}(n)) &= \text{tr}(\mathbf{Q}\mathbf{\Lambda}\mathbf{X}(n)\mathbf{Q}^T) \\ &= \text{tr}(\mathbf{Q}^T\mathbf{Q}\mathbf{\Lambda}\mathbf{X}(n)) \\ &= \text{tr}(\mathbf{\Lambda}\mathbf{X}(n)) \\ &= \sum_{k=1}^M \lambda_k X_k(n). \end{aligned} \quad (5.55)$$

En pratique, on peut montrer qu'il faut une nouvelle condition sur μ :

$$\sum_{k=1}^M M \frac{\mu\lambda_k}{2 - \mu\lambda_k} < 1. \quad (5.56)$$

Si $\mu \ll 2/\lambda_k, \forall k$, cette condition devient :

$$\sum_{k=1}^M M \frac{\mu\lambda_k}{2} < 1, \quad (5.57)$$

d'où

$$\frac{\mu}{2} \sum_{k=1}^M M\lambda_k < 1, \quad (5.58)$$

soit finalement :

$$0 < \mu < \frac{2}{\sum_{k=1}^M M\lambda_k}. \quad (5.59)$$

Or, les λ_k sont les valeurs propres de $\mathbf{R}_{uu} = E[\mathbf{u}(n)\mathbf{u}^T(n)]$ dont les éléments diagonaux sont $E[u_k^2(n)] = \sigma_u^2$. Pour un filtre d'ordre M (M paramètres), on a :

$$\sum_{k=1}^M M\lambda_k = M\sigma_u^2, \quad (5.60)$$

d'où la condition pratique pour μ (sous l'hypothèse $\mu \ll 2/\lambda_{max}$, qui assure une convergence en moyenne quadratique) :

$$0 < \mu < \frac{2}{M\sigma_u^2}. \quad (5.61)$$

5.3.3 Comparaison LMS/Descente du gradient

L'algorithme de descente du gradient tend vers l'erreur minimale J_{min} lorsque $n \rightarrow \infty$ et ceci de façon exponentielle régulière, en raison de l'estimation exacte de \mathbf{R}_{uu} et \mathbf{R}_{ud} .

Au contraire, l'algorithme LMS, qui repose sur des approximations très grossières (et donc à fortes variances) de \mathbf{R}_{uu} et \mathbf{R}_{ud} , converge vers $J(\infty) \neq J_{min}$, avec des oscillations chaotiques autour de J_{min} , et ceci même avec un pas μ correctement choisi.

L'avantage du LMS reste sa simplicité. Cette méthode est aussi très intéressante si les paramètres optimaux, \mathbf{w}_{opt} varient au cours des itérations.

5.3.4 Algorithme LMS normalisé

La correction de l'algorithme LMS :

$$\mu \mathbf{u}(n)e(n) \quad (5.62)$$

dépend directement de $\mathbf{u}(n)$. Si $\mathbf{u}(n)$ est grand, la correction le sera aussi, ce qui entraîne une amplification du bruit du gradient. Pour éviter ce problème, on a proposé un algorithme LMS normalisé caractérisé par l'équation :

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\hat{\mu}}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n)e(n), \quad (5.63)$$

avec $0 < \hat{\mu} < 2$. Pour éviter l'explosion du gain normalisé lorsque $\|\mathbf{u}(n)\|^2 \ll 1$, on régularise le dénominateur en ajoutant un petit scalaire positif qui limite sa valeur supérieure :

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\hat{\mu}}{\|\mathbf{u}(n)\|^2 + \epsilon} \mathbf{u}(n)e(n), \quad (5.64)$$

5.4 Algorithme des moindres carrés

Contrairement à l'équation de Wiener-Hopf, qui cherche à minimiser l'erreur quadratique moyenne $E[e^2(n)]$, on se propose ici de minimiser la somme des erreurs quadratiques :

$$\mathcal{E}(k_2) = \sum_{k=k_1}^{k_2} e^2(k), \quad (5.65)$$

sur une fenêtre $[k_1, k_2]$ de taille limitée. Dans la suite, on se placera dans le cas où $k = 1$, et on introduit une pondération exponentielle dans la fonction de coût :

$$\mathcal{E}(n) = \sum_{k=1}^n \lambda^{n-k} e^2(k). \quad (5.66)$$

Si $\lambda = 1$, $\mathcal{E}(n)$ est simplement la somme des erreurs quadratiques. Si $\lambda < 1$, les erreurs passées sont pondérées avec un poids (facteur d'oubli) qui décroît exponentiellement. Asymptotiquement,

c'est-à-dire pour $n \rightarrow +\infty$, si le signal est ergodique, on montre que l'on tend vers la solution de Wiener-Hopf.

Nous n'étudierons pas l'implantation classique de l'algorithme des moindres carrés, mais sa version récursive, qui se propose de calculer les termes à l'instant $n + 1$ en fonction des termes à l'instant n , sans tout recalculer. L'objectif est de conserver l'optimalité par rapport à $\mathcal{E}(n)$ tout en diminuant la complexité des calculs.

5.4.1 Equations préliminaires

A l'instant n , on a les équations suivantes :

- Somme des erreurs quadratiques : $\mathcal{E}(n) = \sum_{k=1}^n \lambda^{n-k} e^2(k)$;
- Erreur quadratique instantanée à l'instant k , calculée en utilisant toutes les données jusqu'à l'instant n , c'est-à-dire en prenant les paramètres calculés à cet instant, soit $\mathbf{w}(n) : e(k) = d(k) - y(k) = d(k) - \mathbf{w}^T(n)\mathbf{u}(k)$;
- Vecteur d'entrée du filtre à l'instant $k : \mathbf{u}(k) = (u(k), u(k-1), \dots, u(k-M+1))^T$;
- Vecteur de paramètres du filtre à l'instant $n : \mathbf{w}(n) = (w_0(n), w_1(n), \dots, w_{M-1}(n))^T$.

Les équations normales du filtre doivent satisfaire l'équation normale :

$$\Phi_{\mathbf{uu}}(n)\mathbf{w}(n) = \theta(n), \quad (5.67)$$

avec :

$$\begin{aligned} \Phi_{\mathbf{uu}}(n) &= \sum_{k=1}^n \lambda^{n-k} \mathbf{u}(k)\mathbf{u}^T(k) \\ \theta(n) &= \sum_{k=1}^n \lambda^{n-k} \mathbf{u}(k)d(k). \end{aligned} \quad (5.68)$$

Pour $\lambda = 1$, la matrice $\Phi_{\mathbf{uu}}(n)$ est une approximation de la matrice d'auto-corrélation. Elle est symétrique et telle que :

$$\begin{aligned} (\Phi_{\mathbf{uu}}(n))_{ii} &= \frac{1}{n} \sum_{k=1}^n u^2(k-i+1) \\ (\Phi_{\mathbf{uu}}(n))_{ij} &= \frac{1}{n} \sum_{k=1}^n u(k-i+1)u(k-j+1). \end{aligned} \quad (5.69)$$

Dans la mesure où les estimations de $\Phi_{\mathbf{uu}}(n)$ et $\theta(n)$ font intervenir le même facteur $1/n$, on peut ignorer celui-ci dans les équations. Pour $\lambda = 1$; on peut alors voir cette matrice comme le produit de deux matrices de données :

$$\Phi_{\mathbf{uu}}(n) = \mathbf{U}^T \mathbf{U}, \quad (5.70)$$

avec $\mathbf{U}^T = (\mathbf{u}(n), \mathbf{u}(n-1), \dots, \mathbf{u}(1))$, que l'on peut écrire :

$$\mathbf{U}^T = \begin{pmatrix} u(n) & u(n-1) & \dots & u(1) \\ u(n-1) & u(n-2) & & u(0) \\ \vdots & & \ddots & \vdots \\ u(n-M+1) & u(n-M) & \dots & u(1-M+1) \end{pmatrix} \quad (5.71)$$

Dans le cas où $\lambda \neq 1$, les termes d'auto- et d'inter-corrélation sont calculés par :

$$\begin{aligned} \Phi_{\mathbf{uu}}(n) &= \sum_{k=1}^n \lambda^{n-k} \mathbf{u}(k)\mathbf{u}^T(k) \\ \theta(n) &= \sum_{k=1}^n \lambda^{n-k} \mathbf{u}(k)d(k). \end{aligned} \quad (5.72)$$

où l'on ignore toujours le facteur $1/n$ devant les sommes.

5.4.2 Forme récursive pour le calcul de $\Phi_{\mathbf{uu}}(n)$ et $\theta(n)$

Pour éviter de calculer complètement les termes d'inter- et d'auto-corrélation, on cherche une relation de récurrence qui permette un calcul exact plus simple, en utilisant le calcul effectué à l'instant $(n - 1)$: on cherche donc à exprimer $\Phi_{\mathbf{uu}}(n)$ comme un ajustement de $\Phi_{\mathbf{uu}}(n - 1)$. Ce type de relation est dite récursive.

Partant de $\Phi_{\mathbf{uu}}(n)$, on peut écrire :

$$\begin{aligned}\Phi_{\mathbf{uu}}(n) &= \sum_{k=1}^n \lambda^{n-k} \mathbf{u}(k) \mathbf{u}^T(k) \\ &= \lambda \sum_{k=1}^{n-1} \lambda^{(n-1)-k} \mathbf{u}(k) \mathbf{u}^T(k) + \mathbf{u}(n) \mathbf{u}^T(n) \\ &= \lambda \Phi_{\mathbf{uu}}(n-1) + \mathbf{u}(n) \mathbf{u}^T(n).\end{aligned}\quad (5.73)$$

De façon similaire pour $\theta(n)$, on calcule :

$$\begin{aligned}\theta(n) &= \sum_{k=1}^n \lambda^{n-k} \mathbf{u}(k) d(k) \\ &= \lambda \sum_{k=1}^{n-1} \lambda^{(n-1)-k} \mathbf{u}(k) d(k) + \mathbf{u}(n) d(n) \\ &= \lambda \theta(n-1) + \mathbf{u}(n) d(n).\end{aligned}\quad (5.74)$$

La résolution de l'équation normale :

$$\Phi_{\mathbf{uu}}(n) \hat{\mathbf{w}}(n) = \theta(n), \quad (5.75)$$

s'effectue en calculant l'inverse de $\Phi_{\mathbf{uu}}(n)$ à chaque pas de calcul n :

$$\hat{\mathbf{w}}(n) = \Phi_{\mathbf{uu}}^{-1}(n) \theta(n). \quad (5.76)$$

Pour éviter un coût de calcul important, on cherche aussi une forme récursive pour $\Phi_{\mathbf{uu}}^{-1}(n)$, c'est-à-dire à exprimer $\Phi_{\mathbf{uu}}^{-1}(n+1)$ en fonction de $\Phi_{\mathbf{uu}}^{-1}(n)$.

5.4.3 Lemme d'inversion matricielle

Lemme 5.4.1 Soient deux matrices définies positives \mathbf{A} et \mathbf{B} , de dimension $M \times M$, liées par la relation :

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C} \mathbf{D}^{-1} \mathbf{C}^T, \quad (5.77)$$

où \mathbf{D} est une autre matrice définie positive de taille $N \times N$, alors

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B} \mathbf{C} (\mathbf{D} + \mathbf{C}^T \mathbf{B} \mathbf{C})^{-1} \mathbf{C}^T \mathbf{B}. \quad (5.78)$$

La preuve du lemme est obtenue simplement en multipliant les deux équations. Elle est laissée à titre d'exercice.

Nous allons utiliser le lemme d'inversion pour calculer $\Phi_{\mathbf{uu}}^{-1}(n)$ à partir de la forme récurrente

$$\Phi_{\mathbf{uu}}(n) = \lambda \Phi_{\mathbf{uu}}(n-1) + \mathbf{u}(n) \mathbf{u}^T(n), \quad (5.79)$$

en posant simplement :

$$\begin{aligned}\mathbf{A} &= \Phi_{\mathbf{uu}}(n) \\ \mathbf{B}^{-1} &= \lambda \Phi_{\mathbf{uu}}(n-1) \\ \mathbf{C} &= \mathbf{u}(n) \\ \mathbf{D} &= 1.\end{aligned}\quad (5.80)$$

Il vient alors :

$$\Phi_{\mathbf{uu}}^{-1}(n) = \frac{1}{\lambda} \Phi_{\mathbf{uu}}^{-1}(n-1) - \frac{1}{\lambda^2} \frac{\Phi_{\mathbf{uu}}^{-1}(n-1) \mathbf{u}(n) \mathbf{u}^T(n) \Phi_{\mathbf{uu}}^{-1}(n-1)}{1 + \lambda^{-1} \mathbf{u}^T(n) \Phi_{\mathbf{uu}}^{-1}(n-1) \mathbf{u}(n)}. \quad (5.81)$$

Appelons $\mathbf{k}(n)$, le gain vectoriel défini par :

$$\mathbf{k}(n) = \frac{\lambda^{-1} \Phi_{\mathbf{uu}}^{-1}(n-1) \mathbf{u}(n)}{1 + \lambda^{-1} \mathbf{u}^T(n) \Phi_{\mathbf{uu}}^{-1}(n-1) \mathbf{u}(n)}, \quad (5.82)$$

on peut alors écrire :

$$\begin{aligned} \mathbf{k}(n) &= \lambda^{-1} \Phi_{\mathbf{uu}}^{-1}(n-1) \mathbf{u}(n) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^T(n) \Phi_{\mathbf{uu}}^{-1}(n-1) \mathbf{u}(n) \\ &= [\lambda^{-1} \Phi_{\mathbf{uu}}^{-1}(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^T(n) \Phi_{\mathbf{uu}}^{-1}(n-1)] \mathbf{u}(n) \\ &= \Phi_{\mathbf{uu}}^{-1}(n) \mathbf{u}(n) \\ &= \mathbf{P}(n) \mathbf{u}(n), \end{aligned} \quad (5.83)$$

en posant $\mathbf{P}(n) = \Phi_{\mathbf{uu}}^{-1}(n)$. La formule récursive du calcul de $\Phi_{\mathbf{uu}}^{-1}(n)$ est alors :

$$\Phi_{\mathbf{uu}}^{-1}(n) = \lambda^{-1} \Phi_{\mathbf{uu}}^{-1}(n-1) - \mathbf{k}(n) \lambda^{-1} \mathbf{u}^T(n) \Phi_{\mathbf{uu}}^{-1}(n-1), \quad (5.84)$$

ou bien :

$$\mathbf{P}(n) = \lambda^{-1} \mathbf{P}(n-1) - \mathbf{k}(n) \lambda^{-1} \mathbf{u}^T(n) \mathbf{P}(n-1). \quad (5.85)$$

Ce calcul permet de calculer exactement, à chaque pas n , le vecteur des paramètres par la relation :

$$\hat{\mathbf{w}}(n) = \Phi_{\mathbf{uu}}^{-1}(n) \theta(n) = \mathbf{P}(n) \theta(n). \quad (5.86)$$

On est donc sûr que l'algorithme converge vers la solution des moindres carrés. On rappelle que cet algorithme minimise la somme des erreurs quadratiques, mais pas l'erreur quadratique moyenne, sauf asymptotiquement.

5.4.4 Initialisation de l'algorithme des moindres carrés récurrents

Cet algorithme est souvent appelé RLS, pour *recursive least square*. Pour exécuter l'algorithme, il faut définir une valeur initiale, non singulière, de $\Phi_{\mathbf{uu}}^{-1}(0) = \mathbf{P}(0)$.

Cette initialisation peut être réalisée par un premier, et unique, calcul de matrice inverse :

$$\Phi_{\mathbf{uu}}^{-1}(0) = \left(\sum_{k=-n_0}^0 \mathbf{u}(k) \mathbf{u}^T(k) \right)^{-1}. \quad (5.87)$$

On peut aussi, plus simplement, modifier la matrice $\Phi_{\mathbf{uu}}(n)$, de façon à la régulariser afin que l'inverse existe toujours :

$$\Phi_{\mathbf{uu}}(n) = \sum_{k=1}^n \lambda^{n-k} \mathbf{u}(k) \mathbf{u}^T(k) + \delta \lambda^n \mathbf{I}, \quad (5.88)$$

avec $\Phi_{\mathbf{uu}}(0) = \delta \mathbf{I}$, d'où $\mathbf{P}(0) = \Phi_{\mathbf{uu}}^{-1}(0) = \delta^{-1} \mathbf{I}$. Cette procédure peut être vue comme une modification des fenêtres de pondération. La valeur du paramètre δ n'est pas critique. On prend souvent $\delta \ll 0.01 \sigma_u^2$.

Enfin, en absence d'information *a priori*, on initialise le vecteur paramètres à $\mathbf{w}(0) = 0$.

Chapitre 6

Filtre de Kalman discret

6.1 Introduction

6.1.1 Histoire

Ce filtre a été proposé par R. Kalman en 1960 pour résoudre un problème de poursuite de trajectoire dans la préparation des missions Appolo. Rudolf Kalman est né à Budapest, en Hongrie, le 19 mai 1930. Il vit aujourd'hui en Suisse.

6.1.2 Equations de base

Le filtrage de Kalman repose sur un ensemble de deux équations :

- l'équation récursive de modélisation (génération) du processus aléatoire \mathbf{x}_k discret,
- l'observation (mesure) du processus \mathbf{z}_k ,

où on note l'instant discret k par un indice k . Typiquement, on a alors le système :

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k \\ \mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \end{cases} \quad (6.1)$$

où

- \mathbf{x}_k est le vecteur d'état du processus, de taille $n \times 1$,
- $\mathbf{\Phi}_k$ est la matrice de transition d'état, de taille $n \times n$, qui décrit la dynamique des \mathbf{x}_k ,
- \mathbf{w}_k est l'erreur de modélisation du processus, supposée blanche, de covariance connue et de moyenne nulle,
- \mathbf{z}_k est le vecteur de mesure à l'instant k , de taille $m \times 1$,



FIGURE 6.1 – Rudolf Kalman, inventeur du filtrage de Kalman

- \mathbf{H}_k est une matrice de taille $m \times n$,
- \mathbf{v}_k est l'erreur de mesure, de taille $m \times 1$, supposée blanche, de covariance connue et non corrélée avec \mathbf{w}_k .

Dans la suite, on notera :

- $E[\mathbf{w}_k \mathbf{w}_i^T] = \mathbf{Q}_k \delta_{ik}$ (taille $n \times n$),
- $E[\mathbf{v}_k \mathbf{v}_i^T] = \mathbf{R}_k \delta_{ik}$ (taille $n \times n$),
- $E[\mathbf{w}_k \mathbf{v}_i^T] = 0, \forall k, i$.

L'objectif est de déterminer une estimation récursive des \mathbf{x}_k à partir des mesures \mathbf{z}_k et du modèle de génération du processus (première équation).

6.2 Erreurs *a priori* et *a posteriori*

6.2.1 Erreur *a priori*

On se place à l'instant discret k . A cet instant, on dispose d'une estimation initiale fondée sur la connaissance du processus et des mesures jusqu'à l'instant précédent, c'est-à-dire $k - 1$. Cette estimation est appelée estimation *a priori*, et nous la noterons :

$$\hat{\mathbf{x}}_{k/k-1} = \hat{\mathbf{x}}_k^- \quad (6.2)$$

On peut alors définir l'erreur *a priori* :

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \quad (6.3)$$

ainsi que la matrice de covariance de l'erreur *a priori* :

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- (\mathbf{e}_k^-)^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \quad (6.4)$$

6.2.2 Equation de mise à jour et erreur *a posteriori*

A partir de l'estimation *a priori*, $\hat{\mathbf{x}}_k^-$, on propose la mise à jour suivante :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (6.5)$$

Nous justifierons la forme de cette équation, et en particulier du terme correctif, par la suite. Le gain matriciel, \mathbf{K}_k , est à déterminer de façon à fournir une estimation $\hat{\mathbf{x}}_k$ optimale au sens de l'erreur quadratique moyenne (Wiener). On définit alors l'erreur *a posteriori* :

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \quad (6.6)$$

et la matrice de covariance de l'erreur *a posteriori* :

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \quad (6.7)$$

A partir de l'équation de l'estimation $\hat{\mathbf{x}}_k$, on peut écrire :

$$\begin{aligned} \mathbf{x}_k - \hat{\mathbf{x}}_k &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \\ &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \\ &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- - \mathbf{K}_k [\mathbf{H}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{v}_k] \\ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k \mathbf{v}_k. \end{aligned} \quad (6.8)$$

On a alors pour la matrice de covariance de l'erreur *a posteriori* :

$$\begin{aligned}\mathbf{P}_k &= E\left[\left((\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k \mathbf{v}_k\right)\left((\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k \mathbf{v}_k\right)^T\right] \\ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T.\end{aligned}\quad (6.9)$$

Les autres termes de l'espérance sont nuls en raison de la non corrélation de \mathbf{v}_k avec les \mathbf{x}_k et $\hat{\mathbf{x}}_k^-$.

6.2.3 Optimisation du gain \mathbf{K}_k

Dans ce paragraphe nous allons calculer \mathbf{K}_k de sorte que l'erreur quadratique moyenne soit minimale pour toutes les composantes de \mathbf{x}_k , c'est-à-dire telle que la trace de la matrice \mathbf{P}_k soit minimale. En effet :

$$\text{tr}(\mathbf{P}_k) = \sum_{i=1}^n E[(e_k^2(k))_i]. \quad (6.10)$$

On rappelle les deux règles de calcul suivante concernant la dérivée de la trace :

$$\frac{d\text{tr}(\mathbf{A}\mathbf{B})}{d\mathbf{A}} = \mathbf{B}^T = \frac{d\text{tr}(\mathbf{B}^T \mathbf{A}^T)}{d\mathbf{A}}, \quad (6.11)$$

et

$$\frac{d\text{tr}(\mathbf{A}\mathbf{C}\mathbf{A}^T)}{d\mathbf{A}} = 2\mathbf{A}\mathbf{C}, \quad (6.12)$$

où \mathbf{C} est une matrice symétrique. On peut alors développer l'expression de la matrice de covariance de l'erreur *a posteriori* (6.9) afin de l'exprimer comme une fonction de \mathbf{K}_k :

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T. \quad (6.13)$$

En utilisant les règles de calcul rappelées plus haut, calculons maintenant la dérivée de la trace de \mathbf{P}_k par rapport à \mathbf{K}_k :

$$\frac{d\text{tr}(\mathbf{P}_k)}{d\mathbf{K}_k} = 0 - 2(\mathbf{H}_k \mathbf{P}_k^-)^T + 2\mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k). \quad (6.14)$$

Ce terme s'annule si :

$$-2(\mathbf{H}_k \mathbf{P}_k^-)^T + 2\mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) = 0, \quad (6.15)$$

c'est-à-dire si :

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \quad (6.16)$$

Ce gain optimal, au sens de l'erreur quadratique moyenne, c'est-à-dire qui entraîne une erreur *a posteriori* minimale, est appelé gain de Kalman.

6.2.4 Matrice de covariance de l'erreur *a posteriori* optimale

En reportant la valeur optimale du gain (gain de Kalman) dans l'équation (6.13), on obtient la matrice \mathbf{P}_k optimale :

$$\begin{aligned}\mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T \\ &= \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k^T \mathbf{P}_k^-\end{aligned}\quad (6.17)$$

On peut aussi écrire, à partir de cette expression, en remarquant que le gain de Kalman apparaît dans le dernier terme de droite :

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- . \quad (6.18)$$

Enfin, en partant la relation $\mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) = \mathbf{P}_k^- \mathbf{H}_k^T$, puis en transposant l'égalité et en utilisant les symétries de \mathbf{P}_k^- et de \mathbf{R}_k , on obtient :

$$\mathbf{H}_k \mathbf{P}_k^- = (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T . \quad (6.19)$$

En reportant cette expression dans (6.18), on peut alors écrire :

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T . \quad (6.20)$$

6.3 Algorithmes

La mise en oeuvre de l'algorithme de Kalman peut prendre des formes diverses, en particulier selon l'ordre des calculs effectués. Nous proposons dans la suite quelques algorithmes afin d'illustrer ces différentes possibilités.

6.3.1 Algorithme 1

Initialisation : \hat{x}_0^- et \mathbf{P}_0^-

Boucle

-- Calcul du gain de Kalman : $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$

-- Estimation : $\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{x}_k^-)$

-- Calcul de la matrice de covariance de l'erreur : $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$

-- Préparation du pas suivant : $\hat{x}_{k+1}^- = \mathbf{\Phi}_k \hat{x}_k$ et $\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k$

FinBoucle

6.3.2 Forme alternative

Comme nous l'avons dit, la suite des opérations dans la boucle principale peut être modifiée, ce qui peut présenter des avantages numériques, même si, formellement, la boucle globale reste similaire. Dans ce paragraphe, nous proposons un autre algorithme, fondé sur une autre expression du gain de Kalman et de l'inverse de la matrice de covariance de l'erreur.

Expression récursive de \mathbf{P}_k^{-1}

On utilisera l'expression de l'inverse de la matrice de covariance de l'erreur *a posteriori* :

$$\mathbf{P}_k^{-1} = (\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k . \quad (6.21)$$

Pour montrer cette expression, il suffit de vérifier la relation $\mathbf{P}_k \mathbf{P}_k^{-1} = \mathbf{I}$. Pour cela, on utilise pour \mathbf{P}_k la relation (6.17), et on écrit :

$$\begin{aligned} & [\mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^-] [(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k] \\ = & \mathbf{I} + \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \\ = & \mathbf{I} - \mathbf{P}_k^- \mathbf{H}_k^T [-\mathbf{R}_k + (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} + (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1}] \mathbf{H}_k \\ = & \mathbf{I} - \mathbf{P}_k^- \mathbf{H}_k^T [-\mathbf{R}_k + (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} (\mathbf{I} + \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1})] \mathbf{H}_k \\ = & \mathbf{I} + \mathbf{P}_k^- \mathbf{H}_k^T [-\mathbf{R}_k + (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} (\mathbf{R}_k + \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T) \mathbf{R}_k^{-1}] \mathbf{H}_k \\ = & \mathbf{I} . \end{aligned} \quad (6.22)$$

Calcul de \mathbf{K}_k

On cherche une autre équation pour le calcul du gain de Kalman, en partant de l'expression (6.16), et en multipliant à gauche et à droite du premier terme par $\mathbf{P}_k \mathbf{P}_k^{-1}$ et $\mathbf{R}_k^{-1} \mathbf{R}_k$, respectivement. On a alors :

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^{-1} \mathbf{H}_k (\mathbf{H}_k \mathbf{P}_k^{-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ &= \mathbf{P}_k \mathbf{P}_k^{-1} \mathbf{P}_k^{-1} \mathbf{H}_k \mathbf{R}_k^{-1} \mathbf{R}_k (\mathbf{H}_k \mathbf{P}_k^{-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \end{aligned} \quad (6.23)$$

En remplaçant \mathbf{P}_k^{-1} par la relation trouvée au paragraphe précédent, on a :

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k [(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k \mathbf{R}_k^{-1} \mathbf{H}_k] \mathbf{P}_k^{-1} \mathbf{H}_k \mathbf{R}_k^{-1} \mathbf{R}_k (\mathbf{H}_k \mathbf{P}_k^{-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ &= \mathbf{P}_k [\mathbf{I} + \mathbf{H}_k \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k^-] \mathbf{H}_k \mathbf{R}_k^{-1} (\mathbf{H}_k \mathbf{P}_k^{-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} + \mathbf{I})^{-1} \\ &= \mathbf{P}_k [\mathbf{H}_k \mathbf{R}_k^{-1} + \mathbf{H}_k \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k \mathbf{R}_k^{-1}] (\mathbf{H}_k \mathbf{P}_k^{-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} + \mathbf{I})^{-1} \\ &= \mathbf{P}_k \mathbf{H}_k \mathbf{R}_k^{-1} [\mathbf{I} + \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k \mathbf{R}_k^{-1}] (\mathbf{H}_k \mathbf{P}_k^{-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} + \mathbf{I})^{-1} \\ &= \mathbf{P}_k \mathbf{H}_k \mathbf{R}_k^{-1}. \end{aligned} \quad (6.24)$$

Algorithme 2

A partir des deux expressions précédentes, on remarque que le gain de Kalman fait intervenir la matrice \mathbf{P}_k : il faut donc faire le calcul du gain après le calcul de \mathbf{P}_k , contrairement à l'algorithme 1. On a alors l'algorithme 2.

Initialisation : $\hat{\mathbf{x}}_0^-$ et \mathbf{P}_0^-

Boucle

```
-- Calcul de la matrice de covariance de l'erreur :  $\mathbf{P}_k^{-1} = (\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k$  puis calcul de  $\mathbf{P}_k$  par inversion de  $\mathbf{P}_k^{-1}$ 
-- Calcul du gain de Kalman :  $\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k \mathbf{R}_k^{-1}$ 
-- Estimation :  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$ 
-- Préparation du pas suivant :  $\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_k$  et  $\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k$ , puis  $(\mathbf{P}_{k+1}^-)^{-1}$  par inversion de  $\mathbf{P}_{k+1}^-$ 
```

FinBoucle

Cet algorithme demande deux inversions de matrices, au début et à la fin de la boucle, ce qui est parfois coûteux.

6.4 Innovation

Revenons maintenant sur le terme correctif utilisé pour la mise à jour *a posteriori* de l'état dans l'équation (6.25). Ce terme correctif, qui est multiplié par le gain de Kalman, s'écrit :

$$\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \quad (6.25)$$

et nous nous proposons de l'interpréter dans ce paragraphe.

De façon générale, on rappelle l'équation d'observation :

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \quad (6.26)$$

6.4.1 Prédiction de $\hat{\mathbf{z}}_{k/(k-1)}$

A partir des mesures $\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{k-1}$, on peut calculer l'estimation optimale (au sens de l'erreur quadratique moyenne) :

$$\hat{\mathbf{z}}_{k/(k-1)} = \mathbf{H}_k \hat{\mathbf{x}}_{k/(k-1)} + \hat{\mathbf{v}}_{k/(k-1)}. \quad (6.27)$$

Or, selon nos notations, cette estimation a été appelée estimation *a priori* et notée avec un indice supérieur $-$. Ainsi, on a :

$$\hat{\mathbf{x}}_{k/(k-1)} = \hat{\mathbf{x}}_k^-. \quad (6.28)$$

De plus, par hypothèse, \mathbf{v}_k est indépendant des $\mathbf{z}_i, \forall i \leq k-1$. Par conséquent, l'estimation - au sens de l'erreur quadratique moyenne minimale - de \mathbf{v}_k sachant $\mathbf{z}_0, \dots, \mathbf{z}_{k-1}$ est égale à la moyenne conditionnelle (voir mon cours de théorie de l'estimation), qui est ici égale à la moyenne simple, en raison de l'indépendance des variables. Autrement dit :

$$\hat{\mathbf{v}}_{k/(k-1)} = E[\mathbf{v}_k / \mathbf{z}_0, \dots, \mathbf{z}_{k-1}] = E[\mathbf{v}_k] = 0, \quad (6.29)$$

car \mathbf{v}_k est supposée de moyenne nulle. On a donc sent :

$$\hat{\mathbf{z}}_{k/(k-1)} = \mathbf{H}_k \hat{\mathbf{x}}_{k/(k-1)} = \mathbf{H}_k \hat{\mathbf{x}}_k^-. \quad (6.30)$$

6.4.2 Innovation

Le terme correctif peut donc être écrit :

$$\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- = \mathbf{z}_k - \hat{\mathbf{z}}_{k/k-1}. \quad (6.31)$$

Ce terme s'appelle l'innovation (de l'observation) et correspond à l'information *nouvelle* apportée par l'observation à l'instant k , *nouvelle* au sens où elle n'est pas prédite par les observations jusqu'à l'instant $k-1$ inclus.

L'innovation, notée :

$$\alpha_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- = \mathbf{z}_k - \hat{\mathbf{z}}_{k/k-1}. \quad (6.32)$$

présente des propriétés intéressantes, exposées ci-dessous.

Propriété 6.4.1 *L'innovation α_k est orthogonale à $\hat{\mathbf{z}}_{k/k-1}$. Cette propriété est la simple conséquence du théorème de la projection orthogonale. Or, $\hat{\mathbf{z}}_{k/k-1}$ étant une combinaison linéaire des $\mathbf{z}_0, \dots, \mathbf{z}_{k-1}$, l'innovation α_k est orthogonale à $\mathbf{z}_i, \forall i \leq k-1$. On a donc les relations :*

$$E[\alpha_k^T \mathbf{z}_i] = 0, \forall i = 0, 1, \dots, k-1. \quad (6.33)$$

Propriété 6.4.2 *L'innovation α_k est orthogonale aux innovations passées, c'est-à-dire à $\alpha_i, \forall i \leq k-1$. En effet, α_i est une combinaison linéaire des $\mathbf{z}_0, \dots, \mathbf{z}_i$. L'application de la propriété précédente conduit au résultat :*

$$E[\alpha_k^T \alpha_i] = 0, \forall i = 0, 1, \dots, k-1. \quad (6.34)$$

6.4.3 Interprétation de la mise à jour

A partir de l'équation de mise à jour (6.25) :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-), \quad (6.35)$$

on voit que :

- le premier terme de droite s'écrit encore $\hat{\mathbf{x}}_k^- = \Phi_k\hat{\mathbf{x}}_{k-1}$, c'est donc la prédiction *a priori* de \mathbf{x}_k en utilisant le modèle, mais pas l'observation ;
- le second terme est la correction de la prédiction fondée sur le modèle seul, correction qui repose elle sur l'innovation, c'est-à-dire directement sur les observations.

On peut aussi remarquer que le gain de Kalman varie aussi selon la confiance que l'on peut accorder aux observations. En effet, d'après l'équation (6.24), le gain de Kalman peut s'écrire :

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}_k\mathbf{R}_k^{-1}, \quad (6.36)$$

avec $\mathbf{R}_k = E[\mathbf{v}_k\mathbf{v}_k^T]$. Pour simplifier l'interprétation, supposons que $\mathbf{R}_k = \sigma_v^2\mathbf{I}$, c'est-à-dire que les bruits de mesures sont spatialement décorrélés et de même variance. L'inverse de \mathbf{R}_k est alors égale à $\mathbf{R}_k^{-1} = \sigma_v^{-2}\mathbf{I}$ et le gain de Kalman devient simplement :

$$\mathbf{K}_k = \frac{\mathbf{P}_k\mathbf{H}_k}{\sigma_v^2}. \quad (6.37)$$

- Si l'observation n'est pas fiable, c'est-à-dire si le bruit de mesure a une très forte variance ($\sigma_v^2 \rightarrow +\infty$), le gain de Kalman

$$\mathbf{K}_k = \frac{\mathbf{P}_k\mathbf{H}_k}{\sigma_v^2} \rightarrow 0, \quad (6.38)$$

et la mise à jour est réduite à

$$\hat{\mathbf{x}}_k \rightarrow \hat{\mathbf{x}}_k^- = \Phi_{k-1}\hat{\mathbf{x}}_{k-1}. \quad (6.39)$$

- Au contraire, si l'observation est très fiable, la variance $\sigma_v^2 \rightarrow 0$ et le gain de Kalman devient très grand, de sorte que la mise à jour repose de façon prépondérante sur l'observation et très peu sur le modèle.

Si l'erreur d'estimation est grande, on sait que $\text{tr}(\mathbf{P}_k)$ est grande. En supposant que les erreurs sont spatialement décorrélés et de même variance, c'est-à-dire que $\mathbf{P}_k = \sigma_e^2\mathbf{I}$, on en déduit que le gain de Kalman devient :

$$\mathbf{K}_k = \frac{\sigma_e^2}{\sigma_v^2}\mathbf{H}_k. \quad (6.40)$$

Cette dernière équation montre que, sous ces hypothèses simplificatrices, le gain de Kalman fluctue :

- le gain \mathbf{K}_k diminue si l'estimation par le modèle devient plus précise ;
- le gain \mathbf{K}_k augmente si les observations deviennent plus précises.

6.4.4 Stabilité du filtre de Kalman

Pratiquement, le gain de Kalman varie dans le temps, et c'est un des avantages du filtre de Kalman. Cependant, pour des applications où le filtre, après un certain nombre d'itérations, atteint un état quasi-permanent, le gain est constant, et on peut étudier la stabilité de l'équation de mise à jour :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-). \quad (6.41)$$

Pour cette étude, on utilise l'équation du modèle $\hat{\mathbf{x}}_k^- = \Phi_{k-1} \hat{\mathbf{x}}_{k-1}$, si bien que (6.41) devient :

$$\begin{aligned} \hat{\mathbf{x}}_k &= \Phi_{k-1} \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \\ &= (\Phi_{k-1} - \mathbf{K}_k \mathbf{H}_k \Phi_{k-1}) \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k \mathbf{z}_k. \end{aligned} \quad (6.42)$$

En passant dans le domaine de représentation en z (par transformée en z), on arrive à l'équation :

$$\begin{aligned} \hat{\mathbf{X}}_k(z) &= (\Phi_{k-1} - \mathbf{K}_k \mathbf{H}_k \Phi_{k-1}) z^{-1} \hat{\mathbf{X}}_k(z) + \mathbf{K}_k \mathbf{Z}_k(z) \\ (z\mathbf{I} - (\Phi_{k-1} - \mathbf{K}_k \mathbf{H}_k \Phi_{k-1})) \hat{\mathbf{X}}_k(z) &= z \mathbf{K}_k \mathbf{Z}_k(z). \end{aligned} \quad (6.43)$$

La stabilité du filtre est alors liée aux racines du polynôme caractéristique : $\det[z\mathbf{I} - (\Phi_{k-1} - \mathbf{K}_k \mathbf{H}_k \Phi_{k-1})] = 0$, et la stabilité est garantie si les valeurs propres de $\Phi_{k-1} - \mathbf{K}_k \mathbf{H}_k \Phi_{k-1}$ ont leurs parties réelles inférieures à 1.

6.5 Un filtre de Kalman pour éliminer le 50 Hz

6.5.1 Position du problème

Les signaux biomédicaux sont des signaux d'amplitude généralement faible, enregistrés à l'aide d'électrodes à très grande impédance. Les signaux sont pollués par des perturbations d'origines diverses : d'autres signaux d'origine biologique (respiration, électromyogramme, électrocardiogramme, etc.) et des signaux artificiels, en particulier le 50 Hz, présent au niveau des alimentations par le secteur, et dans l'environnement, par rayonnement électro-magnétique.

La fréquence du secteur est généralement très précise : $f = 50$ Hz à 0.5 % près, selon les normes d'EDF, mais le rapport signal à bruit (RSB) peut varier énormément. Pour cette raison, il peut être intéressant d'ajuster le filtrage (fréquence centrale et surtout bande passante) pour maximiser le RSB. C'est dans cet esprit que nous proposons d'implanter ce filtre réjecteur du 50 Hz à l'aide d'un filtre de Kalman.

Ce filtre a été conçu par un de mes doctorants, Reza Sameni (actuellement maître de conférences à l'université de Shiraz, Iran), durant sa thèse (préparée à GIPSA-lab, Grenoble en co-tutelle avec l'Université de Technology Sharif de Téhéran, Iran) pour éliminer le 50 Hz sur des signaux électrocardiographiques (ECG). C'est un très bel exemple pédagogique, par son réalisme et sa simplicité, de conception d'un filtre de Kalman.

6.5.2 Modèle de génération du signal

Le signal à éliminer est supposé de forme sinusoïdale, soit en temps discret :

$$x(n) = B \cos(\omega n + \phi), \quad (6.44)$$

où la pulsation $\omega = 2\pi f$ est supposée connue ($f = 50$ Hz !) et B et ϕ sont inconnues.

Relation de récurrence

Pour mettre en oeuvre le filtre de Kalman, il faut d'abord exprimer $x(n)$ sous forme d'une équation d'état récursive. Pour cela, écrivons d'abord $x(n+1)$:

$$\begin{aligned} x(n+1) &= B \cos(\omega(n+1) + \phi) \\ &= B[\cos(\omega n + \phi) \cos \omega - \sin(\omega n + \phi) \sin \omega], \end{aligned} \quad (6.45)$$

puis $x(n-1)$:

$$\begin{aligned} x(n-1) &= B \cos(\omega(n-1) + \phi) \\ &= B[\cos(\omega n + \phi) \cos \omega + \sin(\omega n + \phi) \sin \omega]. \end{aligned} \quad (6.46)$$

En sommant les deux équations, on a :

$$\begin{aligned} x(n+1) + x(n-1) &= 2B \cos(\omega) \cos(\omega n + \phi) \\ &= 2 \cos(\omega) x(n). \end{aligned} \quad (6.47)$$

On remarque que ni la phase ϕ , ni l'amplitude B n'interviennent explicitement dans cette dernière expression. Cependant, si ces deux quantités varient, les équations précédentes ne sont plus exactement vérifiées. Les deux paramètres ω et ϕ sont "non linéaires" (car ils interviennent à l'intérieur d'une fonction cosinus) et les erreurs sur ces paramètres ne sont pas bien modélisées par un bruit additif, contrairement à l'amplitude B . Cependant, ω et ϕ sont très peu perturbés, en raison des exigences de qualité d'EDF.

L'observation sera donc de la forme :

$$y(n) = x(n) + v(n), \quad (6.48)$$

où $v(n)$ représente les erreurs du modèle et les signaux aléatoires d'origine biologique. Ici, l'objectif est d'estimer $x(n)$ afin de l'éliminer ensuite par soustraction sur $y(n)$, car le signal d'intérêt est le terme aléatoire $v(n)$. Ce signal ne sera pas supposé centré : un ECG n'est pas centré.

Equations d'état

Posons $x'(n) = x(n-1)$ et définissons le vecteur d'état :

$$\mathbf{x}(n) = (x(n) \quad x'(n))^T = (x(n) \quad x(n-1))^T. \quad (6.49)$$

On peut alors écrire l'équation du modèle, en utilisant la relation (6.47) :

$$\begin{pmatrix} x(n+1) \\ x(n) \end{pmatrix} = \begin{pmatrix} 2 \cos \omega & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x(n) \\ x(n-1) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} w(n) \quad (6.50)$$

et

$$y(n) = (1 \quad 0) \begin{pmatrix} x(n) \\ x(n-1) \end{pmatrix} + v(n). \quad (6.51)$$

En posant les notations suivantes :

$$\begin{aligned} \mathbf{x}(n) &= (x(n)x(n-1))^T \\ \mathbf{b} &= (1 \quad 0)^T \\ \mathbf{h}(n) &= (1 \quad 0)^T \\ \Phi(n) &= \begin{pmatrix} 2 \cos \omega & -1 \\ 1 & 0 \end{pmatrix} \end{aligned} \quad (6.52)$$

Remarquons enfin que $\Phi(n) = \Phi$ ne dépendant pas de n , on arrive alors aux équations d'état :

$$\begin{cases} \mathbf{x}(n+1) &= \Phi \mathbf{x}(n) + w(n) \mathbf{b} \\ y(n) &= \mathbf{h}^T \mathbf{x}(n) + v(n) \end{cases} \quad (6.53)$$

Dans ces équations, $v(n)$ n'est pas un signal de moyenne nulle mais est supposé blanc ; le bruit du modèle de génération de signal, $w(n)$, est lui supposé centré et blanc.

6.5.3 Algorithme de Kalman

Dans les équations d'état ci-dessus, on observe $y(n)$ et on connaît \mathbf{h}^T , \mathbf{b} et Φ (ce dernier est supposé constant). Les variances des bruits $w(n)$ et $v(n)$ sont des réels notés :

$$q = E[w^2(n)] \text{ et } r = E[v^2(n)]. \quad (6.54)$$

Il faudra attribuer une valeur initiale à ces variances, qui seront ensuite ajustées à chaque pas de calcul par l'algorithme.

Initialisation

- q peut être choisie comme une valeur arbitrairement petite,
- r , qui est la variance du terme "non power line", peut être approximée par la variance de $y(n)$ supposée centrée par hypothèse, donc $r = E[y^2(n)]$. On peut aussi calculer cette variance en supprimant le 50 Hz sur la DFT de $y(n)$, puis en calculant la variance du signal résiduel.

Algorithme

L'algorithme est constitué de 3 ensembles d'équations :

1. Propagation du pas k à $k+1$:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^- &= \Phi \hat{\mathbf{x}}_k, \\ \mathbf{P}_{k+1}^- &= \Phi \mathbf{P}_k \Phi^T + q \mathbf{b} \mathbf{b}^T; \end{aligned} \quad (6.55)$$

2. Calcul du gain de Kalman

$$\mathbf{K}_k = \frac{\mathbf{P}_k^- \mathbf{h}}{\mathbf{h}^T \mathbf{P}_k^- \mathbf{h} + r}, \quad (6.56)$$

car le terme du dénominateur est un scalaire ;

3. Estimation

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (y_k - \mathbf{h}^T \hat{\mathbf{x}}_k^-), \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{h}^T \mathbf{P}_k^-. \end{aligned} \quad (6.57)$$

6.5.4 Réponse du filtre

Lorsque le filtre de Kalman converge, c'est-à-dire en supposant qu'il atteigne un état stationnaire, on peut se demander quelles sont les propriétés du filtre. C'est ce que nous nous proposons d'étudier dans ce paragraphe.

Transformée en z du filtre

Partant de l'équation :

$$\begin{aligned}\hat{\mathbf{x}}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{h}^T) \hat{\mathbf{x}}_k^- + \mathbf{K}_k y_k \\ &= (\mathbf{I} - \mathbf{K}_k \mathbf{h}^T) \Phi \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k y_k,\end{aligned}\quad (6.58)$$

si $\mathbf{K}_k \rightarrow \bar{\mathbf{K}}$ lorsque $k \rightarrow +\infty$, cette équation devient :

$$\hat{\mathbf{x}}_k = (\mathbf{I} - \bar{\mathbf{K}} \mathbf{h}^T) \hat{\mathbf{x}}_k^- + \bar{\mathbf{K}} y_k. \quad (6.59)$$

Dans le domaine des fréquences (par transformée en z), on a :

$$\begin{aligned}\hat{\mathbf{X}}_k(z) &= (\mathbf{I} - \bar{\mathbf{K}} \mathbf{h}^T) \Phi \hat{\mathbf{X}}_{k-1}(z) + \bar{\mathbf{K}} Y_k(z) \\ &= (\mathbf{I} - \bar{\mathbf{K}} \mathbf{h}^T) \Phi z^{-1} \hat{\mathbf{X}}_k(z) + \bar{\mathbf{K}} Y_k(z),\end{aligned}\quad (6.60)$$

En factorisant les termes en $\hat{\mathbf{X}}_k(z)$, on a :

$$[\mathbf{I} - (\mathbf{I} - \bar{\mathbf{K}} \mathbf{h}^T) \Phi z^{-1}] \hat{\mathbf{X}}_k(z) = \bar{\mathbf{K}} Y_k(z), \quad (6.61)$$

soit finalement :

$$\mathbf{H}(z) = \frac{\hat{\mathbf{X}}_k(z)}{Y_k(z)} = [\mathbf{I} - (\mathbf{I} - \bar{\mathbf{K}} \mathbf{h}^T) \Phi z^{-1}]^{-1} \bar{\mathbf{K}}, \quad (6.62)$$

avec $\bar{\mathbf{K}} = (k_1 k_2)^T$. Puisque $\hat{\mathbf{X}}_k(z) = (\hat{X}_{1k}(z) \hat{X}_{2k}(z))^T$, le filtre a deux composantes, notées $H_1(z)$ et $H_2(z)$. On peut d'abord calculer le dénominateur des filtres. Le calcul n'est pas compliqué, mais demande un peu de soin et d'attention. Pour cela, on pose $M = [\mathbf{I} - (\mathbf{I} - \bar{\mathbf{K}} \mathbf{h}^T) \Phi z^{-1}]$, et on trouve :

$$M = \begin{pmatrix} 1 - 2z^{-1}(1 - k_1) \cos \omega & (1 - k_1)z^{-1} \\ (2k_2 \cos \omega - 1)z^{-1} & 1 - k_2 z^{-1} \end{pmatrix} \quad (6.63)$$

L'inversion de M demande le calcul de son déterminant, qui vaut :

$$\det M = 1 + [2k_1 \cos \omega - 2 \cos \omega - k_2]z^{-1} + (1 - k_1)z^{-2}. \quad (6.64)$$

On en déduit l'inverse de M :

$$M^{-1} = \frac{1}{\det M} \begin{pmatrix} 1 - k_2 z^{-1} & -(1 - k_1)z^{-1} \\ -(2k_2 \cos \omega - 1)z^{-1} & 1 - 2(1 - k_1) \cos \omega z^{-1} \end{pmatrix} \quad (6.65)$$

Le filtre total est égal à $\mathbf{H}(z) = M^{-1} \bar{\mathbf{K}}$ qui vaut :

$$\mathbf{H}(z) = \frac{1}{\det M} \begin{pmatrix} k_1 - k_2 z^{-1} \\ k_2 + (k_1 - 2k_2 \cos \omega)z^{-1} \end{pmatrix} \quad (6.66)$$

Le filtre qui nous intéresse est le filtre $H_1(z)$, qui extrait le 50 Hz et ses harmoniques. Pour avoir le signal filtré, il faut donc calculer :

$$x_f(k) = x(k) - \hat{x}_1(k), \quad (6.67)$$

où le dernier terme est l'estimation des termes à 50 Hz. Le filtre réalisant cette opération est donc le filtre $G(z) = 1 - H_1(z)$, c'est-à-dire :

$$G(z) = \frac{1 - 2 \cos(\omega)z^{-1} + z^{-2}}{\alpha - (2 \cos(\omega) + \beta)z^{-1} + z^{-2}} \quad (6.68)$$

avec $\alpha = 1/(1 - k_1)$ et $\beta = k_2/(1 - k_1)$.

6.5.5 Analyse fonctionnelle du filtre Notch du second ordre

Ce filtre (6.68) a la forme standard d'un filtre Notch du second ordre :

$$\frac{1 + \delta}{2} \frac{1 - 2\gamma z^{-1} + z^{-2}}{1 - 2\gamma(1 + \delta)z^{-1} + \delta z^{-2}}, \quad (6.69)$$

avec $\delta = (1 - \tan(BW/2))/(1 + \tan(BW/2))$ et $\gamma = \cos(\omega)$, où ω est la pulsation du réjecteur ($f = \omega/(2\pi)$ est la fréquence Notch) et BW est la largeur de bande à -3 dB.

Dans la mesure où le gain de Kalman fluctue au cours des itérations de l'algorithme, les paramètres k_1 et k_2 fluctuent aussi, ainsi que α et β . Autrement dit, la fréquence Notch et la largeur de bande sont ajustées en permanence selon le signal et ses caractéristiques.

6.5.6 Programmes et résultats sur des ECG

Le programme, écrit en Matlab en 2007, par un de mes doctorants, Reza Sameni, est donné ci-dessous. Il est constitué d'un programme principal qui lit les données, appelle la fonction KF-Notch, et présente le signal initial, le signal obtenu après filtrage, ainsi que leurs densités spectrales de puissance. La fonction KFNotch est l'implantation du filtre de Kalman, réjecteur du 50 Hz, conçu dans les paragraphe précédent. La boucle principale, avec les différentes étapes de calcul du filtre de Kalman, est située à la fin du programme.

Le programme

```
%
% Test program for Kalman notch filter
%
% Dependencies: The baseline wander toolbox of the Open Source ECG Toolbox
%
% Open Source ECG Toolbox, version 1.0, October 2007
% Released under the GNU General Public License
% Copyright (C) 2007 Reza Sameni
% Sharif University of Technology, Tehran, Iran - GIPSA-LAB, INPG, Grenoble,
% France, reza.sameni@gmail.com
%
% SampleECG2 est un enregistrement de 16 canaux de 60000 échantillons, dont
% une portion seulement est prélevée, sur un seul canal.

clc
clear all
close all;}

% lecture des données
load('SampleECG2.mat'); data = data(1:1500,6);

fs = 1000; %fréquence d'échantillonnage
f0 = 50; % fréquence initiale à rejeter
```

```

% calcul du numéro des échantillons à partir de 0
n = (0:length(data)-1)';

% ECG pollué par un signal harmonique
x = data + .1*sin(2*pi*n*f0/fs)/std(data);

% appel de la fonctionKFNotch
[y1,y2,Pbar,Phat,PSmoothed,Kgain] = KFNotch(x,f0,fs);

%[y1,y2,Pbar,Phat,PSmoothed,Kgain] = KFNotch(x,f0,fs,1e-4,.1*var(x),.9);

% calcul de t, temps correspondant à l'échantillon n
t = n/fs;

% représentation des données initiales, polluées et filtrées
figure;
hold on;
plot(t,data,'b');
plot(t,x,'r');
plot(t,y1,'m');
plot(t,y2,'g');
grid;
xlabel('time (sec.)');
legend('original ECG','noisy ECG','Kalman filter','Kalman smoother');

% représentation de la DSP des données polluées
figure;
psd(x,1000,fs);
title('noisy spectrum');

% représentation de la DSP de la sortie du filtre de Kalman
figure;
psd(y1,1000,fs);
title('Kalman filter output spectrum');

% représentation de la DSP de la sortie du lisseur de Kalman
figure;
psd(y2,1000,fs);
title('Kalman smoother output spectrum');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
function [y1,y2,Pbar,Phat,PSmoothed,Kgain] = KFNotch(x,f0,fs,varargin)
%
% [y1,y2,Pbar,Phat,PSmoothed,Kgain] = KFNotch(x,f0,fs,Q,R,gamma),
% Removing Power-Line noise using a linear Kalman filter and smoother

```

```

%
% inputs:
% x: vector of noisy signals contaminated with power-line
% f0: power-line frequency
% fs: sampling rate
% Q: covariance of the model errors (optional). By default: 1e-4*max(abs(x))
% R: covariance of the power-line noise (optional). By default: var(x)
%
% output:
% y1: vector of denoised signal using the Kalman filter
% y2: vector of denoised signal using the Kalman smoother
% Pbar: covariance matrix of the a priori error vector of the
% Kalman filter
% Phat: covariance matrix of the a posteriori error vector of
% the Kalman filter
% PSmoothed: covar. matrix of the a posteriori error vector of
% the Kalman smoother
% Kgain: vector of Kalman filter gain
%
% Reference:
% R. Sameni, "A Kalman Notch Filter for Removing Power-Line Noise from
% Biomedical Signals", Technical Report, GIPSA-LAB, October 15th 2007.
%
% Open Source ECG Toolbox, version 1.0, October 2007
% Released under the GNU General Public License
% Copyright (C) 2007 Reza Sameni
% Sharif Univ. of Technology, Tehran, Iran -- GIPSA-LAB, INPG, Grenoble,
% France, reza.sameni@gmail.com
%
%////////////////////////////////////
%
% lecture des arguments, pour les compléter par des valeurs par défauts
if(nargin>3 && ~isempty(varargin{1})),
    Q = varargin{1};
else
    Q = 1e-4*max(abs(x));
end

if(nargin>4 && ~isempty(varargin{2})),
    R = varargin{2};
else
    R = var(x);
end

if(nargin>5 && ~isempty(varargin{3})),
    gamma = varargin{3};
else

```

```

    gamma = 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialization : Kalman filter parameters
w = 2*pi*f0/fs;

Wmean = 0;
Vmean = 0;
% valeur initiale de l'estimation a priori
X0 = [x(2) x(1)]';
% valeur initiale de la matrice de variance-covariance de l'erreur a priori
P0 = 1000*Q;

L = 2;
A = [2*cos(w) -1 ; 1 0]; % matrice de transition d'état
H = [1 0]; % matrice d'observation
Xminus = X0; % init. de l'estimation a priori
Pminus = P0; % init. de la matrice de vc des erreurs a priori
Samples = length(x);

VarWinlen = ceil(fs/10);
mem2 = zeros(VarWinlen,1) + R;
Xhat = zeros(2,Samples);
innovations = zeros(1,Samples);
Phat = zeros(2,2,Samples);
Xbar = zeros(2,Samples);
Pbar = zeros(2,2,Samples);
Kgain = zeros(2,Samples);
B = [1 0]';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Forward Filtering Stage
for k = 1 : Samples,
    % Store results
    Xbar(:,k) = Xminus;
    Pbar(:, :, k) = Pminus;

    % Measurement update (a posteriori updates)
    Yminus = H*Xminus + Vmean;

    % Kalman gain
    K = Pminus*H' / (H*Pminus*H' + R);
    % Stabilized Kalman cov. matrix
    Pplus = (eye(L) - K*H) * Pminus * (eye(L) - K*H)' + K*R*K';
    innovations(k) = x(k) - Yminus;
    % a posteriori state estimate

```

```

Xplus = Xminus + K*(innovations(k));
% Observation covariance matrix update (for nonstationary signals)
mem2 = [innovations(k).^2 ; mem2(1:end-1)];
R = gamma*R + (1-gamma)*mean(mem2);
% Prevent nonpositive definiteness of the covariance matrix
Pplus = (Pplus + Pplus')/2;

% Time update (a priori updates)
Xminus = A*Xplus + Wmean;           % State update
Pminus = A*Pplus*A' + B*Q*B';       % Cov. matrix update

% Store results
Xhat(:,k) = Xplus;
Phat(:,:,k) = Pplus;
Kgain(:,k) = K;
end

```

Résultats

La figure 6.2 présente le signal initial (une partie d'un signal ECG dont la fréquence de répétition est de l'ordre du Hz) et le signal pollué par le 50 Hz (de fréquence environ 50 fois plus élevée que l'ECG) et les signaux après filtrage. On voit que les signaux après filtrage se superposent presque parfaitement au signal ECG non bruité.

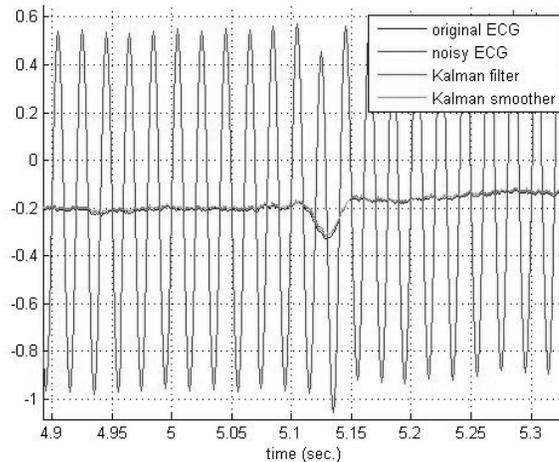


FIGURE 6.2 – Signal ECG initial, signal fortement bruité avec le 50 Hz, et signaux obtenus à partir de deux versions du filtrage de Kalman

Les deux figures 6.3 et 6.4 montrent respectivement les densités spectrales de puissance du signal bruitée par le 50 Hz, et du signal obtenu après filtrage. Sur la première figure, on observe une raie d'amplitude importante à 50 Hz qui correspond à la perturbation liée au secteur. Sur la DSP du signal filtré, on voit clairement la réjection des fréquences aux alentours du 50 Hz. La profondeur de la réjection, et la largeur, dépendent des bruits de modèle et de mesure : comme

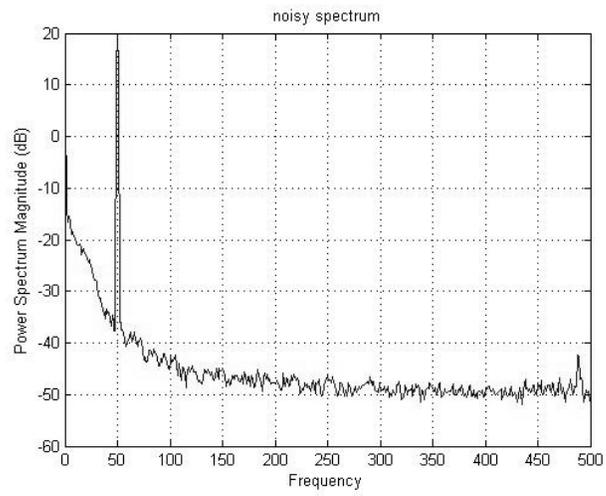


FIGURE 6.3 – DSP du signal bruité : on voit clairement la raie de très forte puissance aux alentours de 50 Hz

nous l'avons expliqué, le filtre de Kalman calcule de façon permanent le meilleur filtre réjecteur, en fonction de la qualité du modèle et des mesures.

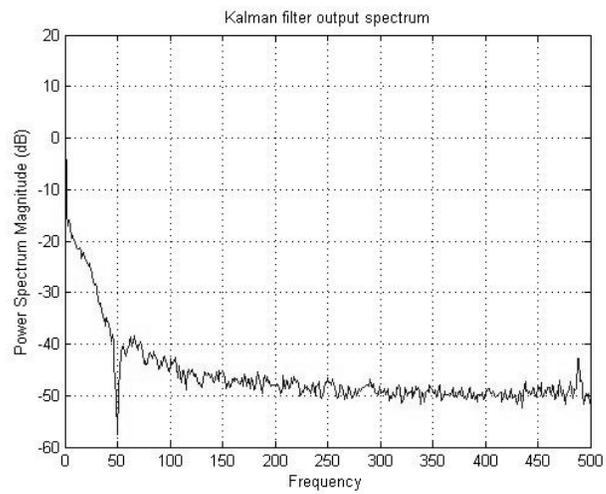


FIGURE 6.4 – DSP du signal filtré : on voit clairement la réjection aux alentours de 50 Hz