

7. Filtrage optimal

3 grandes méthodes de filtrage optimal :

(filtrage sous contrainte d'un signal altéré par un bruit)

■ Filtrage par moindres carrés

- Minimisation de l'énergie de l'erreur entre le signal mesuré et l'estimation (par un modèle) du signal non bruité.
- Modèle adopté pour le signal non altéré = modèle linéaire.
- Il y a lissage du bruit plus que sa réjection.
- Le filtrage des moindres carrés se place en déterministe.
- Il y a contrainte de stationnarité.

■ Filtrage de Wiener

- Méthode des moindres carrés en stochastique.
- Il y a contrainte de stationnarité.

■ Filtrage de Kalman

- Poursuite d'un modèle d'Etat (modèle de Gauss-Markov) nécessaire et relativement précis du signal non bruité.
- Pas de contrainte de stationnarité.
- Si modèle correct → réjection du bruit et non pas lissage.
- Si modèle non réaliste → les résultats peuvent être très mauvais.

I. MOINDRES CARRES

Moindres carrés direct Algorithme LS (*Least Squares*)

Un système linéaire de n équations à n inconnues : $\mathbf{Xh} = \mathbf{y}$
 où :

\mathbf{X} désigne une matrice carrée de dimension $n \times n$

\mathbf{h} désigne un vecteur de dimension $n \times 1$

\mathbf{y} désigne un vecteur de dimension $n \times 1$

possède une solution unique : $\mathbf{h} = \mathbf{X}^{-1} \mathbf{y}$ si \mathbf{X} inversible.

→ Erreur $\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{Xh}$ nulle : $\boldsymbol{\varepsilon} = \mathbf{0}$

En traitement du signal, de nombreux signaux peuvent se modéliser sous la forme \mathbf{Xh} où :

\mathbf{X} désigne une matrice connue

et \mathbf{h} désigne un vecteur inconnu.

Exemple : Filtre RIF de RI \mathbf{h} inconnue :

\mathbf{X} : matrice obtenue à partir de valeurs de l'entrée

\mathbf{Xh} : signal de sortie (produit scalaire = convolution).

En l'absence de bruit, l'observation de n valeurs du signal de sortie suffit donc à estimer \mathbf{h} :

$$\mathbf{h} = \mathbf{X}^{-1} \mathbf{y} \quad (\text{déconvolution})$$

En pratique, il n'en est jamais ainsi (*bruit de mesure*) :

On se donne plus d'observations (k) que d'inconnues (n) :

$$\boxed{k \geq n}$$

La méthode des moindres carrés fournit alors le vecteur \mathbf{h} à estimer (de dimension n) optimal au sens quadratique à partir des k observations.

Notations :

\mathbf{y} : vecteur d'observation

\mathbf{w} bruit additif : $\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{w}$

\mathbf{X} matrice de dimension $k \times n$

\mathbf{h} vecteur de dimension $n \times 1$

\mathbf{y} vecteur de dimension $k \times 1$

$$\boxed{k \geq n} \quad (\text{ système surdéterminé })$$

On a plus d'équations k que d'inconnues n .

Conséquence :

Il n'existe pas en général de vecteur \mathbf{h} tel que $\mathbf{X}\mathbf{h}$ soit parfaitement égal à l'observation \mathbf{y} .

Problème :

Trouver \mathbf{h} tel que \mathbf{Xh} soit le plus proche possible de \mathbf{y}

\Leftrightarrow Minimisation de l'erreur quadratique ε^2 (énergie de l'erreur ε) :

$$\varepsilon^2 = J(\mathbf{h}) = (\mathbf{y} - \mathbf{Xh})^T (\mathbf{y} - \mathbf{Xh})$$

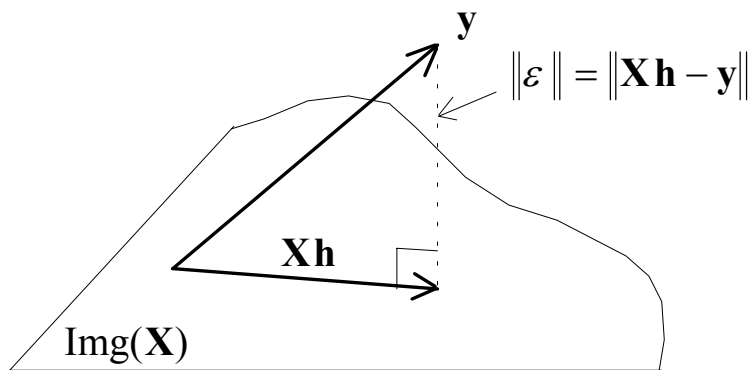
Solution : Théorème de projection :

\rightarrow Détermination de \mathbf{h} tel que $\frac{\partial \varepsilon^2}{\partial \mathbf{h}} = 0$

$$\frac{\partial \varepsilon^2}{\partial \mathbf{h}} = -2 \mathbf{X}^T \cdot (\mathbf{y} - \mathbf{Xh}) = 0 \quad (\text{produit scalaire nul})$$

$$\rightarrow \mathbf{X}^T \cdot (\mathbf{y} - \mathbf{Xh}) = 0 \rightarrow \boxed{\{\varepsilon = (\mathbf{y} - \mathbf{Xh})\} \perp \mathbf{X}^T}$$

\mathbf{h} est tel que l'écart $\varepsilon = \mathbf{y} - \mathbf{Xh}$ est \perp à $\mathbf{X}^T \rightarrow \varepsilon \perp \mathbf{Xh}$:



Théorème de projection

$$\varepsilon \perp \mathbf{X}^T \rightarrow \mathbf{X}^T (\mathbf{y} - \mathbf{Xh}) = \mathbf{0} \rightarrow \mathbf{X}^T \mathbf{y} = \mathbf{X}^T \mathbf{Xh}$$

$$\rightarrow \text{Si la matrice carrée } \mathbf{X}^T \mathbf{X} \text{ inversible : } \boxed{\mathbf{h} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}}$$

Moindres carrés récursif Algorithme RLS

- Moindres carrés direct : $\mathbf{h} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

Algorithme non temps réel (inversion de matrice).

- Moindres carrés récursif :

Mise à jour de \mathbf{h} au fur et à mesure que les données arrivent.

A l'étape i : $\mathbf{h}_i = \mathbf{P}_i \mathbf{X}_i^T \mathbf{y}_i$ avec : $\mathbf{P}_i = (\mathbf{X}_i^T \mathbf{X}_i)^{-1}$

Notation :

$$\mathbf{X}_{i+1} = \begin{bmatrix} \mathbf{X}_i \\ \mathbf{x}_{i+1}^T \end{bmatrix} \quad (\mathbf{x}_{i+1}^T = (i+1)^{\text{ième}} \text{ ligne de la matrice } \mathbf{X}_{i+1})$$

$$\rightarrow \mathbf{X}_{i+1}^T \mathbf{X}_{i+1} = \begin{bmatrix} \mathbf{X}_i^T & \mathbf{x}_{i+1} \end{bmatrix} \begin{bmatrix} \mathbf{X}_i \\ \mathbf{x}_{i+1}^T \end{bmatrix} = \mathbf{X}_i^T \mathbf{X}_i + \mathbf{x}_{i+1}^T \mathbf{x}_{i+1}$$

$$\rightarrow \mathbf{P}_{i+1} = (\mathbf{P}_i^{-1} + \mathbf{x}_{i+1}^T \mathbf{x}_{i+1})^{-1}$$

Identité pour une matrice \mathbf{R} et un vecteur \mathbf{u} :

$$(\mathbf{R} + \mathbf{u} \mathbf{u}^T)^{-1} = \mathbf{R}^{-1} - \frac{\mathbf{R}^{-1} \mathbf{u}^T \mathbf{u} \mathbf{R}^{-1}}{1 + \mathbf{u} \mathbf{R}^{-1} \mathbf{u}^T}$$

Application à \mathbf{P}_{i+1} : $\rightarrow \mathbf{P}_{i+1} = \mathbf{P}_i - \frac{\mathbf{P}_i \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T \mathbf{P}_i}{1 + \mathbf{x}_{i+1}^T \mathbf{P}_i \mathbf{x}_{i+1}}$

$\rightarrow \mathbf{h}_{i+1} = \mathbf{P}_{i+1} \begin{bmatrix} \mathbf{X}_i^T & \mathbf{x}_{i+1} \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_{i+1} \end{bmatrix} = \mathbf{P}_{i+1} (\mathbf{X}_i^T \mathbf{y}_i + \mathbf{x}_{i+1} \mathbf{y}_{i+1})$

$\rightarrow \mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{K}_i (y_{i+1} - \mathbf{x}_{i+1} \mathbf{h}_i)$

avec : $\mathbf{K}_i = \frac{\mathbf{P}_i \mathbf{x}_{i+1}}{1 + \mathbf{x}_{i+1}^T \mathbf{P}_i \mathbf{x}_{i+1}}$ (matrice dim. $k \times n$)

Algorithme des moindres carrés récurrents :

▪ Valeurs initiales : $i = 1$: $\bullet \mathbf{P}_i = \frac{\mathbf{I}}{\delta}$ $\bullet \mathbf{h}_i = \mathbf{0}$ avec :

$\mathbf{I}_{(n+1) \times (n+1)}$: identité et δ coefficient réel : $\delta > 0$ et $\delta \ll 1$

▪ Pour $i = 1, \dots, k-1$ Répéter : $\bullet \mathbf{K}_i = \frac{\mathbf{P}_i \mathbf{x}_{i+1}}{1 + \mathbf{x}_{i+1}^T \mathbf{P}_i \mathbf{x}_{i+1}}$

avec : $\mathbf{x}_i^T = [x_{i+n} \quad x_{i+n-1} \quad \dots \quad x_i]$

$\bullet \mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{K}_i (y_{i+1} - \mathbf{x}_{i+1} \mathbf{h}_i)$

$\bullet \mathbf{P}_{i+1} = \mathbf{P}_i - \frac{\mathbf{P}_i \mathbf{x}_{i+1} \mathbf{x}_{i+1}^T \mathbf{P}_i}{1 + \mathbf{x}_{i+1}^T \mathbf{P}_i \mathbf{x}_{i+1}}$

A condition d'avoir k grand, il est inutile de partir des CIs exactes faisant intervenir le calcul de \mathbf{P}_1 et \mathbf{h}_1 à partir de la relation : $\mathbf{h}_i = \mathbf{P}_i \mathbf{X}_i^T \mathbf{y}_i$.

Moindres carrés moyens **LMS (*Least Mean Squares*)**

Algorithme des *Moindres carrés moyens* (*algorithme LMS*)

≡ *Gradient stochastique*

≡ version récursive du filtre de *Wiener* (*voir § Filtrage de Wiener*).

7 ANNEXE. Filtrage optimal

II. FILTRAGE DE WIENER

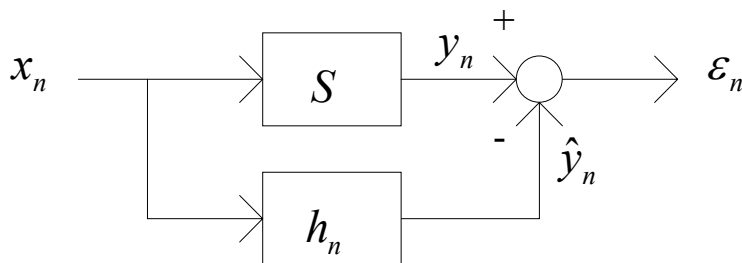
x_n et y_n sont 2 processus aléatoires, réels, centrés, SSL.

Détermination du filtre linéaire de RI Finie (MA) (N points) :

$$\{h_n\} = \{h_0, h_1, \dots, h_{N-1}\}$$

\equiv meilleure approximation du système S (critère quadratique) :

\rightarrow Recherche de $\{h_n\}$ qui minimise l'énergie de l'erreur $\varepsilon_n : E[\varepsilon_n^2]$



$$\varepsilon_n = y_n - \hat{y}_n$$

$$\hat{y}_n = h_n * x_n = \sum_{k=0}^{N-1} h_k x_{n-k} = h_0 x_n + h_1 x_{n-1} + \dots + h_{N-1} x_{n-N+1}$$

Théorème de projection : pour tout $k \in \{0, 1, \dots, N-1\}$:

$$E[(y_n - \hat{y}_n)x_{n-k}] = E[(y_n - h_0 x_n - h_1 x_{n-1} - \dots - h_{N-1} x_{n-N+1})x_{n-k}] = 0$$

$$\Leftrightarrow E[(h_0 x_n + h_1 x_{n-1} + \dots + h_{N-1} x_{n-N+1})x_{n-k}] = E[y_n x_{n-k}]$$

Sous forme matricielle (équation de **Wiener-Hopf**) : $\mathbf{R} \mathbf{h} = \mathbf{r}$

avec : $\mathbf{x}_n = [x_n \quad \dots \quad x_{n-N+1}]^T$ $\mathbf{h} = [h_0 \quad \dots \quad h_{N-1}]^T$

$$\mathbf{R} = E[\mathbf{x}_n \mathbf{x}_n^T] = \begin{bmatrix} R_{XX_0} & \cdots & R_{XX_{N-1}} \\ \vdots & \cdots & \vdots \\ R_{XX_{N-1}} & \cdots & R_{XX_0} \end{bmatrix}$$

$$\mathbf{r} = E[y_n \mathbf{x}_n] = \begin{bmatrix} R_{YX_0} \\ \vdots \\ R_{YX_{N-1}} \end{bmatrix}$$

Filtre de **Wiener**, solution de l'équation de **Wiener-Hopf**: $\mathbf{h} = \mathbf{R}^{-1} \mathbf{r}$

Algorithme du gradient (*version récurrente du filtre de Wiener*)

Equation récurrente : $\mathbf{h}(n) = \mathbf{h}(n - 1) + \mu [\mathbf{r} - \mathbf{R} \mathbf{h}(n - 1)]$

où : $\mathbf{h}(n) \equiv \mathbf{h}$ à l'itération n . Coeff. réel $\mu = \text{pas du gradient}$.

Si : $0 < \mu < \frac{2}{\max_i(\lambda_i)}$ ($\lambda_i = \text{valeurs propres de } \mathbf{R}$)

→ on montre que $\mathbf{h}(n)$ converge vers le filtre de Wiener.

Moindres carrés moyens **LMS** (*Least Mean Squares*)

Si \mathbf{R} et \mathbf{r} connus → algorithme du gradient (**Wiener**) **OK**.

Malheureusement, \mathbf{R} et \mathbf{r} sont généralement inconnus et doivent être estimés à partir des données observées.

Algorithme des *moindres carrés moyens* (\equiv *gradient stochastique* ou encore *LMS - Least Mean Squares*), algorithme adaptatif imaginé par **Widrow** (1985), évite le calcul de ces estimations.

Posons $\mathbf{x}_n = [x_n \quad \cdots \quad x_{n-N+1}]^T$ et $e(n) = y_n - \mathbf{x}_n^T \mathbf{h}(n - 1)$

On peut écrire : $\mathbf{r} - \mathbf{R} \mathbf{h}(n - 1) = E[y_n \mathbf{x}_n] - E[\mathbf{x}_n \mathbf{x}_n^T] \mathbf{h}(n - 1) = E[\mathbf{x}_n e(n)]$

→ L'équation récurrente $\mathbf{h}(n) = \mathbf{h}(n-1) + \mu [\mathbf{r} - \mathbf{R} \mathbf{h}(n-1)]$

devient : $\mathbf{h}(n) = \mathbf{h}(n-1) + \mu E[\mathbf{x}_n e(n)]$

Widrow remplace $E[\mathbf{x}_n e(n)]$ par sa valeur « instantanée » $\mathbf{x}_n e(n)$

→ *Algorithme LMS*

. Valeur initiale : $\mathbf{h}(0) = \mathbf{0}$

. Répéter :

$$\bullet e(n) = y_n - \mathbf{x}_n^T \mathbf{h}(n-1)$$

$$\bullet \mathbf{h}(n) = \mathbf{h}(n-1) + \mu \mathbf{x}_n e(n)$$

III. FILTRAGE DE KALMAN

Kalman s'affranchit de l'hypothèse de stationnarité des signaux.

Il met à profit la **Représentation d'Etat**.

Exemple :

Véhicule se déplaçant sur une droite avec une vitesse v constante.

Position initiale d_0 . Position à l'instant t : $d(t) = d_0 + vt$

$$\text{Equations (} t \text{ discret) } \begin{cases} d(t+1) = d(t) + v(t) \\ v(t+1) = v(t) \quad (=v) \end{cases} \quad \text{CIs: } \begin{cases} d(0) = d_0 \\ v(0) = v \end{cases}$$

En fait, le modèle : vitesse = constante est imprécis :

$$\begin{cases} d(t+1) = d(t) + v(t) + b_1(t) \\ v(t+1) = v(t) + b_2(t) \end{cases} \quad b_1(t), b_2(t) \text{ PAs (bruit de modèle)}$$

$$\rightarrow \begin{bmatrix} d(t+1) \\ v(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} b_1(t) \\ b_2(t) \end{bmatrix} \quad \text{et} \quad y(t) = d(t) + u(t)$$

$y(t)$ = Position $d(t)$ observée ($u(t)$ PA bruit de *mesure*, d'*observation*)

$$\begin{aligned} & \begin{bmatrix} d(t+1) \\ v(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} b_1(t) \\ b_2(t) \end{bmatrix} \\ \rightarrow & \begin{cases} y(t) = [1 & 0] \begin{bmatrix} d(t) \\ v(t) \end{bmatrix} + u(t) \end{cases} \end{aligned}$$

→ Ecriture matricielle :

$$\begin{cases} \mathbf{x}(t+1) = \mathbf{A} \mathbf{x}(t) + \mathbf{b}(t) & \text{équation d'Etat} \\ \mathbf{y}(t) = \mathbf{C} \mathbf{x}(t) + \mathbf{u}(t) & \text{équation d'observation} \end{cases}$$

avec :

- . Etat : $\mathbf{x}(t) = \begin{bmatrix} d(t) \\ v(t) \end{bmatrix}$
- . Observation : $\mathbf{y}(t)$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \mathbf{b}(t) = \begin{bmatrix} b_1(t) \\ b_2(t) \end{bmatrix} \quad \mathbf{C} = [1 \quad 0] \quad \mathbf{u}(t) = u(t)$$

Le filtre de Kalman fournit la meilleure estimation en *moyenne quadratique* de $\mathbf{x}(t)$ à partir de l'observation $\mathbf{y}(t)$.

(minimisation de l'erreur quadratique $E[\mathbf{x}(t) - \hat{\mathbf{x}}(t)]^2$)

. Forme la plus simple :

hypothèses de bruits de modèle et d'observation les pires (bruits blancs)

Filtre de Kalman récursif

Calcul direct :

quand t croît, la dimension des matrices croît de façon « explosive ».

Algorithme du filtre récursif de Kalman

- Valeurs initiales : $\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)]$, $\mathbf{K}(0) = E[\mathbf{x}(0)\mathbf{x}^T(0)]$
- Répéter pour $t \geq 1$:
 - $\mathbf{G}(t) = \mathbf{K}(t-1)\mathbf{C}^T [\mathbf{C}\mathbf{K}(t-1)\mathbf{C}^T + \mathbf{R}_u(0)]^{-1}$
 - $\hat{\mathbf{x}}(t) = \mathbf{A}\hat{\mathbf{x}}(t-1) + \mathbf{G}(t)[\mathbf{y}(t) - \mathbf{C}\mathbf{A}\hat{\mathbf{x}}(t-1)]$
 - $\mathbf{K}(t) = \mathbf{A}[\mathbf{K}(t-1) - \mathbf{G}(t)\mathbf{C}\mathbf{K}(t-1)\mathbf{A}^T + \mathbf{R}_b(0)]$

avec :

$$\mathbf{R}_b(\tau) = E[\mathbf{b}(t+\tau)\mathbf{b}^T(t)] = \begin{bmatrix} \sigma_{b_1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{b_2}^2 & 0 & \vdots \\ \vdots & \vdots & \vdots & 0 \\ 0 & \dots & 0 & \sigma_{b_n}^2 \end{bmatrix} \delta(\tau)$$

$$\mathbf{R}_u(\tau) = E[\mathbf{u}(t+\tau)\mathbf{u}^T(t)] = \begin{bmatrix} \sigma_{u_1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{u_2}^2 & 0 & \vdots \\ \vdots & \vdots & \vdots & 0 \\ 0 & \dots & 0 & \sigma_{u_p}^2 \end{bmatrix} \delta(\tau)$$

(t et τ sont *discrets*)