Chapitre 3 : Analyse spectrale paramétrique et filtrage numérique adaptatif

1. Estimation spectrale

L'estimation spectrale est l'ensemble des méthodes d'estimation de la densité spectrale de puissance.

La densité spectrale de puissance d'un processus stationnaire est la transformée de Fourier de la fonction d'autocorrélation. Elle représente la distribution de la puissance par unité de fréquence.

Les méthodes d'estimation spectrale de puissance peuvent être classées en deux catégories : méthodes non paramétriques ou classiques et méthodes paramétriques ou modernes.

Les méthodes non paramétriques d'estimation spectrale sont basées sur la transformée de Fourier : périodogramme, périodogramme modifié, corrélogramme, …

Les méthodes paramétriques d'estimation spectrale sont basées sur des modèles décrits par un ensemble de paramètres : modèle autorégressif à moyenne ajustée (ARMA), modèle AR, modèle MA, …

2. Modélisation paramétrique

2.1. Modèles à fonction de transfert rationnelle

Plusieurs processus rencontrés en pratique sont bien approximés par un modèle à fonction de transfert rationnelle. Dans ce modèle, la séquence x(n) est modélisée comme étant la sortie d'un système linéaire invariant dans le temps excité par une séquence d'entrée u(n) supposée un bruit blanc reliées par l'équation aux différence linéaire :

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + \sum_{k=0}^{q} b_k u(n-k) \qquad (1)$$

Ce modèle général est appelé modèle autorégressif à moyenne ajustée (ARMA : autoregressive moving average). L'importance de ces modèles vient de leur relation aux filtres linéaires à fonction de transfert rationnelle.

Le bruit u(n) est un bruit qui fait partie du modèle et il est différent du bruit additif ou le bruit observé rencontré dans plusieurs applications de traitement du signal. Tout bruit observé doit être modélisé dans le modèle ARMA.

La fonction de transfert entre l'entrée u(n) et la sortie x(n) pour le modèle ARMA est une fonction rationnelle :

$$H(z) = \frac{B(z)}{A(z)} \qquad (2)$$

$A(z)$: transformée en z de la branche AR $= \sum_{k=0}^{p} a_k z^{-k}$

$B(z)$: transformée en z de la branche MA $= \sum_{k=0}^{q} b_k z^{-k}$

Il est supposé que les zéros de $A(z)$ sont à l'intérieur du cercle unité afin que le filtre associé à $H(z)$ soit causal et stable. Sans cette supposition, (1) ne peut être une représentation valable d'un processus stationnaire au sens large.

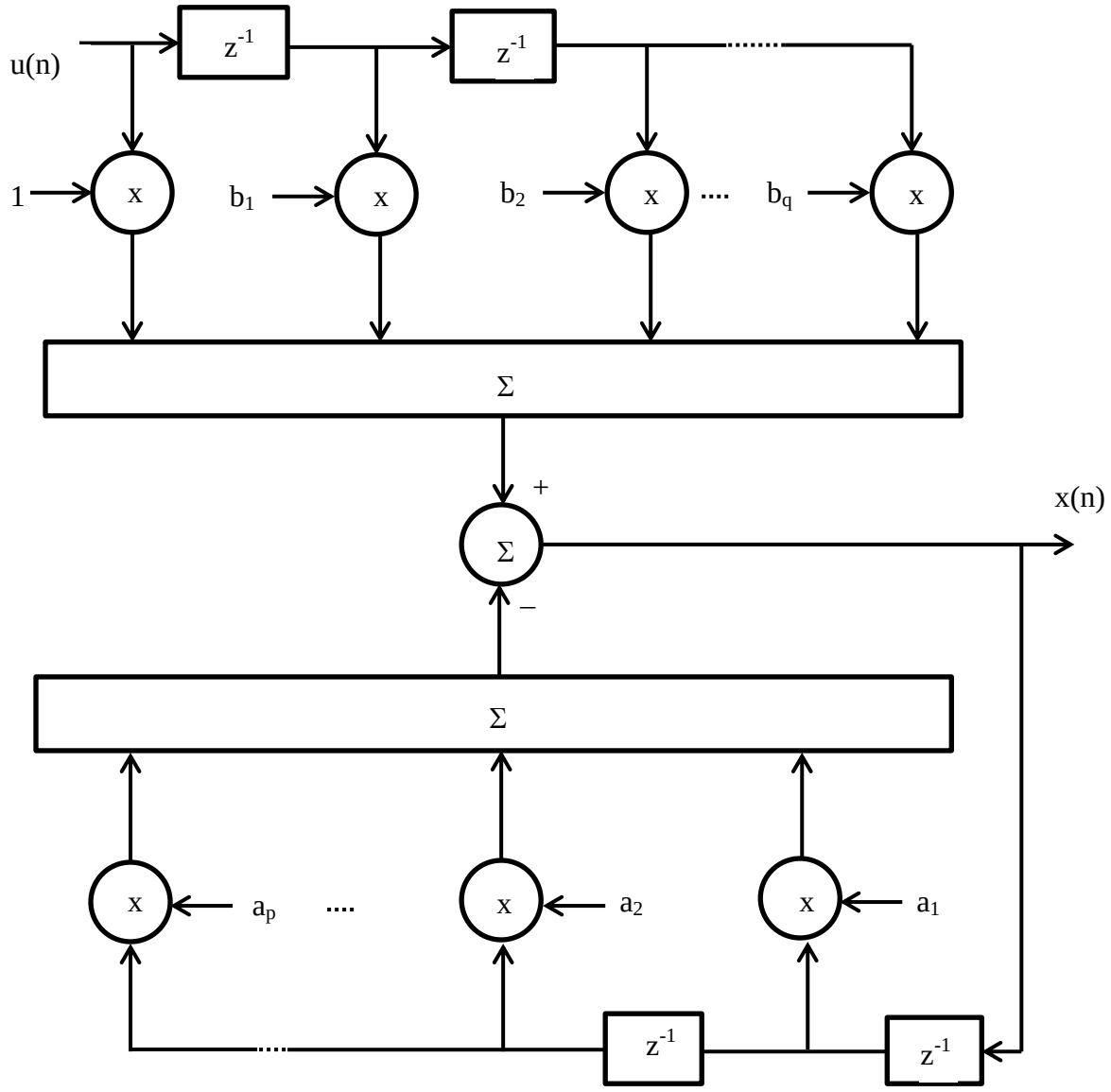

Figure 1 : Modèle autorégressif à moyenne ajustée d'un processus aléatoire.

La transformée en z de la fonction d'autocorrélation de la sortie d'un filtre linéaire $P_{xx}(z)$ est liée à celle de l'entrée $P_{uu}(z)$ par :

$$P_{xx}(z) = H(z)H^*(1/z^*)P_{uu}(z) = \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)}P_{uu}(z) \qquad (3)$$

Lorsque (3) est évaluée sur le cercle unité $z=e^{j2\pi f}$ pour $-1/2 \leq f \leq 1/2$, elle devient la densité spectrale de puissance (psd : power spectral density) $P_{xx}(f)$:

$$P_{ARMA}(f) = P_{xx}(f) = \sigma^2 \left| \frac{B(f)}{A(f)} \right|^2 \qquad (4)$$

$A(f)=A(e^{j2\pi f})$ et $B(f)=B(e^{j2\pi f})$

Sans perte de généralité, on suppose que $a_0=1$ et $b_0=1$ puisque tout gain du filtre peut être incorporé dans σ^2.

Si tous les coefficients a_k sont nuls sauf a_0 alors, il vient :

$$x(n) = \sum_{k=0}^{q} b_k u(n-k) \qquad (5)$$

Le processus est un processus strictement MA d'ordre q et

$$P_{MA}(f) = \sigma^2 |B(f)|^2$$

Ce modèle est parfois appelé modèle tout-zéro (Figure 2). Il est dénoté par MA(q).

Si tous les coefficients b_k sont nuls sauf $b_0=1$ alors, il vient :

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + u(n) \qquad (6)$$

Le processus est un processus strictement AR d'ordre *p*. Avec ce modèle, la valeur courante du processus est exprimée comme une somme pondérée des valeurs passées plus un terme de bruit. La densité spectrale de puissance est :

$$P_{AR}(f) = \frac{\sigma^2}{|A(f)|^2}$$

Ce modèle est parfois appelé modèle tout-pôle (Figure 3). Il est dénoté par AR(*p*).

2.2. Relations des paramètres des modèles à l'autocorrélation

Il est désirable d'avoir une relation explicite entre les paramètres du modèle et la fonction d'autocorrélation. Ces relations sont obtenues en prenant la TZ inverse de $P_{xx}(z)$. Ces relations peuvent aussi être obtenues directement à partir des modèles des séries chronologiques.

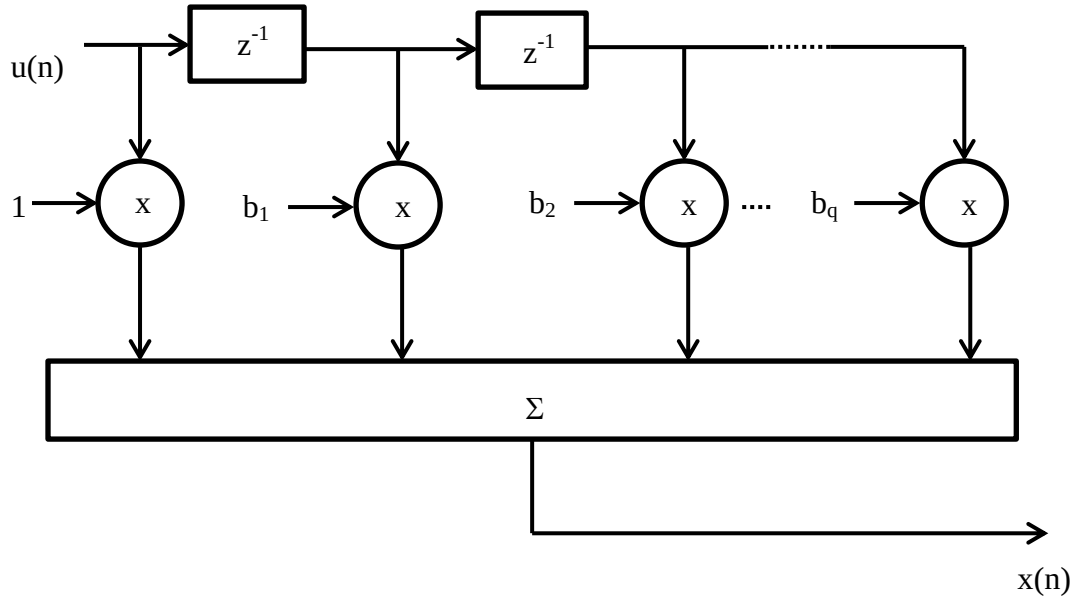

Figure 2 : modèle MA d'un processus aléatoire

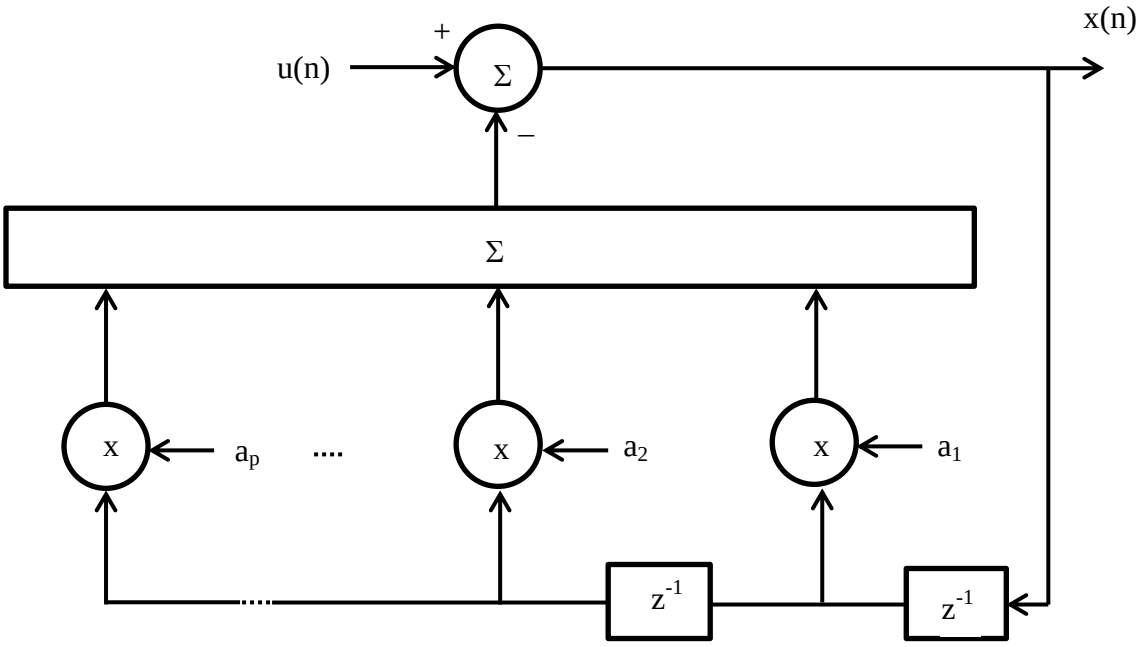

Figure 3 : Modèle AR d'un processus aléatoire.

A partir de (3), on a :

$$P_{xx}(z)A(z) = \frac{B^*(1/z^*)}{A^*(1/z^*)} B(z)\sigma^2$$
$$= H^*(1/z^*)B(z)\sigma^2 \qquad (7)$$

où il est supposé que le bruit u(n) est blanc. En prenant la TZ inverse de (7), on obtient :

$$TZ^{-1}\left[P_{xx}(z)A(z)\right] = r_{xx}(k) * a_k = \sum_{l=0}^{p} a_l \, r_{xx}(k-l)$$

$$TZ^{-1}\left[H^*(1/z^*)B(z)\sigma^2\right]=\sigma^2\sum_{l=0}^{q}b_l\,g(k-l)$$

où $g(l)=TZ^{-1}\left[H^*(1/z^*)\right]$

mais $g(l)=h^*(-l)$, soit donc

$$\sum_{l=0}^{p}a_l r_{xx}(k-l)=\sigma^2\sum_{l=0}^{q}b_l h^*(l+k)$$

En utilisant la causalité de H(z), h(k)=0 pour k<0, et le résultat final pour un processus ARMA devient :

$$r_{xx}(k)=\begin{cases}-\sum_{l=1}^{p}a_l r_{xx}(k-l)+\sigma^2\sum_{l=0}^{q-k}h^*(l)b_{l+k} & \text{pour } k=0,1,\cdots,q\\ -\sum_{l=1}^{p}a_l r_{xx}(k-l) & \text{pour } k\geq q+1\end{cases} \tag{8}$$

Notons que la relation entre les paramètres d'un processus ARMA et la fonction d'autocorrélation est non linéaire. Etant donné la fonction d'autocorrélation, on doit résoudre un ensemble d'équations non linéaires pour trouver les paramètres du modèle. Ceci est dû au terme $\sum_{l=0}^{q-k}h^*(l)b_{l+k}$. Le résultat pour un modèle AR est obtenu en posant $b_l=\delta_l$, ce qui donne :

$$r_{xx}(k)=-\sum_{l=1}^{p}a_l r_{xx}(k-l)+\sigma^2 h^*(-k)$$

Puisque $h^*(-k)=0$ pour k>0 et

$$h^*(0)=\left[\lim_{z\to\infty}H(z)\right]^*=1$$

Il s'en suit que

$$r_{xx}(k)=\begin{cases}-\sum_{l=1}^{p}a_l r_{xx}(k-l) & k\geq 1\\ -\sum_{l=1}^{p}a_l r_{xx}(-l)+\sigma^2 & k=0\end{cases} \tag{9}$$

Les équations (9) sont appelées équations de Yule-Walker. Elles définissent une relation non linéaire entre les paramètres du processus AR et la fonction d'autocorrélation. Cependant, étant donné la fonction d'autocorrélation, on peut déterminer les paramètres AR en résolvant un ensemble d'équations linéaires. Les équations de Yule-Walker peuvent être exprimées sous forme matricielle comme :

$$\underbrace{\begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & \cdots & r_{xx}(-(p-1)) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(-(p-2)) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}((p-1)) & r_{xx}((p-2)) & \cdots & r_{xx}(0) \end{bmatrix}}_{\boldsymbol{R}_{xx}} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} r_{xx}(1) \\ r_{xx}(2) \\ \vdots \\ r_{xx}(p) \end{bmatrix} \tag{10}$$

La matrice R_{xx} est hermitienne ($\boldsymbol{R}_{xx}^{H} = \boldsymbol{R}_{xx}$) et elle Toeplitz puisque les éléments le long d'une diagonale sont égaux. De même, la matrice est définie positive si x(n) ne consiste pas en (p-1) sinusoïdes pures. L'équation (9) peut être résolue efficacement en utilisant l'algorithme de Levinson qui nécessite $O(p^2)$ opérations.

L'équation (10) peut être augmentée en incorporant l'équation en σ^2, ce qui donne :

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & \cdots & r_{xx}(-p) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(-(p-1)) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(p) & r_{xx}((p-1)) & \cdots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a1 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{11}$$

3. Propriétés des processus AR

3.1. Prédiction linéaire des processus AR

La densité spectrale de puissance d'un processus AR dépend des paramètres AR, les propriétés des processus AR sont importantes pour l'interprétation de l'estimateur AR de la densité spectrale de puissance. On suppose que $x(n)$ est un processus AR.

Le problème de la prédiction linéaire est de prédire l'échantillon non observé en se basant sur l'ensemble des données observées $\{x(n-1), x(n-2), \ldots, x(n-p)\}$, c-a-d les p échantillons passés.

Considérons le prédicteur formé à partir de la combinaison linéaire des p échantillons passés :

$$\hat{x}(n) = -\sum_{k=1}^{p} \alpha_k x(n-k) \tag{12}$$

Les coefficients de prédiction $\{\alpha_1, \alpha_2, \ldots, \alpha_p\}$sont choisis de façon à minimiser la puissance de l'erreur de prédiction $e(n)$:

$$p = E\left[\left|e(n)\right|^2\right] = E\left[\left|x(n) - \hat{x}(n)\right|^2\right] \tag{13}$$

Les coefficients de prédiction sont indépendantes de la valeur de n puisque $x(n)$ est supposé stationnaire au sens large. La puissance de l'erreur de prédiction peut être minimisée en posant la dérivée de (13) par rapport à α_k égale à zéro, ce qui donne :

$$E\left[x^*(n-k)\left(x(n) - \hat{x}(n)\right)\right] = 0, \quad k = 1, 2, \cdots, p \tag{14}$$

ou

$$r_{xx}(k) = -\sum_{l=1}^{p} \alpha_l r_{xx}(k-l)\ , \qquad k = 1, 2, \cdots, p \tag{15}$$

Les équations données par (15) sont appelées équations de Wiener-Hopf dans la théorie de la prédiction linéaire.

La puissance de l'erreur de prédiction minimale peut être calculée en utilisant (15), ce qui donne :

$$p_{\min} = E\left[x^*(n)\left(x(n) - \hat{x}(n)\right)\right] = r_{xx}(0) + \sum_{k=1}^{p} \alpha_k r_{xx}(-k) \tag{16}$$

Puisque ces équations sont identiques aux équations de Yule-Walker pour les processus AR, on déduit alors que $\alpha_k = a_k$ pour k=1, 2, …, p et $p_{\min} = \sigma^2$. Les coefficients de prédiction linéaire optimale sont identiques aux paramètres AR et la puissance de l'erreur de prédiction minimale est égale à la variance du bruit d'excitation. Ceci est correct uniquement si l'ordre du processus AR et l'ordre du prédicteur linéaire sont identiques.

En plus, l'erreur de prédiction $e(n)$ n'est autre que $u(n)$ puisque :

$$e(n) = x(n) - \hat{x}(n) = x(n) - \left[-\sum_{k=1}^{p} \alpha_k x(n-k)\right] = x(n) + \sum_{k=1}^{p} a_k x(n-k) = u(n) \tag{17}$$

Les coefficients de prédiction $\{\alpha_1, \alpha_2, \ldots, \alpha_p\}$sont choisis de façon à minimiser la puissance de l'erreur de prédiction $e(n)$.

Soit $\boldsymbol{x}(n-1) = \left[x(n-1)\ x(n-2) \cdots x(n-p)\right]^T$

et soit $\mathbf{R}$=E[$\boldsymbol{x}$(n-1)$\boldsymbol{x}^{\mathrm{H}}$($n$-1)] la matrice de corrélation de dimension pxp

$$\boldsymbol{R} = \begin{bmatrix} r(0) & r(-1) & \cdots & r(-(p-1)) \\ r(1) & r(0) & \cdots & r(-(p-2)) \\ \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & \cdots & r(0) \end{bmatrix}$$

où $r(k)$ est la fonction d'autocorrélation du processus d'entrée au retard k=0, 1, ..., p-1, et soit $\boldsymbol{r}$ le vecteur de cross corrélation entre les p échantillons passés $x(n$-1), $x(n$-2), …, $x(n$-$p)$ et l'échantillon prédit $x(n)$:

$$\boldsymbol{r} = E\left[\boldsymbol{x}(n-1)x^*(n)\right] = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{bmatrix}$$

Les coefficients du prédicteur optimal $\boldsymbol{\alpha}$=$[\alpha_1, \alpha_2, \ldots, \alpha_p]^{\mathrm{T}}$ sont obtenus en résolvant

$$\boldsymbol{R\alpha} = -\boldsymbol{r} \tag{18}$$

et la puissance de l'erreur de prédiction minimale est

$$p_{\min} = r(0) + \boldsymbol{r}^H \boldsymbol{\alpha} \tag{19}$$

Le prédicteur linéaire d'ordre *p* peut être interprété en termes de filtrage comme représenté sur la figure 4. Le filtre de l'erreur de prédiction optimale qui produit l'erreur de prédiction à la sortie pour *x*(*n*) à l'entrée est le filtre inverse $H(z) = 1 + \sum_{k=1}^{p} a_k z^{-k}$.

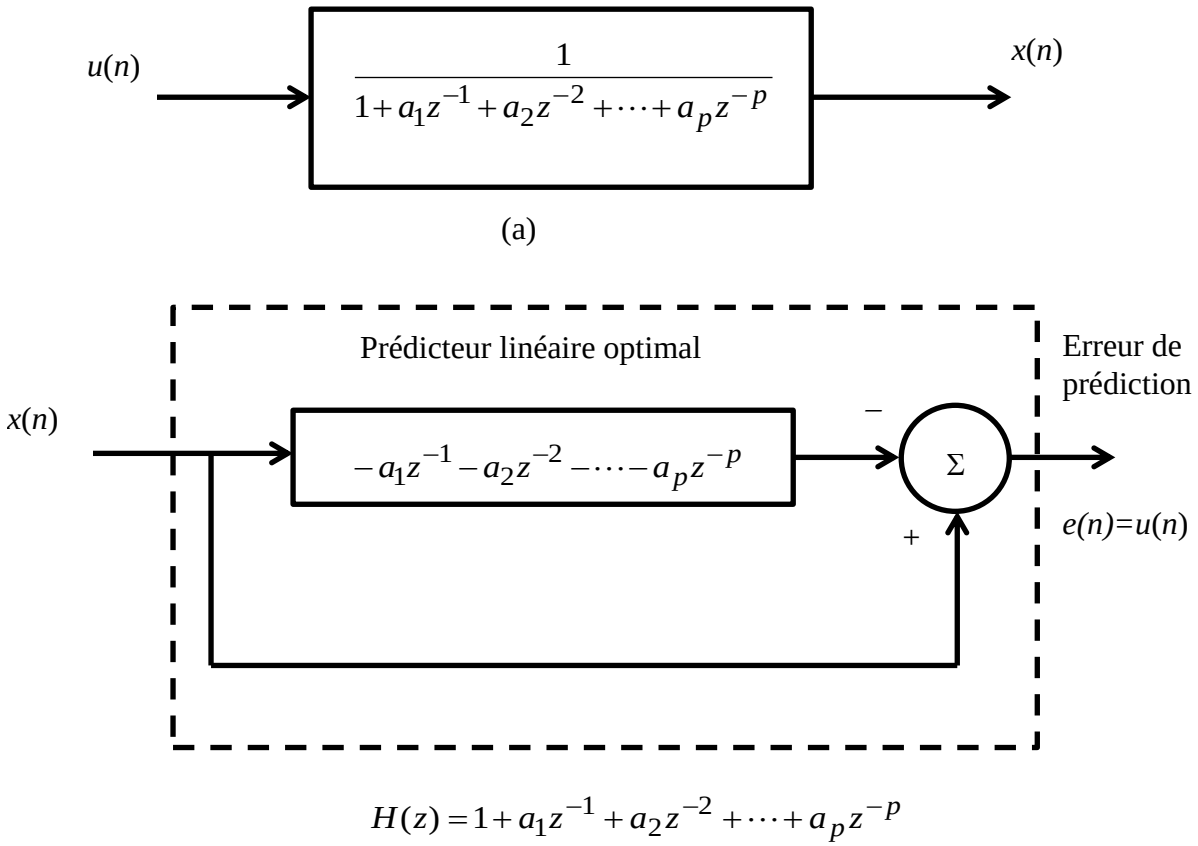

Figure 4 : Prédicteur linéaire d'ordre comme filtre.

La série chronologique de l'erreur de prédiction est u(n) et donc le filtre de l'erreur de prédiction peut être vu comme un filtre de blanchiment.

Le processus AR(*p*) peut être défini par :

$$x(n) = \hat{x}(n) + u(n)$$

où $\hat{x}(n) = -\sum_{k=1}^{p} a_k x(n-k)$ est la prédiction linéaire optimale à un pas basée sur les *p* échantillons passés.

Cette forme prédiction linéaire est dite avant ou progressive. Dans le sens où on utilise les échantillons passés x(n-1), x(n-2), …, x(n-p) pour prédire l'échantillon courant x(n).

3.2. Prédiction linéaire rétrograde ou arrière

La forme de la prédiction linéaire considérée dans la section précédente est dite avant ou progressive. On peut aussi opérer dans la direction arrière. On utilise les échantillons $x(n)$, $x(n\text{-}1)$, …, $x(n\text{-}p\text{+}1)$ pour prédire l'échantillon $x(n\text{-}p)$. Cette forme de prédiction linéaire est dite arrière ou rétrograde.

$$\hat{x}(n-p) = -\sum_{k=1}^{p} \beta_k x(n-k+1) \tag{20}$$

L'erreur de prédiction arrière notée $e_b(n)$ est donnée par :

$$e_b(n) = x(n-p) - \hat{x}(n-p) \tag{21}$$

La puissance de l'erreur de prédiction rétrograde est donnée par

$$p = E\left[\left|e_b(n)\right|^2\right] \tag{22}$$

Les coefficients de prédiction $\{\beta_1, \beta_2, \ldots, \beta_p\}$sont choisis de façon à minimiser la puissance de l'erreur de prédiction rétrograde $e_b(n)$.

Soit $\boldsymbol{x}(n) = \left[x(n)\ x(n-1) \cdots x(n-p+1)\right]^T$ et soit $\boldsymbol{R}$=E[$\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{H}}(n)$] la matrice de corrélation de dimension pxp et soit $\boldsymbol{r}^{B*}$ le vecteur de cross corrélation entre les p échantillons passés et l'échantillon prédit $x(n\text{-}p)$

$$\boldsymbol{r}^{B*} = E\left[\boldsymbol{x}(n)x^H(n-p)\right] = \begin{bmatrix} r(-p) \\ r(-(p-1)) \\ \vdots \\ r(-1) \end{bmatrix}$$

Les coefficients du prédicteur optimal rétrograde sont obtenus en résolvant

$$\boldsymbol{R}\boldsymbol{\beta} = -\boldsymbol{r}^{\mathrm{B}*} \tag{23}$$

et la puissance de l'erreur de prédiction minimale est

$$p_{\min} = r(0) + \boldsymbol{r}^{BT}\boldsymbol{\beta} \tag{24}$$

3.3. Relation entre les prédicteurs direct et rétrograde

En comparant les deux ensembles des équations de Wiener-Hopf (18) et (23) correspondantes aux prédictions linéaires directe et rétrograde, respectivement, on voit que le vecteur du membre de droite de (23) diffère de celui de (18) en deux égards : (1) ses éléments sont

réarrangés en arrière et (2) ils sont complexes conjugués. Pour enlever la première différence, on doit inverser l'ordre des éléments dans lequel les éléments du vecteur du membre de droite de (23) sont arrangés. Cette opération a pour effet de remplacer le membre de gauche de (23) par $\boldsymbol{R}^T\beta^B$, où $\boldsymbol{R}^T$ est la matrice transposée de la matrice de corrélation $\boldsymbol{R}$ et β^B est la version arrière du vecteurβ. On peut donc écrire :

$$\boldsymbol{R}^T\beta^B=-\boldsymbol{r}^* \tag{25}$$

Pour enlever la deuxième différence, on prend le complexe conjugué des deux membres de l'équation (25), ce qui donne :

$$\boldsymbol{R}^H\beta^{B*}=-\boldsymbol{r} \tag{26}$$

Puisque la matrice de corrélation est hermitienne, $\boldsymbol{R}^H=\boldsymbol{R}$, on peut donc reformuler les équations de Wiener-Hopf pour la prédiction rétrograde comme :

$$\boldsymbol{R}\beta^{B*}=-\boldsymbol{r} \tag{27}$$

Maintenant, on peut comparer les équations (18) et (27) et donc déduire la relation fondamentale entre les coefficients du prédicteur optimal rétrograde et ceux du prédicteur direct correspondant :

$$\beta^{B*}=\boldsymbol{\alpha} \tag{28}$$

L'équation (28) montre qu'on peut modifier le prédicteur rétrograde en un prédicteur direct en inversant la séquence dans laquelle les coefficients sont positionnés en en prenant aussi leurs complexes conjugués.

On montre aussi que les puissances des erreurs de prédiction directe et rétrograde sont identiques. Pour cela, on observe que le produit $\boldsymbol{r}^{BT}\beta$ **et** $\boldsymbol{r}^T\beta^H$ sont les mêmes. Donc, on peut réécrire l'équation (24) comme :

$$p_{\min} = r(0) + \boldsymbol{r}^T\boldsymbol{\beta}^B \tag{29}$$

En prenant le complexe conjugué des deux membres de (29) et en sachant que $p_{\min}$ et $r(0)$ ne sont pas affectés par cette opération puisqu'ils sont des scalaires à valeurs réelles, on obtient :

$$p_{\min} = r(0) + \boldsymbol{r}^H\boldsymbol{\beta}^{B*} \tag{30}$$

En comparant ce résultat avec l'équation (19) et en utilisant l'équivalence de l'équation (28), on déduit que les puissances des erreurs de prédiction directe et arrière ont exactement les mêmes valeurs.

3.4. Filtre de l'erreur de prédiction rétrograde

L'erreur de prédiction rétrograde est donnée par :

$$e_b(n) = x(n-p) - \hat{x}(n-p) = x(n-p) + \sum_{k=1}^{p} \beta_k x(n-k+1) \tag{31}$$

Définissons les coefficients du filtre de l'erreur de prédiction en fonction des coefficients du prédicteur comme :

$$c_{p,k} = \begin{cases} \beta_{k+1} & k = 0, 1, \cdots, p-1 \\ 1 & k = p \end{cases} \tag{32}$$

L'équation (31) peut être réécrite comme :

$$e_b(n) = \sum_{k=1}^{p} c_{p,k} x(n-k) \tag{33}$$

L'équation (28) exprime le vecteur des coefficients du prédicteur rétrograde en fonction du vecteur des coefficients du prédicteur direct. On peut exprimer cette relation sous forme scalaire comme :

$$\beta^*_{p-k+1} = \alpha_k, \quad k = 1, 2, \cdots, p$$

Ou encore

$$\beta_k = \alpha^*_{p-k+1}, \quad k = 1, 2, \cdots, p \tag{34}$$

Donc, en substituant (34) dans (32), il vient :

$$c_{p,k} = \begin{cases} \alpha^*_{p-k} & k = 0, 1, \cdots, p-1 \\ 1 & k = p \end{cases} \tag{35}$$

Ou encore, en utilisant la relation entre les coefficients de prédiction et les paramètres AR, l'équation (33) s'écrit comme :

$$e_b(n) = \sum_{k=1}^{p} a^*_{p,p-k} x(n-k) \tag{36}$$

On peut combiner les équations de Wiener-Hopf données par (27) et l'équation définissant la puissance de l'erreur de prédiction minimale en une seule équation :

$$\begin{bmatrix} \boldsymbol{R} & \boldsymbol{r}^{B*} \\ \boldsymbol{r}^{BT} & r(0) \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ p_{\min} \end{bmatrix} \tag{37}$$

où **0** est le vecteur nul de dimension px1 et $\boldsymbol{R}$ la matrice de corrélation du vecteur $\boldsymbol{x}(n)$ des p échantillons d'entrée.

On peut aussi exprimer la relation matricielle donnée par (37) comme un système de p+1 équations simultanées

$$\sum_{l=0}^{p} a_{p,p-l}^{*} r(l-i) = \begin{cases} 0 & i = 0, 1, \cdots, p-1 \\ p_{\min} & i = p \end{cases} \tag{38}$$

Les équations (37) ou (38) sont appelées les équations de Wiener-Hopf augmentées du filtre de l'erreur de prédiction rétrograde d'ordre *p*.

4. Algorithme de Levinson-Durbin

La méthode de calcul des coefficients du filtre de prédiction de l'erreur en résolvant les équations de Wiener-Hopf augmentées est récursive et exploite la structure de Toeplitz de la matrice de corrélation. Elle est appelée algorithme de Levinson-Durbin. Elle était utilisée pour la première fois par Levinson (1947) puis reformulée indépendamment par Durbin (1960).

La procédure utilise la solution des équations de Wiener-Hopf augmentées pour un filtre de l'erreur de prédiction d'ordre *m*-1 pour calculer la solution correrspondante pour un filtre de l'erreur de prédiction d'ordre *m*.

Résumé de l'algorithme de Levinson-Durbin

L'algorithme récursif est initialisé par

$$a_{1,1} = -\frac{r_{xx}(1)}{r_{xx}(0)}$$

$$p_1 = \left(1 - \left|a_{1,1}\right|^2\right) r_{xx}(0)$$

Pour *m*=2, 3, …, *M*

$$\Gamma_m = a_{m,m} = -\frac{r_{xx}(m) + \sum_{i=1}^{m-1} a_{m-1,i} r_{xx}(m-i)}{p_{m-1}} \tag{39}$$

$$a_{m,k} = a_{m-1,k} + \Gamma_m a_{m-1,m-k}^{*}, \; k=1, 2, \ldots, m\text{-}1 \tag{40}$$

$$p_m = \left(1 - \left|\Gamma_m\right|^2\right) p_{m-1} \tag{41}$$

5. Estimation spectrale autorégressive

5.1. Méthode d'autocorrélation

Dans la méthode d'autocorrélation, les paramètres AR sont estimés en minimisant une estimation de la puissance de l'erreur de prédiction :

$$\hat{p} = \frac{1}{N} \sum_{n=-\infty}^{+\infty} \left| x(n) + \sum_{k=1}^{p} a_k x(n-k) \right|^2 \tag{42}$$

Les échantillons du processus $x(n)$ non observés (en dehors de l'intervalle $0 \le n \le N\text{-}1$) sont supposés nuls. L'estimation de la puissance de l'erreur de prédiction est minimisée en posant égale à zéro la dérivée de par rapport à la partie réelle et la partie imaginaire de a_k. Ceci peut être réalisé en utilisant le gradient complexe pour obtenir :

$$\frac{1}{N}\sum_{n=-\infty}^{+\infty}\left(x(n)+\sum_{k=1}^{p}a_k x(n-k)\right)x^*(n-l)=0, \qquad l=1,2,\cdots,p \tag{43}$$

Sous forme matricielle, l'ensemble des équations devient :

$$\begin{bmatrix} \hat{r}_{xx}(0) & \hat{r}_{xx}(-1) & \cdots & \hat{r}_{xx}(-(p-1)) \\ \hat{r}_{xx}(1) & \hat{r}_{xx}(0) & \cdots & \hat{r}_{xx}(-(p-2)) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_{xx}(p-1) & \hat{r}_{xx}(p-2) & \cdots & \hat{r}_{xx}(0) \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} \hat{r}_{xx}(1) \\ \hat{r}_{xx}(2) \\ \vdots \\ \hat{r}_{xx}(p) \end{bmatrix} \tag{44}$$

où

$$\hat{r}_{xx}(k)=\begin{cases} \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1+k}x^*(n)x(n-k) & k=0,1,\cdots,p \\ \hat{r}_{xx}^*(-k) & k=-(p-1),-(p-2),\cdots,-1 \end{cases} \tag{45}$$

Qui est l'estimateur biaisée de l'autocorrélation.

La matrice dans (44) est Hermitienne et Toeplitz et définie positive. L'appellation alternative méthode de Yule-Walker est due à l'équivalence de la méthode d'autocorrélation avec les équations de Yule-Walker de avec un estimateur biaisé de la fonction d'autocorrélation.

L'algorithme de Levinson-Durbin peut être utilisé pour résoudre les équations et les pôles estimés sont garantis d'être à l'intérieur du cercle unité.

L'estimation de la variance du bruit blanc σ^2 est $\hat{p}_{\min}$:

$$\begin{aligned}\hat{\sigma}^2=\hat{p}_{\min}&=\frac{1}{N}\sum_{n=-\infty}^{+\infty}\left|x(n)+\sum_{k=1}^{p}a_k x(n-k)\right|^2 \\ &=\frac{1}{N}\sum_{n=-\infty}^{+\infty}\left[\left(x(n)+\sum_{k=1}^{p}a_k x(n-k)\right)x^*(n)+\left(x(n)+\sum_{k=1}^{p}a_k x(n-k)\right)\sum_{l=1}^{p}a_l^* x^*(n-l)\right]\end{aligned}$$

A partir de (68), le second terme de la somme sur n est nul, ce qui donne

$$\hat{\sigma}^2=\hat{r}_{xx}(0)+\sum_{k=1}^{p}a_k\hat{r}_{xx}(-k) \tag{46}$$

$\hat{\sigma}^2$ peut-être aussi écrite comme :

$$\hat{\sigma}^2 = \hat{r}_{xx}(0)\prod_{i=1}^{p}\left(1-\left|\Gamma_i\right|^2\right) \tag{47}$$

5.2. Méthode de la covariance

Dans la méthode de covariance, les paramètres AR sont estimés en minimisant une estimation de la puissance de l'erreur de prédiction :

$$\hat{p} = \frac{1}{N-p}\sum_{n=p}^{N-1}\left|x(n)+\sum_{k=1}^{p}a_k x(n-k)\right|^2 \tag{48}$$

Notons que la seule différence entre la méthode d'autocorrélation et la méthode de covariance est l'intervalle de sommation dans l'estimation de la puissance de l'erreur de prédiction. La minimisation de (48) donne l'ensemble des équations suivantes :

$$\begin{bmatrix} c_{xx}(1,1) & c_{xx}(1,2) & \cdots & c_{xx}(1,p) \\ c_{xx}(2,1) & c_{xx}(2,2) & \cdots & c_{xx}(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ c_{xx}(p,1) & c_{xx}(p,2) & \cdots & c_{xx}(p,p) \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} c_{xx}(1,0) \\ c_{xx}(2,0) \\ \vdots \\ c_{xx}(p,0) \end{bmatrix} \tag{74}$$

où

$$c_{xx}(j,k) = \frac{1}{N-p}\sum_{n=p}^{N-1}x^*(n-j)x(n-k)$$

La variance du bruit blanc est estimée par :

$$\hat{\sigma}^2 = \hat{p}_{\min} = c_{xx}(0,0) + \sum_{k=1}^{p}a_k c_{xx}(0,k) \tag{49}$$

5.3. Méthode de Burg

Contrairement aux méthodes d'autocorrélation et de la covariance qui estiment les paramètres AR directement, la méthode de Burg estime les coefficients de réflexion et utilise l'algorithme de Levinson-Durbin pour obtenir une estimation des paramètres AR. Si les estimations des coefficients de réflexion $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_p\}$ sont disponibles, les paramètres AR peuvent être estimés comme :

$$\hat{r}_{xx}(0) = \frac{1}{N}\sum_{n=0}^{N-1}\left|x(n)\right|^2$$

$$a_1 = \Gamma_1$$

$$\hat{p}_1 = \left(1-\left|\Gamma_1\right|^2\right)r_{xx}(0)$$

Pour k=2, 3, …, p

$$a_{k,i} = \begin{cases} a_{k-1,i} + \Gamma_k a^*_{k-1,k-i}, & i = 1,2,\cdots,k-1 \\ \Gamma_k, & i = k \end{cases} \tag{50}$$

$$\hat{p}_k = \left(1 - \left|a_{k,k}\right|^2\right)\hat{p}_{k-1} \tag{51}$$

Les estimations des paramètres AR dont données par $\{a_{p,1}, a_{p,2}, a_{p,p}\}$ et l'estimation de la variance du bruit blanc est $\hat{p}_k$.

Burg propose pour estimer Γ_k de minimiser la moyenne des estimations des erreurs de prédiction directe et rétrograde. Pour obtenir Γ_k, on minimise :

$$\hat{p}_k = \frac{1}{2}\left(\hat{p}_k^f + \hat{p}_k^b\right) \tag{52}$$

où

$$\hat{p}_k^f = \frac{1}{N-k}\sum_{n=k}^{N-1}\left|x(n) + \sum_{i=1}^{k} a_{k,i}x(n-i)\right|^2 \tag{53}$$

$$\hat{p}_k^b = \frac{1}{N-k}\sum_{n=0}^{N-1-k}\left|x(n) + \sum_{i=1}^{k} a_{k,i}^* x(n+i)\right|^2 \tag{54}$$

et

$$a_{k,i} = \begin{cases} a_{k-1,i} + \Gamma_k a_{k-1,k-i}^*, & i = 1,2,\cdots,k-1 \\ \Gamma_k, & i = k \end{cases} \tag{55}$$

Définissons les erreurs de prédiction directe et rétrograde par :

$$\hat{e}_k^f(n) = x(n) + \sum_{i=1}^{k} a_{k,i}x(n-i) \tag{56}$$

$$\hat{e}_k^b(n) = x(n) + \sum_{i=1}^{k} a_{k,i}^* x(n-k+i) \tag{57}$$

La puissance de l'erreur de prédiction directe devient :

$$\hat{p}_k^f = \frac{1}{N-k}\sum_{n=k}^{N-1}\left|\hat{e}_k^f(n)\right|^2 \tag{58}$$

$$\hat{p}_k^b = \frac{1}{N-k}\sum_{n=k}^{N-1}\left|\hat{e}_k^b(n)\right|^2 \tag{59}$$

En appliquant les équations du filtre de l'erreur de prédiction en treillis, on peut écrire :

$$\hat{e}_k^f(n) = \hat{e}_{k-1}^f(n) + \Gamma_k \hat{e}_{k-1}^b(n-1) \tag{60}$$

$$\hat{e}_k^b(n) = \hat{e}_{k-1}^b(n-1) + \Gamma_k^* \hat{e}_{k-1}^f(n) \tag{61}$$

avec

$$\hat{e}_0^f(n) = \hat{e}_0^b(n) = x(n)$$

En substituant ces relations dans (60) et (61) puis dans (52), on obtient :

$$\hat{p}_k = \frac{1}{2(N-k)} \sum \left[\left| \hat{e}_{k-1}^f(n) + \Gamma_k \hat{e}_{k-1}^b(n-1) \right|^2 + \left| \hat{e}_{k-1}^b(n-1) + \Gamma_k^* \hat{e}_{k-1}^f(n) \right|^2 \right] \qquad (62)$$

En minimisant $\hat{p}_k$ par rapport à Γ_k, on obtient :

$$\frac{\partial \hat{p}_k}{\partial \Gamma_k} = \frac{1}{N-k} \sum_{n=k}^{N-1} \left\{ \left(\hat{e}_{k-1}^f(n) + \Gamma_k \hat{e}_{k-1}^b(n-1) \right) \hat{e}_{k-1}^b(n-1)^* + \left(\hat{e}_{k-1}^b(n-1) + \Gamma_k \hat{e}_{k-1}^f(n)^* \right) \hat{e}_{k-1}^f(n) \right\} = 0$$

Ce qui donne

$$\Gamma_k = \frac{-2 \sum_{n=k}^{N-1} \hat{e}_{k-1}^f(n) \hat{e}_{k-1}^b(n-1)^*}{\sum_{n=k}^{N-1} \left(\left| \hat{e}_{k-1}^f(n) \right|^2 + \left| \hat{e}_{k-1}^b(n-1) \right|^2 \right)}$$

Résumé de la méthode de Burg

$$\hat{r}_{xx}(0) = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2$$

$$\hat{p}_0 = \hat{r}_{xx}(0)$$

$$\hat{e}_0^f(n) = x(n), \quad n = 1,2,\cdots,N-1$$

$$\hat{e}_0^b(n) = x(n), \quad n = 0,1,\cdots,N-2$$

Pour k=1, 2, …, p

$$\Gamma_k = \frac{-2 \sum_{n=k}^{N-1} \hat{e}_{k-1}^f(n) \hat{e}_{k-1}^b(n-1)^*}{\sum_{n=k}^{N-1} \left(\left| \hat{e}_{k-1}^f(n) \right|^2 + \left| \hat{e}_{k-1}^b(n-1) \right|^2 \right)}$$

$$\hat{p}_k = \left(1 - |\Gamma_k|^2 \right) \hat{p}_{k-1}$$

$$a_{k,i} = \begin{cases} a_{k-1,i} + \Gamma_k a_{k-1,k-i}^*, & i = 1,2,\cdots,k-1 \\ \Gamma_k, & i = k \end{cases}$$

$$\hat{e}_k^f(n) = \hat{e}_{k-1}^f(n) + \Gamma_k \hat{e}_{k-1}^b(n-1), \quad n = k+1, k+2, \cdots, N\text{-}1$$

$$\hat{e}_k^b(n) = \hat{e}_{k-1}^b(n-1) + \Gamma_k^* \hat{e}_{k-1}^f(n), \quad n = k, k+1, \cdots, N\text{-}2$$

6. Filtrage adaptatif

Dans de nombreuses applications, le filtrage ne peut être réalisé avec succès en utilisant des filtres numériques avec des coefficients fixes car soit nous n'avons pas suffisamment d'informations pour concevoir un filtre numérique avec des coefficients fixes ou les critères de conception changent au cours du fonctionnement du filtre. La plupart de ces applications peuvent être résolues avec succès en utilisant un type spécial de filtres « intelligents » connus collectivement sous le nom de filtres adaptatifs. La caractéristique des filtres adaptatifs est qu'ils peuvent modifier leur réponse pour améliorer leurs performances pendant le fonctionnement sans aucune intervention de l'utilisateur.

Un filtre adaptatif est un filtre dont les coefficients (c'est-à-dire les valeurs de réponse impulsionnelle, {h(k)}) sont mis à jour à chaque instant. Les coefficients du filtre sont adaptés au fur et à mesure que le filtre reçoit à son entrée un échantillon afin de fournir une estimation optimale $\hat{x}(n)$ du signal original $x(n)$ à l'instant n.

La figure 5 montre la structure d'un filtre adaptatif.

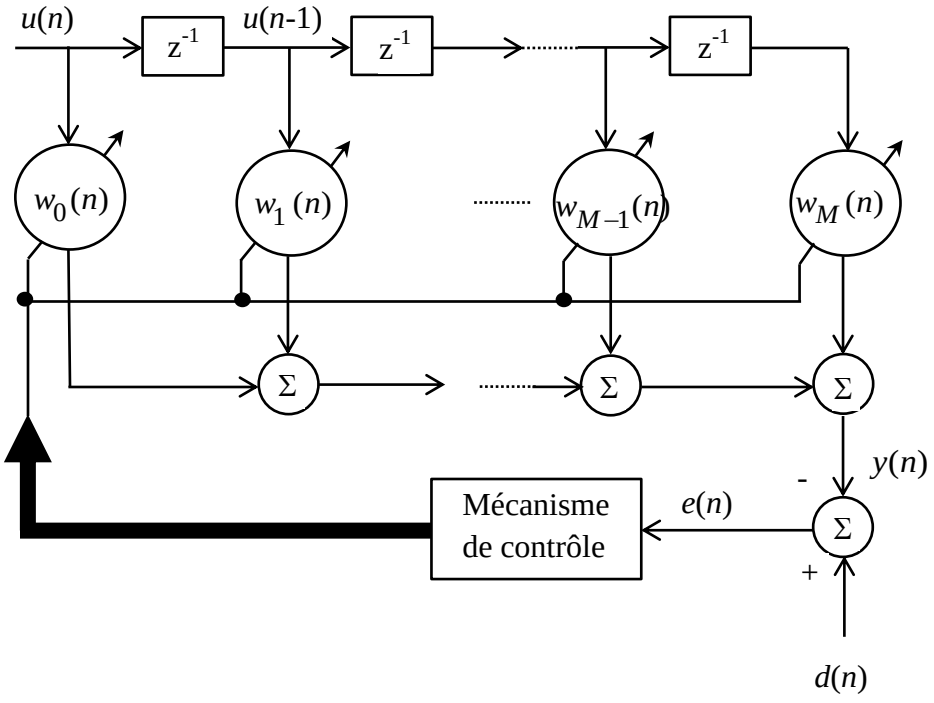

Figure 5: Structure d'un filtre adaptatif

6.1. Algorithme du gradient stochastique (algorithme LMS)

L'algorithme très connue sous le nom de l'algorithme du gradient stochastique (LMS : least mean square) est décrit comme suit :

$$\begin{pmatrix}\text{nouvelle valeur}\\ \text{du vecteur}\\ \text{des coefficients}\end{pmatrix}=\begin{pmatrix}\text{ancienne valeur}\\ \text{du vecteur}\\ \text{des coefficients}\end{pmatrix}+\begin{pmatrix}\text{paramètre}\\ \text{du taux}\\ \text{d'apprentissage}\end{pmatrix}\begin{pmatrix}\text{vecteur}\\ \\ \text{d'entrée}\end{pmatrix}\begin{pmatrix}\text{Signal}\\ \text{erreur}\end{pmatrix}$$

où le signal erreur est défini comme étant la différence entre la réponse désirée et la réponse réelle du filtre produite par le signal d'entrée.

La figure montre la structure de base de l'algorithme LMS.

L'algorithme LMS est simple et il est capable de réaliser des performances satisfaisantes dans les bonnes conditions.

L'algorithme LMS peut être décrit comme suit :

1. Sortie du filtre :
   $y(n)=\hat{\boldsymbol{w}}^T(n)\boldsymbol{u}(n)$ (63)
2. Erreur d'estimation
   $e(n)=d(n)-y(n)$ (64)
3. Adaptation du vecteur des coefficients de pondération
   $\hat{\boldsymbol{w}}(n+1)=\hat{\boldsymbol{w}}(n)+\mu\boldsymbol{u}(n)e\ (n)$ (65)

où $\boldsymbol{u}(n)=[u(n)\ u(n-1)\cdots u(n-M)]^T$ est le vecteur d'entrée du filtre à l'instant n,

$\boldsymbol{y}(n)=[y(n)\ y(n-1)\cdots y(n-M)]^T$ est le vecteur de sortie du filtre à l'instant n

$\boldsymbol{w}(n)=[w_0(n)\ w_1(n)\cdots w_M(n]^T$ est le vecteur des coefficients du filtre à l'instant n

μ est une constant entre 0 et 1 appelée facteur d'oubli. Le paramètre μ contrôle la taille de correction incrémentale appliquée au vecteur des coefficients de pondération d'une itération à l'autre.

La procédure itérative commence avec une valeur initiale $\hat{\boldsymbol{w}}(0)$. Un choix convenable pour cette valeur initiale est le vecteur nul, $\hat{\boldsymbol{w}}(0)=\boldsymbol{0}$.

6.2. Algorithme des moindres carrés récursifs RLS

Le développement de l'algorithme RLS utilise la méthode des moindres carrés et un résultat de base de l'algèbre linéaire connu sous le nom de lemme d'inversion matricielle.

6.2.1. Préliminaires

La fonction coût à minimiser est

$$E(n) = \sum_{i=1}^{n} \beta(n,i)|e(i)|^2 \qquad (66)$$

où β(*n*, *i*) est un facteur de pondération ou facteur d'oubli, *e*(*i*) est la différence entre la réponse désirée *d*(*i*) et la sortie *y*(*i*) produite par le filtre dont les échantillons d'entrée (à l'instant *i*) sont *u*(*i*), *u*(*i*-1), …, *u*(*i*–*M*). L'erreur e(i) est définie par

$$e(i) = d(i) - y(i) = d(i) - \boldsymbol{w}(n)\boldsymbol{u}(i) \qquad (67)$$

où $\boldsymbol{u}(i)$ est le vecteur d'entrée à l'instant *i* défini par

$$\boldsymbol{u}(i) = \begin{bmatrix} u(i) & u(i-1) & \cdots & u(i-M) \end{bmatrix}^T \qquad (68)$$

et $\boldsymbol{w}(n)$ est le vecteur des coefficients de pondération à l'instant n défini par

$$\boldsymbol{w}(n) = \begin{bmatrix} w_0(n) & w_1(n) & \cdots & w_M(n) \end{bmatrix}^T \qquad (69)$$

Les coefficients de pondération du filtre restent fixes durant l'intervalle d'observation $1 \le i \le n$.

Le facteur de pondération β(*n*, *i*) dans l'équation (66) est tel que

$$0 < \beta(n, i) \le 1, \; i=1, 2, \ldots, n \qquad (70)$$

Le facteur de pondération β(*n*, *i*) est utilisé pour assurer que les données dans le passé distant sont « oubliées » pour permettre de suivre les variations statistiques des données observables lorsque le filtre opère dans un environnement non stationnaire. Une forme très utilisée de ce facteur d'oubli est le facteur exponentiel défini par

$$\beta(n,i) = \lambda^{n-i}, \qquad i = 1,2,\cdots,n \qquad (71)$$

où λ est une constante positive proche mais inférieure à 1. Lorsque λ est égal à 1, on a la méthode ordinaire des moindres carrés. L'inverse de 1-λ est une mesure de la mémoire de l'algorithme. Le cas particulier λ=1 correspond à une mémoire infinie. Dans la méthode des moindres carrés à pondération exponentielle, la fonction coût à minimiser est

$$E(n) = \sum_{i=1}^{n} \lambda^{n-i}|e(i)|^2 \qquad (72)$$

La valeur optimale du vecteur de coefficients de pondération $\hat{\boldsymbol{w}}(n)$ pour laquelle la fonction coût E(n) atteint sa valeur minimale est définie par les équations normales exprimées sous forme matricielle par :

$$\boldsymbol{\Phi}(n)\hat{\boldsymbol{w}}(n) = \boldsymbol{\Theta}(n) \qquad (73)$$

La matrice de corrélation $\boldsymbol{\Phi}(n)$ est définie par

$$\boldsymbol{\Phi}(n) = \sum_{i=1}^{n} \lambda^{n-i} \boldsymbol{u}(i)\boldsymbol{u}^{T}(i) \qquad (74)$$

Le vecteur Mx1 $\boldsymbol{\Theta}(n)$ de cross corrélation entre les échantillons d'entrée du filtre et la réponse désirée est définie par

$$\boldsymbol{\Theta}(n) = \sum_{i=1}^{n} \lambda^{n-i} \boldsymbol{u}(i)d(i) \qquad (75)$$

En isolant le terme correspondant à $i=n$ du reste de la sommation du membre de droite de (74), on peut écrire :

$$\boldsymbol{\Phi}(n) = \lambda \left[\sum_{i=1}^{n-1} \lambda^{n-1-i} \boldsymbol{u}(i)\boldsymbol{u}^{T}(i) \right] + \boldsymbol{u}(n)\boldsymbol{u}^{T}(n) \qquad (76)$$

Par définition, l'expression entre crochets du membre de droite de (76) est la matrice de corrélation $\boldsymbol{\Phi}(n\text{-}1)$. Donc, on a la relation récursive suivante pour la mise à jour de la valeur de la matrice de corrélation des échantillons d'entrée:

$$\boldsymbol{\Phi}(n) = \lambda\boldsymbol{\Phi}(n-1) + \boldsymbol{u}(n)\boldsymbol{u}^{H}(n) \qquad (77)$$

où $\boldsymbol{\Phi}(n\text{-}1)$ est l'ancienne valeur de la matrice de corrélation et le produit $\boldsymbol{u}(n)\boldsymbol{u}^{T}(n)$ joue le rôle du terme de correction dans l'opération de mise à jour.

De manière similaire, on peut utiliser l'équation (75) pour développer la relation récursive suivante pour la mise à jour du vecteur de cross corrélation entre les échantillons d'entrée et la réponse désirée :

$$\boldsymbol{\Theta}(n) = \lambda\boldsymbol{\Theta}(n-1) + \boldsymbol{u}(n)d(n) \qquad (78)$$

Pour calculer l'estimation $\hat{\boldsymbol{w}}(n)$ au sens des moindres carrés du vecteur des coefficients de pondération selon (73), on doit déterminer l'inverse de la matrice de corrélation $\boldsymbol{\Phi}(n)$. Cependant, en pratique, on évite d'effectuer une telle opération parce que prend beaucoup de temps, particulièrement lorsque le nombre de coefficients de pondération M est grand. De même, on veut calculer récursivement l'estimation des moindres carrés du vecteur des coefficients de pondération $\hat{\boldsymbol{w}}(n)$ pour n=0, 1, 2, … On peut réaliser ces deux objectifs en utilisant le résultat de l'algèbre matriciel connu comme le lemme d'inversion matricielle. On suppose que les conditions initiales ont été choisies pour assurer la non singularité de la matrice de corrélation $\boldsymbol{\Phi}(n)$.

Lemme d'inversion matricielle

Soit A et B deux matrices MxM définies positives reliées par :

$$\boldsymbol{A} = \boldsymbol{B}^{-1} + \boldsymbol{C}\boldsymbol{D}^{-1}\boldsymbol{C}^{H} \qquad (79)$$

où $\boldsymbol{D}$ est une autre matrice NxN définie positive et $\boldsymbol{C}$ une matrice MxN.

Selon le lemme d'inversion matricielle, on peut exprimer l'inverse de la matrice A comme suit :

$$\boldsymbol{A}^{-1} = \boldsymbol{B} - \boldsymbol{B}\boldsymbol{C}\left(\boldsymbol{D} + \boldsymbol{C}^{H}\boldsymbol{B}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^{H}\boldsymbol{B} \qquad (80)$$

Le lemme d'inversion matricielle est aussi connu dans la littérature sous le nom d'identité de Woodbury.

6.2.2. Algorithme des moindres carrés récursifs à pondération exponentielle

Avec la matrice de corrélation $\boldsymbol{\Phi}(n)$ supposée définie positive et donc non singulière, on peut appliquer le lemme d'inversion matricielle à l'équation récursive (77) en faisant les identités suivantes :

$\boldsymbol{A}=\boldsymbol{\Phi}(n)$

$\boldsymbol{B}=\lambda\boldsymbol{\Phi}(n\text{-}1)$

$\boldsymbol{C}=\mathrm{u}(n)$

$D=\mathrm{I}$

On obtient l'équation récursive suivante pour l'inverse de la matrice de corrélation :

$$\Phi^{-1}(n) = \lambda^{-1}\Phi^{-1}(n-1) - \frac{\lambda^{-2}\boldsymbol{\Phi}^{-1}(n-1)\boldsymbol{u}(n)\boldsymbol{u}^{T}(n)\boldsymbol{\Phi}^{-1}(n-1)}{1+\lambda^{-1}\boldsymbol{u}^{T}(n)\boldsymbol{\Phi}^{-1}(n-1)\boldsymbol{u}(n)} \qquad (81)$$

Pour faciliter le calcul, soit

$$\boldsymbol{P}(n) = \boldsymbol{\Phi}^{-1}(n) \qquad (82)$$

Et

$$\boldsymbol{k}(n) = \frac{\lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{u}(n)}{1+\lambda^{-1}\boldsymbol{u}^{T}(n)\boldsymbol{P}(n-1)\boldsymbol{u}(n)} \qquad (83)$$

En utilisant ces définitions, on peut réécrire l'équation (81) comme suit

$$\boldsymbol{P}(n) = \lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}\boldsymbol{k}(n)\boldsymbol{u}^{T}(n)\boldsymbol{P}(n-1) \qquad (84)$$

Notons que $\boldsymbol{P}(n)$ a les dimensions d'une matrice MxM alors que $\boldsymbol{k}(n)$ a les dimensions d'un vecteur Mx1. On se réfère au vecteur $\boldsymbol{k}(n)$ comme le vecteur gain. L'équation (84) est l'équation de Ricatti pour l'algorithme RLS.

En réarrangeant l'équation (83), on peut écrire :

$$\boldsymbol{k}(n) = \lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{u}(n) - \lambda^{-1}\boldsymbol{k}(n)\boldsymbol{u}^T(n)\boldsymbol{P}(n-1)\boldsymbol{u}(n)$$

$$= \left[\lambda^{-1}P(n-1) - \lambda^{-1}\boldsymbol{k}(n)\boldsymbol{u}^H(n)\boldsymbol{P}(n-1)\right]\boldsymbol{u}(n) \qquad (85)$$

A partir de l'équation (84), on voit que l'expression entre crochets dans l'équation (85) est égale à $\boldsymbol{P}(n)$. Donc, on peut simplifier l'équation (85) comme suit :

$$\boldsymbol{k}(n) = \boldsymbol{P}(n)\boldsymbol{u}(n) \qquad (86)$$

Ce résultat, avec $P(n) = \Phi^{-1}(n)$ peut être utilisé comme définition pour le gain $\boldsymbol{k}(n)$.

Mise à jour pour le vecteur des coefficients de pondération

En utilisant les équations (73), (78) et (82), on peut exprimer l'estimation du vecteur des coefficients de pondération au sens des moindres carrés à l'instant n comme suit :

$$\hat{\boldsymbol{w}}(n) = \boldsymbol{\Phi}^{-1}(n)\boldsymbol{\Theta}(n) = \boldsymbol{P}(n)\boldsymbol{\Theta}(n) = \lambda\boldsymbol{P}(n)\boldsymbol{\Theta}(n-1) + \boldsymbol{P}(n)\boldsymbol{u}(n)\boldsymbol{d}(n) \qquad (87)$$

En substituant l'équation (84) pour P(n) dans le membre de droite du premier terme de l'équation (87), on obtient :

$$\hat{\boldsymbol{w}}(n) = \boldsymbol{P}(n-1)\boldsymbol{\Theta}(n-1) - \boldsymbol{k}(n)\boldsymbol{u}^T(n)\boldsymbol{P}(n-1)\boldsymbol{\Theta}(n-1) + \boldsymbol{P}(n)\boldsymbol{u}(n)\boldsymbol{d}(n)$$

$$= \boldsymbol{\Phi}^{-1}(n-1)\boldsymbol{\Theta}(n-1) - \boldsymbol{k}(n)u^T(n)\boldsymbol{\Phi}^{-1}(n-1)\boldsymbol{\Theta}(n-1) + \boldsymbol{P}(n)\boldsymbol{u}(n)\boldsymbol{d}(n)$$

$$= \hat{\boldsymbol{w}}(n-1) - \boldsymbol{k}(n)\boldsymbol{u}^T(n)\hat{\boldsymbol{w}}(n-1) + \boldsymbol{P}(n)\boldsymbol{u}(n)\boldsymbol{d}(n) \qquad (88)$$

En utilisant l'égalité $\boldsymbol{k}(n)=\boldsymbol{P}(n)\boldsymbol{u}(n)$ de l'équation (86), on obtient :

$$\hat{\boldsymbol{w}}(n) = \hat{\boldsymbol{w}}(n-1) + \boldsymbol{k}(n)\left[d(n) - \boldsymbol{u}^T(n)\hat{\boldsymbol{w}}(n-1)\right] = \hat{\boldsymbol{w}}(n\text{-}1) + \boldsymbol{k}(n)\alpha(n) \qquad (89)$$

où $\alpha(n)$ est l'innovation définie par

$$\alpha(n) = d(n) - \boldsymbol{u}^T(n)\boldsymbol{w}(n-1) = d(n) - \hat{\boldsymbol{w}}^{\mathrm{T}}(n\text{-}1)\,\boldsymbol{u}(n) \qquad (90)$$

Le produit $\hat{\boldsymbol{w}}^T(n-1)\boldsymbol{u}(n)$ représente une estimation de la réponse désirée $d(n)$ basée sur l'ancienne estimation du vecteur des coefficients de pondération à l'instant n-1. $\alpha(n)$ est référé aussi comme l'erreur d'estimation a priori.

L'équation (89) pour la mise à jour du vecteur des coefficients de pondération et l'équation (90) pour l'innovation suggèrent que le schéma bloc de la figure 6 pour l'algorithme récursif RLS.

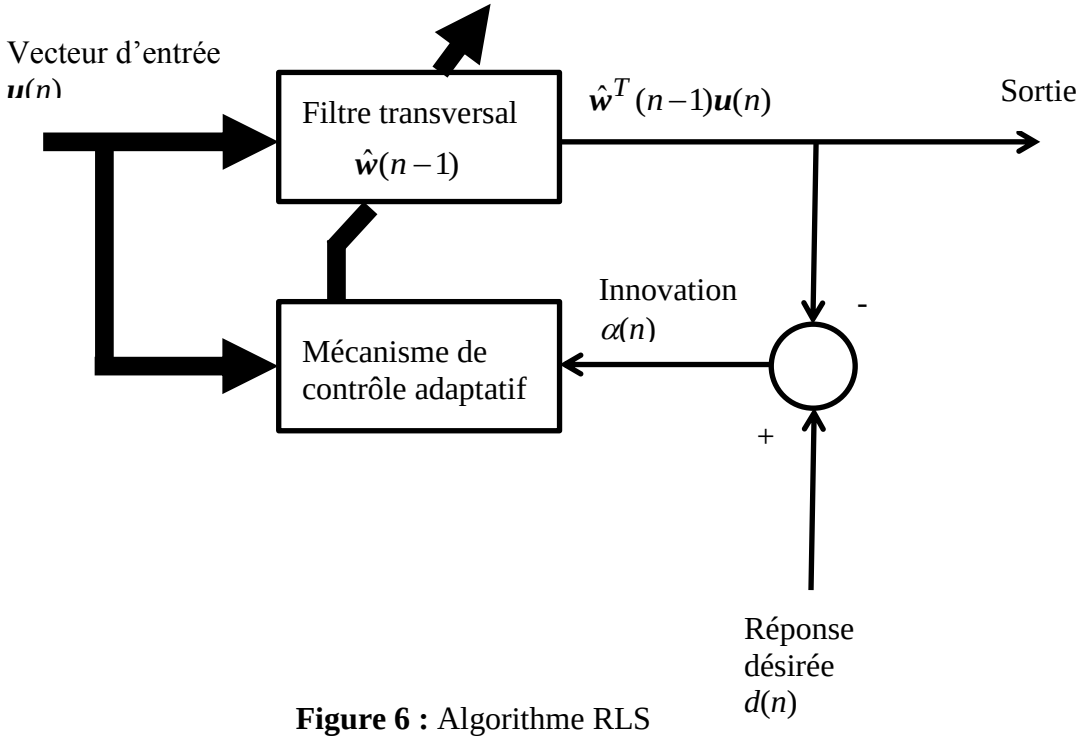

Figure 6 : Algorithme RLS

L'innovation $\alpha(n)$ est en général différente de l'erreur d'estimation a posteriori

$$e(n) = d(n) - \hat{\boldsymbol{w}}^T(n)\boldsymbol{u}(n) \qquad (91)$$

dont le calcul fait intervenir l'estimation courante des moindres carrés du vecteur des coefficients de pondération disponible à l'instant n. L'innovation $\alpha(n)$ peut être vue comme une valeur tentative de $e(n)$ avant la mise à jour du vecteur des coefficients de pondération.

$\alpha(n)$ sera référé comme étant l'erreur d'estimation a priori et $e(n)$ l'erreur d'estimation a posteriori.

Les équations (83), (90), (89) et (84) collectivement dans cet ordre constituent l'algorithme RLS résumé comme suit :

$$\boldsymbol{k}(n) = \frac{\lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{u}(n)}{1+\lambda^{-1}\boldsymbol{u}^T(n)\boldsymbol{P}(n-1)\boldsymbol{u}(n)}$$

$$\alpha(n) = d(n) - \hat{\boldsymbol{w}}^{\mathrm{T}}(n-1)\,\boldsymbol{u}(n)$$

$$\hat{\boldsymbol{w}}(n) = \hat{\boldsymbol{w}}(n-1) + \boldsymbol{k}(n)\alpha(n)$$

$$\boldsymbol{P}(n) = \lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}\boldsymbol{k}(n)\boldsymbol{u}^T(n)\boldsymbol{P}(n-1)$$

Le schéma bloc de la figure montre 6 une représentation de l'algorithme RLS qui complète le schéma bloc de la figure précédente.

Initialisation de l'algorithme RLS

L'application de l'algorithme RLS nécessite l'initialisation de la relation récursive (84) en choisissant une valeur de départ $\boldsymbol{P}(0)$ qui assure la non singularité de la matrice de corrélation $\boldsymbol{\Phi}(n)$. Ceci peut être effectué en évaluant l'inverse de

$$\left[\sum_{i=-n_0}^{0}\lambda^{-i}u(i)u^T(i)\right]^{-1}$$

Où le vecteur des échantillons de pondération d'entrée est obtenu à partir du bloc de données pour $-n_0 \le i \le 0$

Une procédure plus simple est de modifier légèrement l'expression pour la fonction de corrélation $\boldsymbol{\Phi}(n)$ en écrivant

$$\boldsymbol{\Phi}(n)=\sum_{i=1}^{n}\lambda^{n-i}\boldsymbol{u}(i)\boldsymbol{u}^T(i)+\delta\lambda^n\boldsymbol{I} \qquad (92)$$

où $\boldsymbol{I}$ est la matrice identité MxM et δ une petite constante positive. En posant n=0 dans l'équation (71), on obtient

$$\boldsymbol{\Phi}(0)=\delta\,\boldsymbol{I}$$

Soit donc

$$\boldsymbol{P}(0)=\delta^{-1}\boldsymbol{I} \qquad (93)$$

Le vecteur des coefficients de pondération peut être initialisé au vecteur nul

$$\hat{\boldsymbol{w}}(0)=\boldsymbol{0} \qquad (94)$$

Résumé de l'algorithme RLS

Initialisation de l'algorithme en posant

$$\boldsymbol{P}(0)=\delta^{-1}\boldsymbol{I},\ \delta \text{ une petite constante positive}$$

$$\hat{\boldsymbol{w}}(0)=\boldsymbol{0}$$

Pour chaque instant n=1, 2, … calculer

$$\boldsymbol{\pi}(n)=\boldsymbol{u}^H(n)\boldsymbol{P}(n-1)$$

$$\kappa(n)=\lambda+\boldsymbol{\pi}(n)\boldsymbol{u}(n)$$

$$\boldsymbol{k}(n)=\frac{\boldsymbol{P}(n-1)\boldsymbol{u}(n)}{\kappa(n)}$$

$$\alpha(n) = d(n) - \hat{\boldsymbol{w}}^{\mathrm{H}}(n-1)\,\boldsymbol{u}(n)$$

$$\boldsymbol{P}'(n-1) = \boldsymbol{k}(n)\boldsymbol{\pi}(n)$$

$$\boldsymbol{P}(n) = \frac{1}{\lambda}\left(\boldsymbol{P}(n-1) - \boldsymbol{P}'(n-1)\right)$$