

Filtrage adaptatif : théorie et applications
Volume 1
théorie et algorithmes

sous la direction de
François MICHAUT
Maurice BELLANGER

version 6
26 Juin 2005

LISTE DES AUTEURS

BELLANGER Maurice
CNAM PARIS
adresse : 292 rue Saint Martin 75141 Paris Cedex 03
mel : <bellang@cnam.fr>

BROSSIER Jean-Marc
LIS/ENSIEG Grenoble
BP46. 38402 Saint Martin d'Hères cedex.
mel : <jean-marc.brossier@lis.inpg.fr>

DELMAS Jean Pierre CITI Department
Institut National des Télécommunications Evry
9 rue Charles Fourier 91011 EVRY
email jean-pierre.delmas@int-evry.fr

GILLOIRE Andre
France Telecom Lannion
adresse : , Technopôle Anticipa, 2 Av Pierre Marzin 22307 Lannion Cedex
mel : <andre.gilloire@rd.francetelecom.com>

MICHAUT François
ENSEA Cergy (Département Signal et Télécoms)
6 avenue du Ponceau 95000 Cergy
mel : michaut@ensea.fr

REGALIA Phillip A.
Institut National des Telecommunications Evry
9, rue Charles Fourier F-91011 Evry cedex France
Phillip.Regalia@int-evry.fr or p.regalia@ieee.org

SCALART Pascal
ENSSAT, Lannion
6 rue de Kerampont BP 80518, 22305 Lannion Cedex
Mel : pascal.scalart@enssat.fr

Notations et abréviations

$\mathbb{R} \mathbb{C} \mathbb{Z} \mathbb{N}$	Ensembles numériques
$H = L^2(\Omega)$	Espace de Hilbert des v.a. de carré intégrable
$H = \mathbb{R}^d$	Espace euclidien standard
A , x	Matrices et vecteurs fixes en gras droit (possibilité)
$\langle \mathbf{x}, \mathbf{y} \rangle$	Produit scalaire sur H espace de Hilbert
ω	Epreuve aléatoire ou réalisation
p.a.	Processus aléatoire
p.s.	Propriété "presque sûre" (avec proba 1)
m.q.	En moyenne quadratique (convergence)
$E(X) = m_X$	Espérance du vecteur aléatoire X
$X_c = X - m_X$	Variable ou vecteur aléatoire centré
$E(X Y)$	Espérance conditionnelle à la tribu $\sigma(Y)$
$EL(X Y)$	Estimation linéaire en m.q. sur $Vect(Y)$
$cov(X, Y)$	Covariance de 2 variables aléatoires
$\Gamma(X, Y) = E(X_c Y_c^T) = cov(X, Y)$	Matrice intercovariance de vect. aléatoires
$var(X) = \sigma_X^2$	Variance de X
i.i.d., bruit blanc fort	Indépendant identiquement distribué (p.a.)
x_n ou $x(n)$ ou $x(n, \omega)$ $n \in \mathbb{Z}$	Signal numérique aléatoire
$\Gamma_x(n, m) = cov(x_n, x_m)$	Fonction de covariance d'un p.a.
$\gamma_x(p) = cov(x_n, x_{n-p})$	Fonction de corrélation de x_n
$\gamma_{xy}(p) = cov(x_n, y_{n-p})$	Fonction d'intercorrélation de x et y
$\overset{\circ}{\gamma}_x(\nu) = \sum_{-\infty}^{+\infty} e^{-2i\pi\nu p} \gamma_X(p)$	Densité spectrale (DSP)
$\bar{\gamma}_x(z) = \sum_{p=-\infty}^{+\infty} \gamma_X(p) z^{-p}$	Densité en z de x
$\nu \in [1]$	Intervalle des fréquences de longueur 1
$\overset{\circ}{x}(\nu)$	TF de la suite numérique x
$\bar{x}(z)$	TZ de la suite x
$Df(a)$	Différentielle de $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$
$J(\theta)$	Fonction de coût de l'algorithme
EQM	Erreur quadratique moyenne
$J_{\min} = J(\theta^*)$	EQM minimale du θ^* optimal
θ ou $w \in \mathbb{R}^d$	Vecteur des paramètres à estimer
$\mu_n \mu \lambda_n \lambda = 1 - \mu$	Pas ou facteur d'oubli de l'algorithme
$\nabla J(w) \in \mathbb{R}^d$	Gradient en w de la fonction $J : \mathbb{R}^d \rightarrow \mathbb{R}$
$\nabla^2 J(w)$	Hessien de $J(w)$ (matrice $n \times n$ symétrique)
$A^T A^*$	Transposée ou transconjuguée d'une matrice
R = $E(U.U^T)$	Matrice d'autocorrélation de (u_n) , > 0
$cond(R) = \lambda_{\max}/\lambda_1$	Conditionnement de la matrice $R > 0$
$\rho(A) = \max(\lambda_i(A))$	Rayon spectral de la matrice carrée A
$Tr(A) = \sum \lambda_i$	Trace de la matrice carrée A
$Sp(A)$	Spectre (ensemble des v.p.) de A
$Re \lambda(A) > 0$	v.p. de A à partie réelle > 0
$U(n), \varphi_n, X(n)$	vecteur d'état de l'algorithme

Avant-Propos

Le Traité IC2 relève le double défi de l'accessibilité à tous des sciences et techniques, et du transfert de connaissances, dans le domaine Information, Traitement du Signal et Communications. Il était donc nécessaire qu'un de ses volumes fasse le point sur le Filtrage Adaptatif, domaine qui a connu une grande activité dans la communauté du Traitement du Signal et des images depuis les années 1970. C'est l'objectif de cet ouvrage, qui sera publié en deux volumes :

Volume 1 : Théorie et algorithmes

Volume 2 : Applications du Filtrage adaptatif en Signal et Télécommunications

Le Volume 1 présenté ici se situe au niveau des algorithmes de réalisation adaptative (en ligne, dans des contextes éventuellement non stationnaires) des tâches d'optimisation du TS : les domaines d'applications sont nombreux (filtrage optimal de Wiener, annulation d'écho et suppression de bruit, égalisation ou filtrage inverse, analyse spectrale, compression de l'information, modélisation), et plusieurs d'entre eux seront approfondis dans le Volume 2.

Pour des raisons de simplicité d'écriture et d'énoncé des résultats, nous traiterons exclusivement du Traitement de Signaux réels monodimensionnels, même si la plupart des méthodes exposées s'étendent à des cas complexes ou multidimensionnels. Nous nous proposons donc :

1) d'exposer les principes de base et les méthodes générales qui sous-tendent la multiplicité des approches que l'on trouve dans la littérature spécialisée : quelques idées simples doivent servir de guide pour se retrouver dans ce maquis, pour classifier les algorithmes particuliers, et être capable de les adapter à un besoin spécifique,

2) de donner un accès rapide et direct aux algorithmes, leur compréhension, leur utilisation, et les résultats que l'on peut en attendre : quelques exemples de base seront développés et permettront d'illustrer (y compris par des simulations) et de comparer les différents algorithmes,

3) de donner les résultats mathématiques plus sophistiqués nécessaires à l'étude des problèmes de convergence des algorithmes : ces compléments pourront alors servir d'introduction et de guide général pour aborder des ouvrages plus spécialisés ou des articles donnés en références.

Cet ouvrage est une synthèse de nombreux travaux faits dans le domaine. Il pourra intéresser aussi bien des étudiants de 2^o et 3^o cycle universitaires ou des écoles d'ingénieurs utilisant le Traitement du Signal, que les ingénieurs ou scientifiques travaillant dans le domaine du Traitement du Signal et des Télécommunications. Le langage mathématique utilisé est accessible au scientifique-ingénieur muni d'une culture standard de niveau second cycle universitaire en algèbre linéaire et probabilités-statistique. Des connaissances à ce niveau sont également requises concernant les concepts de base et problèmes du Traitement du Signal .

Le contenu du livre

Le Chapitre 1 présente le problème de base du Filtrage de Wiener, dans sa version de filtrage RIF. Il est remis dans le contexte général des problèmes d'optimisation en TS, et on donne ensuite les algorithmes standards d'Analyse numérique permettant la résolution de l'optimisation déterministe de la fonction de coût de Wiener (Gradients, Moindres carrés).

Le Chapitre 2 étudie les deux grandes classes d'algorithmes adaptatifs utilisés pour le Filtrage transverse (RIF) : gradients stochastiques (LMS) et Moindres carrés récursifs (RLS). Ceci constitue l'essentiel des utilisations courantes des traitements adaptatifs en TS, et les propriétés de convergence y sont étudiées par des méthodes spécifiques à ces deux algorithmes. Les variantes intermédiaires (NLMS, Projection affine APA, algorithmes en treillis) sont aussi présentées. La convergence et les performances des algorithmes sont étudiés dans les contextes stationnaires.

Le Chapitre 3 étend ces algorithmes à des modèles où la fonction de coût $J(\theta)$ n'est plus quadratique en θ . Les méthodes générales d'étude de la convergence transitoire, du comportement asymptotique de fluctuation en situation stationnaire, puis de poursuite de modèle en contexte lentement variable, sont développées et illustrées sur plusieurs exemples simples. En particulier des méthodes d'approximation de sous-espaces sont développées en détail, en montrant à la fois l'intérêt et aussi les limitations des méthodes et résultats généraux de convergence.

Le Chapitre 4 présente plusieurs approches des Algorithmes de Moindres carrés rapides (MCR) indispensables pour la mise en oeuvre en temps réel de l'algorithme RLS : il s'agit de réduire la complexité de calcul pour la rendre proportionnelle à la longueur M du filtre adaptatif. Sont ainsi développés les algorithmes transversaux rapides, puis les versions en treillis ou à base de décomposition QR. Le contrôle de la stabilité numérique de ces algorithmes est aussi abordé.

Le Chapitre 5 approfondit le Filtrage RII adaptatif avec les différentes approches classiques des automaticiens (Equation d'erreur ou Erreur de sortie, Steiglitz-Mac Bride, Hyperstabilité). Les propriétés de convergence sont données. Ce chapitre complète donc naturellement les approches des Chapitres 2 et 3.

Contenu du Volume 2

Ch6 Adaptation sans référence : l'algorithme CMA
Inbar Fijalkow, Phillip Regalia

Ch7 Implémentations et problèmes associés
Olivier Sentieys

Ch8 Annulation d'écho
Jacob Benesty, André Gilloire, François Capman

Ch9 Egalisation et communications
Jacques Palicot, Maurice Bellanger, Inbar Fijalkow

Ch10 Antennes adaptatives, algorithmes avec contrainte
Pascal Chevalier, Jean-Marc Romano

Ch11 Apprentissage non linéaire
Jacques Palicot

Notations Une liste de notations de base est jointe à l'ouvrage. Cependant, la diversité des auteurs ayant travaillé de façon autonome sur les différents chapitres n'a pu permettre une normalisation systématique et uniforme des notations. Le lecteur devra donc en tenir compte et s'adapter à la spécificité de certaines notations dans chaque chapitre : nous avons essayé qu'il n'y ait pas d'ambiguïté dans l'interprétation de celles-ci dans leur contexte spécifique de chaque chapitre.

Références : Chaque chapitre a sa bibliographie intégrée. Les ouvrages et articles cités sont repérés par nom d'auteur et année, se référant à la bibliographie .

Nous remercions chaleureusement tous les auteurs qui ont bien voulu contribuer à la réalisation de cet ouvrage, et lui donner ainsi toute sa richesse issue de la diversité de leurs expériences d'enseignants, de chercheurs et d'ingénieurs de la communauté du Traitement du Signal.

François Michaut, Maurice Bellanger

le 29 Mai 2005

Table des matières

Avant-Propos	5
PREMIÈRE PARTIE. VOLUME 1 : THÉORIE ET ALGORITHMES	15
Chapitre 1. Filtrage adaptatif : le problème de base	17
Jean-Marc BROSSIER, François MICHAUT	
1.1. Traitement du signal et optimisation	17
1.1.1. Pourquoi du filtrage adaptatif ?	17
1.1.2. Soustraction de bruit, égalisation et identification	18
1.2. Estimation optimale	20
1.2.1. Estimation du maximum de vraisemblance et bayésienne	20
1.2.2. Estimation en moyenne quadratique (m.q.)	22
1.2.3. Deux approches de l'ELMQ : Wiener et Kalman	23
1.3. Le filtrage de Wiener RIF	26
1.3.1. La solution RIF : équations normales	26
1.3.2. Le point de vue déterministe : moindres carrés	27
1.4. Optimisation déterministe : les algorithmes	29
1.4.1. Gradient déterministe	29
1.4.2. Newton et quasi-Newton	31
1.4.3. Autres méthodes : gradient conjugué...	33
1.5. Passage aux algorithmes adaptatifs et rôle de l'ergodisme	33
1.5.1. Choix du critère à optimiser	34
1.5.2. Choix d'un algorithme déterministe d'optimisation	35
1.5.3. Algorithme stochastique et estimation statistique	36
1.5.4. Le lien entre point de vue déterministe et probabiliste : l'ergodisme.	38
1.6. Bibliographie	41
Chapitre 2. Filtrage RIF adaptatif	43
François MICHAUT, André GILLOIRE, Pascal SCALART	

2.1. Gradient stochastique	43
2.1.1. Le gradient stochastique : LMS	44
2.1.2. Convergence du LMS	45
2.1.3. Variantes du LMS : NLMS, formes par blocs	54
2.2. Moindres carrés récurrents : RLS	60
2.2.1. L'algorithme récursif : RLS	60
2.2.2. Algorithme adaptatif : facteur d'oubli	62
2.2.3. Convergence et propriétés statistiques	63
2.2.4. Simulations et comparaisons RLS-LMS	69
2.3. Algorithmes de projection affine	73
2.3.1. Introduction	73
2.3.2. Liens avec les autres familles d'algorithmes	76
2.3.3. Régularisation directe de la matrice de covariance des données	77
2.3.4. Convergence de l'algorithme APA	79
2.3.5. Complexité algorithmique et algorithmes rapides	80
2.4. Structures en treillis	81
2.4.1. Introduction	81
2.4.2. Le treillis gradient et variantes	83
2.4.3. Le treillis Moindres Carrés	85
2.5. Performances comparées des algorithmes adaptatifs	87
2.5.1. Performances de APA et NLMS	87
2.5.2. Performances des algorithmes en treillis	91
2.6. Bibliographie	95
Chapitre 3. Algorithmes adaptatifs : méthodes générales	99
Jean-Pierre DELMAS, Jean-Marc BROSSIER , François MICHAUT	
3.1. Nécessité de traitements adaptatifs plus généraux	100
3.1.1. Filtrage RII, modélisation ARMA	100
3.1.2. Egalisation linéaire aveugle	101
3.1.3. Estimation de retards	103
3.1.4. Approximation de sous-espaces, réseau de neurones	103
3.1.5. Conception des algorithmes	105
3.2. Forme générale d'algorithmes, méthodes pour la convergence (ODE)	107
3.2.1. Forme générale, vecteur d'état	107
3.2.2. Algorithme déterministe et méthode de l'ODE	109
3.2.3. Stabilité de l'ODE et fonction de Liapounoff	113
3.2.4. Exemples : ODE, convergence, condition SPR	115
3.3. Algorithmes à pas constant : fluctuation stationnaire	118
3.3.1. Fluctuation : Approximation-Diffusion gaussienne et EDS	120
3.3.2. Exemples	122
3.4. Poursuite de non-stationnarités	124
3.4.1. Problématique de la poursuite	124
3.4.2. Exemple de l'estimateur de moyenne	126

3.4.3. Méthode générale : diffusions gaussiennes et ODE	128
3.4.4. Le cas standard : LMS et RLS, exemples	131
3.5. Application : Méthodes d'approximation de sous-espaces	134
3.5.1. Construction des algorithmes	134
3.5.2. Etude de convergence	140
3.6. Conclusion	143
3.7. Bibliographie	143
Chapitre 4. Algorithmes rapides, vitesse et stabilité numérique	147
Maurice BELLANGER	
4.1. Introduction et notations	147
4.2. Filtres transverses rapides	148
4.2.1. Equations de récurrence du filtre d'ordre M	148
4.2.2. Relations entre les variables des Moindres carrés	150
4.2.3. Algorithme MCR basé sur les erreurs <i>a priori</i>	155
4.2.4. Algorithme MCR basé sur l'ensemble des erreurs de prédiction	159
4.2.5. Conditions de stabilité et initialisation	161
4.2.6. Contrôle de l'accumulation des erreurs d'arrondi	166
4.2.7. Moindres carrés avec décimation	168
4.3. Algorithmes en treillis	169
4.3.1. Récurrences sur l'ordre pour les coefficients de prédiction	169
4.3.2. Relations de récurrence sur l'ordre pour les coefficients du filtre	173
4.3.3. Relations de récurrence temporelle	175
4.3.4. Algorithmes MCR pour les structures en treillis	176
4.3.5. Algorithmes pour treillis avec normalisation	178
4.3.6. Calcul des coefficients des filtres transversaux	184
4.4. Algorithmes de décomposition QR	184
4.4.1. L'opération de rotation	185
4.4.2. La décomposition QR	186
4.4.3. Rotations en prédiction linéaire arrière	187
4.4.4. Rotation en prédiction linéaire avant	191
4.4.5. L'algorithme QR rapide	193
4.4.6. Equivalence avec le treillis	195
4.5. Conclusion	197
4.6. Bibliographie	197
Chapitre 5. Filtrage RII Adaptatif	199
Phillip REGALIA	
5.1. Introduction	199
5.2. Principes d'Identification du Système	200
5.3. Méthode d'erreur d'équation	200
5.3.1. Développement Algorithmique	204
5.3.2. Propriétés de Modélisation	206

14 Filtrage adaptatif

5.4. Méthode de minimisation de l'erreur de sortie	209
5.5. Méthode de Steiglitz et McBride	213
5.5.1. Développement Algorithmique	214
5.6. Algorithmes à Base d'Hyperstabilité	217
5.7. Algorithme Hyperstable Simplifié	222
5.8. Convergence des Algorithmes en Erreur de Sortie	223
5.8.1. Méthode de Gradient	225
5.8.2. Algorithme de Steiglitz-McBride	226
5.8.3. SHARF	229
5.9. Approximation dans le Cas Sous-Modélisé	231
5.10. Conclusions	235
5.11. Bibliographie	235
Index	239
5.12.*	242
5.13.*	242
5.14.*	242
5.15.*	242
5.16.*	242
5.17.*	243

PREMIÈRE PARTIE

Volume 1 : théorie et algorithmes

Chapitre 1

Filtrage adaptatif : le problème de base

CONTENU.— Ce chapitre introductif présente la démarche adaptative dans le contexte des applications en traitement du signal. La formalisation en termes d'estimation optimale est abordée ainsi que les principales approches statistiques. On se concentre ensuite sur le problème de base, le plus simple et en même temps le plus répandu dans les applications réelles : le filtrage de Wiener RIF (Réponse Impulsionnelle Finie) en contexte stationnaire. Enfin sont abordées les méthodes d'optimisation standards de l'analyse numérique qui servent de modèle à la construction des algorithmes adaptatifs. On termine par les propriétés d'ergodisme des signaux qui permettent de relier le point de vue probabiliste théorique — à la Wiener — et le point de vue déterministe pratique des méthodes adaptatives de traitement des données en ligne.

1.1. Traitement du signal et optimisation

1.1.1. Pourquoi du filtrage adaptatif?

Les méthodes adaptatives en traitement du signal visent l'adaptation automatique des opérateurs de traitement aux propriétés statistiques des signaux et des systèmes, ainsi que l'adaptation à leurs variations dans le temps. Il s'agit donc d'un mélange bien pondéré entre la stationnarité, qui permet grâce à la permanence dans le temps de propriétés statistiques de se débarrasser, ou tout au moins de réduire, les fluctuations purement aléatoires, et la non-stationnarité, c'est-à-dire la variation «lente» au cours du temps de ces propriétés, sans laquelle il n'y aurait pas besoin de l'adaptatif : il suffirait de calculer une fois pour toute le «filtre optimal» puis de le mettre en ligne.

Chapitre rédigé par Jean-Marc BROSSIER et François MICHAUT.

Ces méthodes ont connu un essor considérable depuis les années 60, dû au développement du traitement numérique et à l'augmentation constante de la puissance des processeurs de traitement (DSP, *Digital Signal Processors*), permettant la mise en œuvre en temps réel d'algorithmes de plus en plus sophistiqués, à des cadences de plus en plus rapides. Elles sont arrivées à une certaine maturité aussi bien en ce qui concerne le développement et l'implémentation des algorithmes, que du point de vue des outils théoriques d'étude des performances. Ce chapitre se propose d'en donner une vue synthétique, non exhaustive mais suffisante, pour permettre au lecteur d'y trouver rapidement les outils et résultats qui l'intéressent, et éventuellement les références vers des ouvrages permettant d'approfondir des aspects spécifiques.

1.1.2. Quelques exemples : soustraction de bruit, égalisation et identification.

Ce paragraphe présente succinctement trois exemples classiques d'application du filtrage adaptatif : la soustraction de bruit, l'égalisation et l'identification directe. Ces exemples ne constituent qu'une infime fraction des applications classiques du filtrage adaptatif parmi lesquelles figurent notamment l'annulation d'écho, certains codages de compression, la formation de voies ainsi que de nombreuses techniques de traitement d'antenne. Le lecteur intéressé par des présentations détaillées d'applications du filtrage adaptatif peut se référer, entre autres, aux articles fondateurs [Widrow60, 67, 75] et [Bellanger 85].

Soustraction de bruit. Le schéma typique d'un dispositif de soustraction de bruit est celui de la figure (1.1). Un signal observé se compose d'un signal utile, non observé que l'on souhaite estimer, pollué de manière additive par un bruit supposé indépendant du signal utile. Lorsque ce bruit sur l'observation est obtenu par filtrage linéaire d'une source de bruit auprès de laquelle il est possible de placer un capteur, il devient envisageable d'estimer le bruit pour ensuite le soustraire du signal observé.

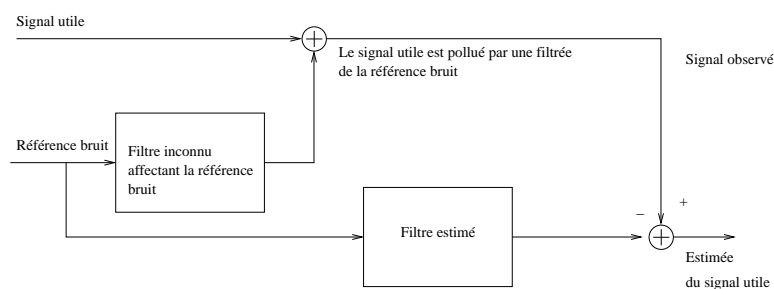


Figure 1.1. Soustraction de bruit.

Egalisation. Le problème est celui de la figure (1.2) : l'observation est une version perturbée par un bruit blanc additif b , de variance σ_b^2 , de la sortie d'un filtre $H(z)$ —

le canal de transmission — attaqué par un bruit blanc normé (les données transmises). Le but est d'estimer les données à partir des observations. Si l'on choisit de minimiser la puissance de l'erreur entre les données transmises et la sortie du filtre égaliseur, la meilleure solution linéaire est le filtre de Wiener dont la fonction de transfert en z s'écrit :

$$W(z) = \frac{H^*(1/z^*)}{H(z)H^*(1/z^*) + \sigma_b^2}.$$

En l'absence de bruit ($\sigma_b^2 = 0$), $W(z)$ se réduit au filtre inverse $1/H$. En présence de bruit, de fortes valeurs du gain $1/H$ peuvent conduire à trop amplifier le bruit, la solution de Wiener régularise le filtre inverse grâce à la prise en compte de la constante σ_b^2 : elle est semblable au filtre inverse aux fréquences pour lesquelles le rapport signal à bruit est fort mais amplifie moins les zones de faible rapport signal à bruit.

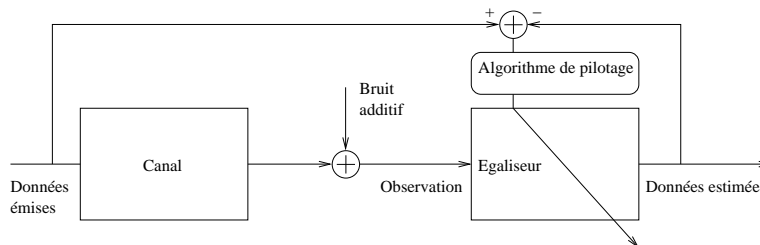


Figure 1.2. L'égalisation est une «identification inverse».

Identification. Ayant accès à l'entrée et à la sortie d'un filtre linéaire dont la sortie est bruitée, un problème d'identification directe consiste à estimer le filtre linéaire inconnu. Ce problème correspond au schéma de la figure (1.3). Lorsque le système inconnu est susceptible de varier au fil du temps, le processus d'identification peut être effectué à l'aide d'un traitement adaptatif.

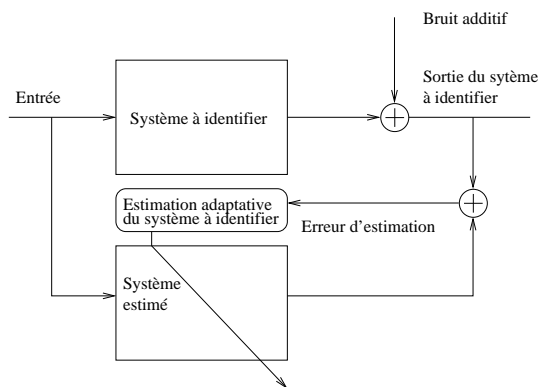


Figure 1.3. Identification directe.

Voyons sur cet exemple pourquoi la solution du problème se formalise en termes d'optimisation d'un critère de coût, ici l'*erreur quadratique moyenne* (EQM) minimale. Notons u_n le signal d'entrée, $v_n = [H_*(z)]u_n$ la sortie du système à identifier $H_*(z)$, b_n le bruit additif de sortie, $x_n = v_n + b_n$ la sortie bruitée (signal de référence), et enfin $y_n = [H(z)]u_n$ la sortie du système estimé $H(z)$. On compare les deux systèmes grâce à l'*erreur d'estimation* $e_n = x_n - y_n = b_n + (H_* - H)u_n$. La puissance moyenne, ou variance pour des signaux centrés, de cette erreur est nommée EQM de l'identification, et se calcule facilement ici compte-tenu de l'indépendance des signaux aléatoires u_n et b_n :

$$EQM(H) = E(e_n^2) = \sigma_b^2 + \text{var}\{(H_* - H)u_n\} \geq \sigma_b^2. \quad (1.1)$$

L'identification du système H_* coïncide ici avec la minimisation de l'EQM.

1.2. Estimation optimale

Avant d'envisager l'adaptativité des traitements, ce paragraphe éclaire les notions indispensables à la définition de l'objectif à atteindre.

Les exemples du paragraphe précédent, et plus particulièrement celui de l'identification, indiquent le rôle central que joue l'estimation optimale. Celle-ci se décline selon différents critères tels que la moyenne quadratique (espérance conditionnelle, projection orthogonale, ...), le maximum de vraisemblance ou bien encore l'estimation bayésienne avec ses notions de coût et de risque. Bien que l'estimation en moyenne quadratique soit de très loin la technique la plus employée, d'autres critères d'optimisation que l'EQM peuvent être indispensables dans certains cas ; dans les problèmes d'estimation sans référence, par exemple, des critères qui utilisent des moments d'ordre différent de deux pour différencier des signaux indiscernables à l'ordre deux sont nécessaires.

Ce paragraphe débute par une très brève présentation des critères généraux classiques tels que le maximum de vraisemblance ou l'estimation bayésienne. La suite est consacrée à l'estimation en moyenne quadratique, technique fondamentale pour laquelle deux exemples célèbres servent d'illustration : le filtre de Kalman et celui de Wiener.

1.2.1. Estimation non linéaire : maximum de vraisemblance, estimateurs bayésiens.

Nous donnons ici un bref rappel de quelques critères classiques d'estimation utilisés en statistique. En général, ces critères mènent à des méthodes non-linéaires.

Le point de vue de l'estimation paramétrique est le suivant : une observation aléatoire vectorielle Y possède une loi de probabilité qui dépend d'un vecteur de paramètres à estimer θ . Dans le contexte signal, θ représente les paramètres du modèle de filtrage qui relie l'observation Y au signal de référence X .

La vraisemblance de l'observation Y est alors $L(\theta) = p(y|\theta)$, avec $\theta \in \Theta \subset \mathbb{R}^d$.

Maximum de vraisemblance. On choisit comme estimateur de θ , la valeur qui maximise la vraisemblance des observations :

$$\hat{\theta}_{MV} = \text{Arg max}\{L(\theta), \theta \in \Theta\}. \quad (1.2)$$

Ce critère, qui ne privilégie *a priori* aucune des valeurs de θ , est naturellement adapté au cas d'un paramètre déterministe mais il peut également s'utiliser pour estimer un paramètre aléatoire lorsque sa distribution *a priori* est non informative (en quelque sorte «uniforme» pour un intervalle borné).

S'il existe, cet estimateur du maximum de vraisemblance est asymptotiquement efficace, c'est-à-dire qu'il atteint la borne de Cramer-Rao (cf paragraphe 1.5.3).

Estimation bayésienne. L'approche bayésienne consiste à introduire une information *a priori* sur les valeurs possibles du paramètre θ , en le considérant lui-même comme aléatoire de loi *a priori* connue $p(\theta)$.

On peut alors choisir l'estimateur $\hat{\theta}(Y)$ qui maximise la **probabilité a posteriori** de l'observation Y :

$$\hat{\theta}_{MAP} = \text{Arg max}\{p(\theta|y), \theta \in \Theta\} = \text{Arg max}\{p(\theta)p(y|\theta), \theta \in \Theta\}. \quad (1.3)$$

Une généralisation de cette approche consiste à associer à chaque estimation de θ un certain coût C évaluant la qualité de l'estimation. Ce coût peut être le carré de l'erreur (critère de l'EQM) ou autre chose. On note $\varepsilon(Y) = \hat{\theta}(Y) - \theta$ l'erreur d'estimation, différence entre la valeur estimée $\hat{\theta}(Y)$ et la vraie valeur θ . Un **coût** est une fonction positive C qui quantifie l'importance que l'on attache à l'erreur $\varepsilon(Y)$ commise lors de l'estimation de θ . Le **risque** est la valeur moyenne du coût $R = E\{C(\varepsilon)\} = \int \int C\{\hat{\theta}(y) - \theta\} p(y, \theta) d\theta dy$. L'estimation bayésienne optimale au sens du coût choisi est celle qui minimise le risque correspondant, solution de l'équation

$$\frac{\partial}{\partial \hat{\theta}(y)} \left\{ \int C\{\hat{\theta}(y) - \theta\} p(\theta | y) d\theta \right\} = 0.$$

Cette équation admet deux cas particuliers remarquables :

Coût uniforme. La solution obtenue à l'aide d'un coût uniforme est la solution au sens du maximum de probabilité *a posteriori* (MAP).

Coût quadratique. Pour le coût quadratique $C(\varepsilon) = \varepsilon^2$, la relation précédente devient $2 \int \{\hat{\theta}(y) - \theta\} p(\theta | y) d\theta = 0$. En remarquant que $\int p(\theta | y) d\theta = 1$, on trouve pour estimateur optimal l'espérance conditionnelle :

$$\hat{\theta}(Y) = \int \theta p(\theta | Y) d\theta = E(\theta|Y).$$

Lorsque la densité conditionnelle $p(\theta | Y)$ est symétrique autour de son maximum $\hat{\theta}_{\text{MAP}}$ l'estimation au sens du maximum *a posteriori* coïncide avec la solution du minimum d'erreur quadratique.

Le cas du coût quadratique, simple cas particulier des méthodes précédentes, est d'une telle importance pratique que nous lui consacrons le paragraphe suivant.

1.2.2. Estimation en moyenne quadratique (m.q.) et théorème de projection.

Soit H un espace de Hilbert¹. Le résultat fondamental d'approximation dans les espaces de Hilbert, est le théorème de la projection orthogonale :

THÉORÈME.— [Projection orthogonale]

Soient H un espace de Hilbert muni du produit scalaire $\langle \cdot, \cdot \rangle$, $x \in H$ et $U \subset H$ un sous-espace de Hilbert ; alors la meilleure approximation en norme quadratique de x par un élément de U existe et est unique, c'est la projection orthogonale de x sur U caractérisée par :

$$\hat{x} = p_U(x) \Leftrightarrow \forall y \in U, \|x - \hat{x}\| \leq \|x - y\| \Leftrightarrow \hat{x} \in U \text{ et } x - \hat{x} \perp U. \quad (1.4)$$

Différents espaces de Hilbert trouvent leur place en traitement du signal. Pour le traitement des signaux aléatoires, l'espace de Hilbert adapté est l'ensemble $H = L^2(\Omega)$ des variables aléatoires (v.a.) de variance finie, muni du produit scalaire $\langle X, Y \rangle = E(XY^*)$ et de la pseudo-norme associée. Le théorème de la projection orthogonale s'applique donc directement aux problèmes d'estimation au sens de l'EQM minimale.

1. On appelle espace pré-hilbertien un espace vectoriel muni d'un produit scalaire et de la norme associée. Un espace de Hilbert est un espace pré-hilbertien complet, c'est-à-dire dans lequel toute suite de Cauchy converge.

Dans la suite, les v.a. sont supposées centrées. Le problème générique d'estimation en m.q. est alors le suivant :

Etant donné un vecteur aléatoire X non observé et un vecteur aléatoire Y , observation partielle et bruitée de X , statistiquement dépendante de X ; calculer une fonction des observations $\hat{X}(Y)$, qui estime au mieux X à partir des observations Y .

L'espérance conditionnelle : globalement optimale. L'espérance conditionnelle de X par rapport à l'ensemble des observations Y constitue la meilleure approximation de X en moyenne quadratique, dans la classe $L^2(Y)$ de toutes les fonctions non-linéaires de Y , de carré intégrable. Pour des v.a. centrées, l'espérance conditionnelle $\hat{X} = E(X | Y)$ est l'estimateur du minimum de variance, c'est-à-dire que pour toute fonction F (de \mathbb{R}^m dans \mathbb{R}^n) et $\forall \lambda \in \mathbb{R}^n$:

$$E \left\{ \lambda^T (X - \hat{X}) \right\}^2 \leq E \left\{ \lambda^T (X - F(Y)) \right\}^2. \quad (1.5)$$

Cependant, le calcul de cette espérance conditionnelle est généralement difficile, et nécessite la connaissance de la loi de probabilité conjointe du couple (X, Y) , c'est pourquoi on se limite souvent à l'estimation linéaire optimale.

Estimation linéaire en m.q. (ELMQ). Le meilleur estimateur linéaire $EL(\bullet | Y)$ de X à partir de Y possède des propriétés semblables à celles de la moyenne conditionnelle. En particulier, $EL(X | Y)$ est linéaire en X . De plus, si Y_1 est un ensemble d'observations qui contient l'ensemble d'observations Y_2 , on a : $EL(EL(X | Y_1) | Y_2) = EL(X | Y_2)$. Dans le cas gaussien, la décorrélation équivaut à l'indépendance d'où $E(\bullet | Y) = EL(\bullet | Y)$, l'estimateur linéaire est globalement optimal. Ceci signifie en particulier que si un algorithme calcule l'espérance conditionnelle à partir des deux premiers moments (moyenne et covariance) dans le cas gaussien, le même algorithme utilisé en contexte non gaussien réalise une projection linéaire. $EL(\bullet | Y)$ est l'estimateur linéaire de variance minimale, mais il est en général sous-optimal dans la classe $L^2(Y)$.

1.2.3. Deux approches de l'ELMQ : Wiener et Kalman

Il s'agit de deux approches complémentaires de l'ELMQ, obtenues en fixant *a priori* la structure de l'estimateur pour ensuite identifier les paramètres de cette structure. Elles recherchent l'optimum dans la classe des filtres linéaires et diffèrent par le mode de représentation des signaux. Elles reposent sur le même critère d'optimisation et conduisent donc essentiellement aux mêmes résultats².

2. Dans le cas de signaux stationnaires

Représentation interne des signaux et filtre de Kalman

Kalman a proposé (1961) pour résoudre le problème de l'ELMQ une structure de filtre basée sur une représentation interne des signaux. La représentation interne adoptée par Kalman caractérise un signal aléatoire par un modèle d'état, elle se prête particulièrement bien au maniement des systèmes non stationnaires. Elle a été construite au départ par les automaticiens, dans des contextes de poursuite optimale de trajectoire, et on la retrouve donc naturellement dans l'étude des algorithmes de poursuite des non-stationnarités (chapitre 3).

De la représentation interne des signaux découlent :

- une formulation récursive de la solution optimale, avec un horizon $D(n) = [0, \dots, n]$ croissant avec le temps,
- la prise en compte naturelle des non-stationnarités,
- un modèle d'état du système (le vecteur X_k) et des observations Y_k qui se compose d'un couple {équation d'état, équation d'observation} :

$$X_k = A_k X_{k-1} + W_{k-1} \quad (1.6)$$

$$Y_k = C_k X_k + V_k \quad (1.7)$$

Notations :

- W_k est le bruit de modèle, bruit blanc centré, de covariance Q_k .
- V_k bruit de «mesure» blanc de covariance R_k , indépendant de W_k .
- Y_k est une observation partielle de l'état, bruitée par V_k .
- Ψ_k est l'espace engendré par les observations $\{Y_1, \dots, Y_k\}$.
- Les matrices A_k et C_k reflètent la dynamique de l'état et la manière dont celui-ci est observé.

L'estimation linéaire optimale en moyenne quadratique de X_k connaissant toutes les observations jusqu'à l'instant m est notée $\tilde{X}_{k|m} = \text{EL}(X_k | \Psi_m)$. L'erreur d'estimation est notée $\tilde{X}_{k|m} = X_k - \hat{X}_{k|m}$ et la matrice de covariance de l'estimateur $P_{k|m} = \text{E}(\tilde{X}_{k|m} \tilde{X}_{k|m}^T)$. Le filtre de Kalman calcule l'estimation optimale $\hat{X}_{k|k}$ au sens de l'ELMQ de l'état X_k à partir de l'ensemble des observations effectuées jusqu'à l'instant k , sous une forme récursive [Chui 87] :

Algorithme de Kalman

$$\left\{ \begin{array}{ll} P_{n|n-1} = A_n P_{n-1|n-1} A_n^T + Q_{n-1} & \text{Equation variance 1} \\ K_n = P_{n|n-1} C_n^T (R_n + C_n P_{n|n-1} C_n^T)^{-1} & \text{Gain de Kalman} \\ P_{n|n} = (I - K_n C_n) P_{n|n-1} & \text{Equation variance 2} \\ \hat{X}_{n|n-1} = A_n \hat{X}_{n-1|n-1} & \text{Prédiction de l'état} \\ \hat{X}_{n|n} = \hat{X}_{n|n-1} + K_n (Y_n - C_n \hat{X}_{n|n-1}) & \text{Estimation de l'état} \end{array} \right.$$

Filtre de Kalman adaptatif. La mise en œuvre d'un filtre de Kalman suppose une connaissance *a priori* parfaite des paramètres du modèle d'état. En pratique, il arrive fréquemment que le modèle d'état ne soit connu qu'imparfaitement. Dans ce cas, il doit être partiellement estimé à l'aide des observations elles-mêmes. La structure du filtre de Kalman devient alors dépendante des données : le filtrage est dit adaptatif. La manière la plus courante d'estimer une partie du modèle en même temps que les paramètres utiles consiste à étendre le vecteur d'état du système pour y inclure les paramètres inconnus. De cette façon, l'estimation du vecteur d'état conduit à chaque itération à une mise à jour des paramètres à estimer et des paramètres inconnus du vecteur d'état. Nous renvoyons le lecteur intéressé à [Ljung 83].

Représentation externe des signaux et filtre de Wiener

L'approche de Norbert Wiener repose sur une représentation externe des signaux, c'est-à-dire une représentation qui décrit leurs propriétés statistiques : densités de probabilité et leurs sous-produits (moments, fonctions de corrélation, ...). En limitant l'étude aux filtres linéaires, seules les propriétés statistiques d'ordre un et deux interviennent. Ainsi, l'approche de Wiener repose sur la connaissance des propriétés statistiques d'ordre deux (fonctions de corrélation dans le domaine temporel ou interspectres et spectres dans le domaine des fréquences). Son importance tient à la fois à la simplicité du calcul théorique de la solution optimale et à la généralité de cette démarche.

Le contexte du filtrage de Wiener est résumé dans le schéma de la figure 1.4 : le signal observé est u_n , il est filtré par le filtre de Wiener $W(z)$, afin d'approcher au mieux en m.q. le *signal de référence* x_n . L'erreur d'estimation, à minimiser en m.q., est e_n . Tous ces signaux sont supposés aléatoires, centrés et stationnaires à l'ordre deux. Notons $D \subset \mathbb{Z}$ l'horizon d'observation — ou mémoire du filtre — ; le théorème de projection permet de caractériser le filtre optimal d'horizon D par l'orthogonalité entre l'espace d'observation $U_n = \{u_{n-k}, k \in D\}$ et l'erreur d'estimation :

$$y_n = \sum_{k \in D} w_k u_{n-k} \quad E(e_n U_n) = 0 \text{ avec } e_n = x_n - y_n. \quad (1.8)$$

Autrement dit, y_n estime la partie de x_n corrélée linéairement à u_n , et l'erreur d'estimation e_n est *décorrélée de l'espace d'observation*.

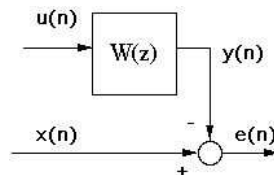


Figure 1.4. Contexte du filtrage de Wiener.

Solution générale non causale. La meilleure estimation est obtenue pour l'horizon maximal soit $D = \mathbb{Z}$, et dans ce cas la condition de décorrélation (1.8) donne une équation de convolution

$$w * \gamma_u(p) = \gamma_{xu}(p) \quad \forall p$$

dont la solution se calcule par Transformée en Z (TZ bilatérale) :

$$W(z) = \frac{\overline{\gamma}_{xu}(z)}{\overline{\gamma}_u(z)}. \quad (1.9)$$

Le filtre obtenu est généralement non causal et de durée infinie. Lorsque l'on recherche un filtre vérifiant certaines contraintes, il est nécessaire de reprendre le calcul précédent en tenant compte de ces contraintes. En pratique, il est clair que la contrainte de causalité est nécessaire à une réalisation temps réel. En général, la causalité ne peut pas être strictement respectée et l'on se contente d'une estimation retardée. Autrement dit, il faut tronquer une partie de la réponse anticausale du filtre. Pour la partie causale du filtre, deux solutions peuvent être envisagées : soit la tronquer pour obtenir une structure à réponse impulsionnelle finie (RIF) ; soit la modéliser par un modèle avec pôles et zéros. Dans ce premier chapitre, nous ne considérons que la première possibilité (paragraphe 1.3.1), la modélisation à Réponse Impulsionnelle Infinie (RII) étant abordée aux chapitres 3 et 5.

1.3. Le filtrage de Wiener RIF

Nous abordons maintenant plus en détails le problème de base, le plus simple de l'ELMQ, et le plus courant en filtrage adaptatif : le filtrage de Wiener à horizon d'observation (mémoire du filtre) fini $D_M = \{0, \dots, M-1\}$. Le filtre est alors RIF d'ordre M et le vecteur des paramètres à estimer $w = [w_0, \dots, w_{M-1}]^T$ correspond à la réponse impulsionnelle finie du filtre. L'espace d'observation est alors de dimension finie $U_n = [u_n, \dots, u_{n-M+1}]^T$.

Nous utilisons toujours le point de vue probabiliste : on raisonne sur la totalité des trajectoires de processus aléatoires, le traitement utilise les statistiques d'ordre deux des signaux et cherche à réaliser l'EQM minimale, ce qui revient à la décorrélérer l'erreur e_n et l'observation U_n .

1.3.1. La solution RIF : équations normales

L'estimation s'écrit $\widehat{x}_n = \sum_{k=0}^{M-1} w_k u_{n-k} = U_n^T w$. La matrice d'autocorrélation de U_n (ou matrice de covariance de ce vecteur aléatoire) et le vecteur d'intercorrélacion entre la référence x_n et l'observation U_n seront notés ainsi :

$$\Gamma_U = E(U_n U_n^T) \equiv \mathbf{R} \quad \Gamma_{xU} = E(x_n U_n) \equiv \mathbf{p} \quad (1.10)$$

et l'EQM s'exprime alors comme une fonction quadratique de w :

$$J(w) = \sigma_x^2 - 2\mathbf{p}^T w + w^T \mathbf{R} w. \quad (1.11)$$

Les équations de décorrélation caractérisant le filtre optimal (1.8) deviennent ici un système linéaire $M \times M$ dit des *équations normales*

$$\mathbf{R} w = \mathbf{p}, \quad (1.12)$$

équivalent à l'annulation du gradient $\nabla J(w) = 2(\mathbf{R} w - \mathbf{p})$. Si l'espace d'observation est de rang plein M , la matrice \mathbf{R} est inversible et il existe une solution unique de Wiener : $w_* = \mathbf{R}^{-1} \mathbf{p}$. Le calcul de l'EQM minimale découle de l'orthogonalité entre $X_k - \widehat{X}_k$ et \widehat{X}_k :

$$\varepsilon_{\min}^2 = E(\widehat{x}_k - x_k)^2 = E[x_k(x_k - \widehat{x}_k)] = \gamma_x(0) - \mathbf{p}^T w_*. \quad (1.13)$$

L'autocorrélation Γ_U et l'intercorrélation Γ_{xU} doivent être connues *a priori*. Lorsque tel n'est pas le cas, ces quantités doivent être préalablement estimées. Une alternative est le point de vue déterministe.

1.3.2. Le point de vue déterministe : moindres carrés

Dans les approches stochastiques de Wiener et de Kalman, les données sont considérées comme une réalisation particulière d'un processus aléatoire. Les propriétés statistiques de ce processus sont supposées connues au second ordre. De ce fait, la procédure de traitement obtenue ne dépend pas d'une réalisation particulière mais seulement des propriétés statistiques connues *a priori*. On privilégie ainsi les réalisations les plus probables au détriment de celles qui le sont moins.

Dans une approche déterministe au contraire, le traitement est adapté à la réalisation particulière que l'on observe. On obtient ainsi un algorithme dépendant des données : le meilleur estimateur pour la séquence particulière observée.

Bien sûr, les deux approches se rejoignent en pratique dès lors que les propriétés statistiques des signaux doivent être estimées à partir des données observées. La passerelle entre les deux approches repose sur l'ergodisme des signaux mis en jeu (cf paragraphe 1.5.4).

Le critère des moindres carrés

Le vecteur des paramètres, ici les coefficients du filtre RIF, est toujours noté $\theta = (w_0, \dots, w_{M-1})^T$. Le modèle de liaison entre le signal (x_n) à estimer, le signal (u_n) d'observation à filtrer et l'estimateur y_n est décrit par le modèle de régression suivant :

$$x_n = \varphi_n^T \theta_* + b_n \quad y_n = \varphi_n^T \theta \quad (1.14)$$

où $\varphi_n = U_n = (u_n, u_{n-1}, \dots, u_{n-M+1})^T$ est le vecteur d'observation, appelé dans le cas des moindres carrés le *vecteur de régression* et b_n représente le bruit du modèle de régression, ou bruit de sortie du filtre optimal θ_* .

Moindres carrés : résolution et interprétation

Il s'agit d'une méthode d'optimisation «en bloc», version déterministe de l'optimisation probabiliste de Wiener : on suppose connues les données (x_k, u_k) sur un bloc de taille n , et on minimise le critère géométrique $J_n(\theta)$ sur l'erreur $e_k = x_k - \varphi_k^T \theta$:

$$J_n(\theta) = \frac{1}{n} \sum_{k=1}^n e_k^2. \quad (1.15)$$

Le lien avec la minimisation de $J(\theta) = \mathbb{E}(e_n^2)$ pour le filtre de Wiener, est fait par l'ergodisme (loi des grands nombres) $J(\theta) = \lim_{n \rightarrow \infty} J_n(\theta)$ (cf paragraphe 1.5.4).

On travaille donc dans l'espace euclidien $H = \mathbb{R}^n$ muni du produit scalaire

$$\langle X, Y \rangle = X^T Y = \sum_{i=1}^n X_i Y_i. \quad (1.16)$$

Introduisons les vecteurs de \mathbb{R}^n :

$$X = (x_1, \dots, x_n)^T \quad Y = (y_1, \dots, y_n)^T \quad B = (b_1, \dots, b_n)^T \quad E = (e_1, \dots, e_n)^T$$

la base de l'espace d'observation obtenue par décalage temporel des vecteurs

$$\Phi_i = [u_{1-i}, \dots, u_{n-i}]^T, \quad i = 0..M-1$$

et l'espace d'observation

$$\Phi = [\Phi_0, \dots, \Phi_{M-1}] \quad \text{matrice des observations,} \quad (1.17)$$

l'écart des moindres carrés s'écrit alors

$$J_n(\theta) = \frac{1}{n} \sum_{k=1}^n e_k^2 = \frac{1}{n} \|X\|^2 - 2r_n^T \theta + \theta^T R_n \theta \quad (1.18)$$

fonction quadratique de θ , avec les matrices et vecteurs

$$R_n = \frac{1}{n} \Phi^T \Phi = \frac{1}{n} \sum_1^n \varphi_k \varphi_k^T, \quad (1.19)$$

$$r_n = \frac{1}{n} \Phi^T X = \frac{1}{n} \sum_1^n x_k \varphi_k. \quad (1.20)$$

Ces matrices et vecteurs ont une interprétation statistique simple : ce sont les estimateurs empiriques de la matrice $\mathbf{R} = \mathbb{E}(\varphi_n \varphi_n^T)$ et du vecteur $\mathbf{p} = \mathbb{E}(x_n \varphi_n)$ intervenant dans les équations du filtrage optimal de Wiener (1.12). La fonction quadratique $J_n(\theta)$

a un minimum unique obtenu en annulant le gradient $\nabla J_n(\theta) = 2(R_n\theta - r_n)$ d'où les *équations normales empiriques* :

$$R_n\theta_n = r_n. \quad (1.21)$$

Cette méthode des moindres carrés est une méthode en bloc de traitement des données observées sur $[1, \dots, n]$. Elle fournit une suite θ_n d'estimateurs du paramètre optimal θ_* de Wiener défini par (1.12). On verra au chapitre 2 comment la transformer en un algorithme adaptatif, récursif, de traitement en ligne des données. On voit bien que la convergence des solutions de (1.21) vers celle de (1.12) découle des propriétés d'ergodisme au second ordre des signaux traités (x_n, u_n) .

1.4. Optimisation déterministe : les algorithmes

Nous avons vu comment les problèmes de traitement du signal se ramènent à l'optimisation d'un critère de coût fonction d'un vecteur de paramètres. Le cas du filtrage de Wiener RIF, notre problème de base, conduit à une fonction $J(w)$ quadratique. La solution en est facile : c'est l'annulation du gradient, donc un système linéaire en w (équations normales). Cependant la mise en œuvre adaptative passe par une démarche récursive, basée sur les algorithmes numériques d'optimisation. Nous en étudions ici les principaux représentants, qui servent de modèles pour la construction des algorithmes adaptatifs (chapitres 2 et 3).

De plus, ces démarches récursives sont indispensables dans les cas d'optimisation d'une fonction non-quadratique qui ne se ramènent pas à la résolution d'un système linéaire (chapitre 3).

1.4.1. Gradient déterministe

On cherche à minimiser, par un algorithme itératif, la fonction de coût $J(w) = \sigma_x^2 - 2\mathbf{p}^T w + w^T \mathbf{R}w$. On a $\nabla J(w) = 2(\mathbf{R}w - \mathbf{p})$. L'algorithme du gradient à pas μ_n pour minimiser $J(w)$ consiste à se déplacer, à partir de l'estimation w_{n-1} , dans la direction de plus grande pente $-\nabla J(w_{n-1})$, ce qui donne la forme itérative suivante :

$$w_n = w_{n-1} + \mu_n(\mathbf{p} - \mathbf{R}w_{n-1}) \quad (GD). \quad (1.22)$$

On part d'un vecteur initial w_0 arbitraire (en général, en l'absence d'informations préalables, $w_0 = 0$) et on étudie la convergence de (GD) vers le filtre optimal w_* défini par les *équations normales* :

$$\mathbf{R}w_* = \mathbf{p}. \quad (1.23)$$

Raisonnons sur l'écart à l'optimum $V_n = w_n - w_*$ pour lequel on obtient, par substitution de (1.23) dans (GD), la récurrence linéaire homogène

$$V_n = (I - \mu_n \mathbf{R})V_{n-1}. \quad (1.24)$$

La matrice d'autocorrélation \mathbf{R} est positive, supposons la définie positive, c'est-à-dire inversible, afin de garantir l'existence d'une solution unique w_* et notons $0 < \lambda_1 \leq \dots \leq \lambda_M$ ses valeurs propres.

Algorithme à pas décroissant

On prend μ_n suite positive décroissante vers 0. Pour n assez grand, les valeurs propres de $I - \mu_n \mathbf{R}$ sont $\gamma_k = 1 - \mu_n \lambda_k > 0$ donc $\|I - \mu_n \mathbf{R}\| = 1 - \mu_n \lambda_1$ et

$$\|V_n\| \leq \prod_{k=n_0}^n (1 - \mu_k \lambda_1) \|V_{n_0}\| = r_n \|V_{n_0}\|,$$

l'égalité ayant lieu pour le vecteur propre de λ_1 , la convergence vers 0 de V_n aura lieu si et seulement si $\lim_{n \rightarrow +\infty} r_n = 0$ c'est-à-dire

$$\prod_{k=n_0}^{\infty} (1 - \mu_k \lambda_1) = 0 \Leftrightarrow \sum_1^{\infty} \mu_k = +\infty. \quad (1.25)$$

Elle est donc assurée si $\mu_k \approx 1/k$, mais s'avère très lente (non géométrique) puisque $\lim r_{n+1}/r_n = \lim(1 - \mu_n \lambda_1) = 1$. On vérifie par exemple que $r_n = 1/n$ lorsque $\mu_k \lambda_1 = 1/k$. Ceci illustre la lenteur de la décroissance vers 0.

Algorithme à pas constant μ

Pour un pas constant, la récurrence (1.24) se résout en

$$V_n = (I - \mu \mathbf{R})^n V_0 \Rightarrow \|V_n\| \leq \beta(\mu)^n \|V_0\| \quad (1.26)$$

avec $\beta(\mu) = \|I - \mu \mathbf{R}\| = \max_k |1 - \mu \lambda_k|$. La condition de convergence est donc ici

$$\beta(\mu) < 1 \Leftrightarrow 0 < \mu < \frac{2}{\lambda_{max}(\mathbf{R})} = \frac{2}{\lambda_M}. \quad (1.27)$$

On aura alors une *convergence linéaire de taux* $\beta(\mu)$. Ce taux se calcule aisément en remarquant que le maximum des $|1 - \mu \lambda_i|$ ne peut être que $1 - \mu \lambda_1$ ou $\mu \lambda_M - 1$, la séparation se faisant pour l'égalité : $\mu = \mu_{opt} = \frac{2}{\lambda_1 + \lambda_M}$, le taux de convergence optimal est donc

$$\beta_{opt} = \beta(\mu_{opt}) = \frac{\lambda_M - \lambda_1}{\lambda_M + \lambda_1} = \frac{C - 1}{C + 1} \quad \text{avec } C = \text{cond}_2(\mathbf{R}) = \frac{\lambda_M}{\lambda_1}. \quad (1.28)$$

La conclusion, particulièrement importante du point de vue des applications temps réel, est donc que *la vitesse de convergence du gradient à pas fixe μ est fortement dépendante à la fois du choix du pas de calcul μ , et du conditionnement de la matrice d'autocorrélation R qui limite la vitesse maximale possible.*

Courbe d'apprentissage du Gradient

Au lieu de caractériser la convergence de l'algorithme par l'écart $\|V_n\|$, il est fréquent de le faire par la courbe de décroissance vers J_{\min} de la fonction de coût $J(n) = J(w_n)$. La fonction de coût quadratique s'écrit

$$J(n) = J(w_n) = J_{\min} + V_n^T \mathbf{R} V_n.$$

On décompose la matrice \mathbf{R} dans sa base propre orthonormée $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, on définit le filtre réduit $\omega_n = \mathbf{Q}V_n$, et $\gamma_i = 1 - \mu\lambda_i$ les v.p. de $I - \mu\mathbf{R}$. On obtient

$$J(n) - J_{\min} = \omega_n^T \mathbf{\Lambda} \omega_n = \sum_{i=1}^M \lambda_i \omega_0(i)^2 \gamma_i^{2n}, \quad (1.29)$$

cette courbe est donc un mélange d'exponentielles (les différents *modes* propres, de taux γ_i^2). Quand $n \rightarrow \infty$, la partie prépondérante est donnée par le $|\gamma_i|$ maximum, et la *constante de temps* de l'algorithme τ est définie par

$$J(n) - J_{\min} \approx A.e^{-n/\tau} = \lambda_k \omega_0(k)^2 \beta^{2n} \Rightarrow \tau = \frac{-1}{2 \log \beta}. \quad (1.30)$$

CAS DU PETIT PAS : pour $\mu \rightarrow 0$, on a $\beta = 1 - \mu\lambda_1 \rightarrow 1$ (convergence très lente) et $\log(1 - \mu\lambda_1) \approx -\mu\lambda_1$, donc $\tau \approx \frac{1}{2\mu\lambda_1}$. Les propriétés du (GD) à pas fixe sont résumées dans le tableau suivant :

Condition de convergence	$0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{R})}$
Taux de convergence	$\beta(\mu) = \max 1 - \mu\lambda_i < 1$
Pas optimal	$\mu_{opt} = \frac{2}{\lambda_1 + \lambda_M}$
Vitesse et conditionnement	$\beta_{opt} = \frac{Cond(\mathbf{R}) - 1}{Cond(\mathbf{R}) + 1}$
Constante de temps	$\tau = \frac{-1}{2 \log \beta}$, petit pas : $\tau \approx \frac{1}{2\mu\lambda_1}$

Tableau 1.1. Propriétés du Gradient déterministe à pas fixe μ

1.4.2. Newton et quasi-Newton

Les algorithmes de gradient ont une faiblesse irréductible : ils peuvent devenir très lents si le Hessien est mal conditionné. Les méthodes newtoniennes visent à résoudre ce problème. Pour une fonction de coût générique $J(\theta)$ de classe C^2 , la méthode de Newton résout par linéarisation le système d'équations non-linéaires $\nabla J(\theta) = 0$. Cela donne l'algorithme itératif suivant :

$$\theta_k = \theta_{k-1} - \nabla^2 J(\theta_{k-1})^{-1} \cdot \nabla J(\theta_{k-1}). \quad (1.31)$$

Bien sûr, dans le cas d'une fonction de coût quadratique, il donne la solution exacte du premier coup puisqu'il équivaut à la résolution des équations normales (1.12). Cet algorithme est rapide, indépendamment du conditionnement du Hessien, sa vitesse est quadratique, c'est-à-dire que l'erreur ϵ_k décroît selon une loi du type $\epsilon_{k+1} \leq C\epsilon_k^2$. Revers de la médaille, il ne possède pas la propriété de descente, c'est-à-dire qu'il n'existe pas de garantie de décroissance de $J(\theta)$ au cours des itérations ; il risque donc de diverger.

Pour cette raison, on le remplace par des algorithmes quasi-Newtoniens, qui rétablissent cette propriété de descente :

$$\theta_k = \theta_{k-1} - \mu_k S_k \nabla J(\theta_{k-1}) \quad (1.32)$$

où S_k est une suite de **gains matriciels**, matrices positives approchant $\nabla^2 J(\theta_*)^{-1}$. La positivité des matrices S_k garantit la propriété de descente, avec le choix des pas scalaires μ_k adéquats, enfin la convergence vers l'inverse du Hessien donne une vitesse asymptotique au moins *superlinéaire* (linéaire de taux β_k tendant vers 0), indépendante du conditionnement du Hessien.

Illustrons cette démarche sur la méthode des moindres carrés : soit θ_{n-1} la solution des moindres carrés à l'itération $n - 1$. On applique la méthode de Newton sur $J_n(\theta)$ pour calculer la nouvelle itération θ_n , on a

$$\nabla J_n(\theta) = 2(R_n \theta - r_n) \quad \nabla^2 J_n(\theta) = 2R_n$$

d'où le calcul du gradient

$$\nabla J_n(\theta_{n-1}) = 2(R_n \theta_{n-1} - r_n) = -\frac{2}{n} \varphi_n e_n$$

et la méthode de Newton donne ici la solution exacte en une itération :

$$\theta_n = \theta_{n-1} + \frac{1}{n} R_n^{-1} e_n \varphi_n. \quad (1.33)$$

Nous obtenons ainsi un algorithme de moindres carrés récursifs, qui est étudié au chapitre 2 en détail.

On peut aussi l'interpréter comme une version stochastique d'une méthode quasi-Newton appliquée à la fonction d'EQM de Wiener $J(\theta) = \mathbb{E}(e_n(\theta)^2)$: en effet on a $\frac{1}{2} \nabla J(\theta) = -\mathbb{E}(e_n \varphi_n)$, donc $e_n \varphi_n$ est une approximation stochastique de l'opposé du gradient, tandis que par ergodisme des signaux à l'ordre 2 on aura la convergence $R_n^{-1} \rightarrow \mathbf{R}^{-1} = \frac{1}{2} \nabla^2 J(\theta)^{-1}$.

L'approche newtonienne donne la solution exacte des moindres carrés dans le cas (spécifique à Wiener-RIF) d'une fonction de coût quadratique. Appliquée à des situations plus générales (cf chapitre 3) elle donne un algorithme itératif approché. On établit pour les méthodes quasi-Newton une vitesse de convergence au moins superlinéaire, et en général quadratique (cf [Minoux 83]).

1.4.3. Autres méthodes : gradient conjugué...

L'inconvénient des méthodes quasi-Newton est leur complexité spatiale et temporelle importante en $O(M^2)$ par itération si M est la taille du vecteur de paramètres à estimer. Il en va ainsi pour les moindres carrés récursifs³. Des alternatives intermédiaires entre les gradients et quasi-Newton sont utiles. Certaines sont développées au chapitre 2 (méthodes de projection) ou au chapitre 4 (algorithmes rapides). Nous citerons ici pour référence les algorithmes de Gradient-conjugué.

Le principe en est, pour une fonction f quadratique de Hessien A , de construire en M itérations une suite d_k de directions conjuguées (c'est-à-dire une base orthogonale pour la métrique $x^T Ax$), et donc de résoudre exactement le problème linéaire $\nabla f(x) = Ax - b = 0$ par combinaison linéaire itérative des directions d_k . Pour une fonction f quelconque, cela donne l'algorithme :

Algorithme de Fletcher-Reeves

- 1) Choix x_0 , direction initiale $d_0 = -\nabla f(x_0)$.
- 2) Itération : $x_{k+1} = x_k + \mu_k d_k$ avec le pas μ_k minimisant $f(x_k + \mu d_k)$.
- 3) Nouvelle direction : $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$ avec $\beta_k = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$.

On établit pour cet algorithme une vitesse de convergence au moins superlinéaire en M étapes, voire quadratique en M étapes (donc facteur M de ralentissement par rapport à quasi-Newton, cf [Minoux 83]).

1.5. Passage aux algorithmes adaptatifs et rôle de l'ergodisme

Les objectifs de l'adaptatif. Typiquement, l'adaptatif permet de poursuivre un système linéaire qui varie au cours du temps. Les aspects non stationnaires et non linéaires peuvent être liés : un système non linéaire dont le point de fonctionnement varie peut être approché par un système linéaire évolutif. Il est bien sûr possible d'étendre les procédures adaptatives aux problèmes d'estimation non linéaires. Nous terminons ici, en l'illustrant sur le problème de base (filtrage de Wiener RIF), la démarche générale de conception d'un algorithme adaptatif. Les phases détaillées de construction et d'analyse des performances d'un tel système font l'objet du chapitre 2 pour le problème de base et du chapitre 3 pour le cas général.

La méthodologie pour passer d'un critère à un algorithme adaptatif se compose des trois phases suivantes :

- formuler le problème en terme d'optimisation d'un critère,
- choisir un algorithme itératif «déterministe» pour la minimisation,

3. RLS : Recursive Least Squares

- passer à une version stochastique de traitement en ligne des données.

Nous détaillons maintenant ces différentes étapes pour finalement aborder le rôle de l'ergodisme dans le fonctionnement d'un algorithme stochastique.

1.5.1. *Choix du critère à optimiser*

L'objectif à atteindre peut être défini essentiellement de deux manières :

Choix d'une fonction de coût qui dépend des paramètres à estimer. Cette fonction de coût doit être minimale lorsque ces paramètres atteignent leur valeur optimale. Comme la plupart des problèmes peuvent être formulés de cette manière et que cette approche permet d'exploiter directement les algorithmes classiques de l'analyse numérique, il s'agit de la méthode la plus fréquemment utilisée. Comme nous l'avons dit, le coût quadratique est le plus souvent retenu du fait de la simplicité des algorithmes qui en découlent. Dans certains problèmes cependant, notamment des problèmes d'égalisation aveugle, l'utilisation d'une fonction quadratique ne permet pas d'aboutir et il est indispensable d'utiliser d'autres coûts ; la mise en œuvre et l'étude des algorithmes obtenus s'en trouvent considérablement compliquées. L'une des difficultés majeures, liée aux coûts non quadratiques, réside dans l'existence de minima locaux dans lesquels la procédure de recherche de minimum risque de s'égarer.

Choix d'une fonction d'erreur qui s'annule lorsque les paramètres à identifier atteignent leur valeur optimale. Dans cette optique, l'optimum est vu comme un point stationnaire de l'algorithme adaptatif. Choisir une fonction d'erreur constitue une méthode plus générale que choisir une fonction de coût. En effet, tous les signaux d'erreur ne dérivent pas d'un potentiel alors que de tout potentiel dérive un signal d'erreur⁴. Certains algorithmes de séparation de sources par exemple ne peuvent pas être vus comme des procédures de minimisation, ils sont uniquement définis à partir de leurs points stationnaires.

L'exemple suivant illustre le choix d'une fonction de coût ou d'une mesure d'erreur dans le cas simple de l'estimateur de moyenne.

EXEMPLE 1 : ESTIMATION D'UNE MOYENNE. Si X_n est une suite i.i.d. de v.a. dont on souhaite estimer la moyenne, la fonction de coût :

$$J(\theta) = E(X_n - \theta)^2 \quad (1.34)$$

4. Dans \mathbb{R}^3 par exemple, une fonction à valeurs réelles dont le rotationnel est non nul ne dérive pas d'un potentiel, on ne peut donc pas en rechercher les zéros par une méthode de minimisation puisque ses minima n'annulent aucun gradient.

est minimale lorsque $\partial J(\theta) / \partial \theta = 0$, c'est-à-dire lorsque $\theta = \theta_* = E(X_n)$.

La fonction d'erreur :

$$H(\theta, X_n) = X_n - \theta$$

est nulle en moyenne pour $\theta = \theta_* = E(X_n)$, c'est à dire $E(X_n - \theta_*) = E H(\theta_*, X_n) = 0$. Ainsi, l'algorithme :

$$\theta_n = \theta_{n-1} + \mu_n E(X_n - \theta_{n-1})$$

atteint un état d'équilibre lorsque θ est égal à la moyenne de X_n . La stabilité de l'équilibre est liée au fait que la fonction $J(\theta)$ est minimale en θ_* . De façon équivalente, la stabilité de l'équilibre est liée au fait que $\partial E H(\theta, X_n) / \partial \theta = -1 \leq 0$ en θ_* .

1.5.2. Choix d'un algorithme déterministe d'optimisation

Les grandes classes d'algorithmes ont été données ci-dessus en section 1.4. Le choix se fait essentiellement entre gradient et quasi-Newton, en fonction des contraintes de temps de calcul et de complexité d'implémentation.

Prise en compte de contraintes

Une fonction de coût étant définie, il arrive fréquemment que des contraintes supplémentaires apparaissent dans un problème de minimisation. Les trajectoires des procédures de recherche de minimum doivent alors être astreintes à rester sur une certaine variété : elles ne peuvent plus se déplacer librement dans la totalité de l'espace des paramètres du filtre. L'utilisation d'une procédure telle que le gradient ne garantit évidemment pas cette condition. La recherche de minimum sous contrainte est bien connue, elle fait appel à la méthode de Lagrange. Dans le cas des algorithmes itératifs, il est nécessaire de procéder au calcul différentiel sur des variétés ou d'effectuer des projections de façon à s'assurer que la trajectoire appartient toujours à la variété de contrainte. A titre d'illustration, nous donnons maintenant quelques exemples d'utilisation courante.

EXEMPLE 2 : ALGORITHME DU GRADIENT SUR LA SPHÈRE UNITÉ EN ÉGALISATION Il est parfois nécessaire de ne pas modifier l'énergie globale contenue dans le signal. Dans ce cas, le signal sortant de l'égaliseur doit être normé. L'une des façons d'utiliser l'algorithme du gradient sous cette contrainte consiste à effectuer à chaque itération une opération du type : $\theta_n \leftarrow \frac{\theta_n}{\|\theta_n\|}$. Cette façon de procéder fonctionne à la condition que le signal d'entrée de l'égaliseur soit blanc. Ceci n'est évidemment pas le cas en général, il faut donc assurer un blanchiment préalable (à l'aide d'un prédicteur linéaire par exemple). Cette procédure de minimisation sous contrainte a été introduite par [Benveniste]. Elle a ensuite été utilisée par [Shalvi et Weinstein] pour construire un algorithme d'égalisation aveugle.

Le problème des minima locaux

Lorsque le seul minimum de la fonction de coût choisie est celui qui correspond à l'égaliseur optimal (ceci est en particulier le cas pour des fonctions convexes, *a fortiori* pour les fonctions quadratiques), les algorithmes de minimisation classiques conduisent à ce minimum.

En général, la fonction de coût choisie peut comporter des minima locaux. Dans ce cas, l'utilisation des algorithmes précédents ne conduit pas forcément à la bonne valeur pour l'égaliseur. Il n'existe aucune solution générale à ce problème. Seules quelques astuces permettent de favoriser le bon fonctionnement de l'algorithme. Par exemple, une initialisation judicieuse — proche du minimum global — de l'algorithme ou bien encore la prise en compte de contraintes supplémentaires (judicieusement choisies, elles peuvent permettre d'améliorer la situation, voire parfois de supprimer les minima parasites).

1.5.3. Algorithme stochastique et estimation statistique

Abordons ici la phase finale : l'approximation stochastique du gradient. Reprenons pour cela l'exemple du gradient déterministe pour notre problème de base, défini en (1.22). L'algorithme travaille sur la fonction d'EQM théorique $J(w) = E(e_n^2)$ et utilise les statistiques des signaux, auto et intercorrélations (\mathbf{R}, \mathbf{p}) . On peut aussi le réécrire explicitement avec l'opérateur d'espérance mathématique

$$w_n = w_{n-1} + \mu_n E(e_n U_n) \quad (1.35)$$

et l'on voit qu'il ne s'agit pas encore d'un algorithme adaptatif de traitement des données en ligne. L'approximation stochastique va modifier l'expression exacte du gradient $-\frac{1}{2}\nabla J(w) = E(e_n U_n)$ en supprimant l'espérance mathématique E (non réalisable sur les données) : ce qui revient à le remplacer par le *gradient instantané de l'erreur* $-\frac{1}{2}\nabla e_n^2 = e_n U_n$. On obtient alors un algorithme de gradient stochastique (étudié en détail au chapitre 2 — LMS —)

$$w_n = w_{n-1} + \mu_n e_n U_n \quad (1.36)$$

que l'on voit ici comme une version simplifiée de l'algorithme de moindres carrés récurrents (1.33). Tous les deux utilisent ce gradient stochastique instantané. Nous tirons de ces deux formes particulières, l'écriture d'une forme générale d'un algorithme adaptatif.

La forme générale est donc itérative et traitant les données, à mémoire finie :

$$\theta_n = \theta_{n-1} + \mu_n H(\theta_{n-1}, X_n). \quad (1.37)$$

On voit apparaître les deux constituants essentiels de l'algorithme :

- le vecteur d'état $X_n \in \mathbb{R}^d$, mémoire finie incluant les données du passé et les données interne à l'algorithme, nécessaires à l'itération,
- le champ de vecteurs de l'algorithme, fonction déterministe $H(\theta, X)$ à valeurs dans \mathbb{R}^m

Le schéma de cet algorithme est donné par la figure 1.5.

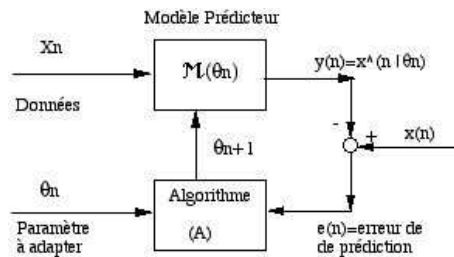


Figure 1.5. Algorithme adaptatif pour identifier θ .

La caractéristique principale est le caractère aléatoire de la suite θ_n générée par l'algorithme, directement dirigée par les données aléatoires traitées. La fluctuation aléatoire liée à l'approximation stochastique s'ajoute ici à l'approximation numérique du paramètre optimal θ_* par l'algorithme déterministe. Le lien entre le comportement des algorithmes déterministe et stochastique est évidemment le problème crucial et difficile du traitement adaptatif : il fait l'objet des chapitres 2 et 3.

Pour l'étude des performances de tels algorithmes stochastiques, il est utile de les interpréter comme la construction d'une suite θ_n d'estimateurs du vecteur déterministe objectif θ_* . On s'intéresse donc à différentes propriétés de ces estimateurs statistiques :

Convergence (ou consistance) : $\theta_n \rightarrow \theta_*$ (au sens p.s. ou m.q.)

Erreur quadratique, covariance : $\Gamma_n = E((\theta_n - \theta_*)(\theta_n - \theta_*)^T)$

Efficacité asymptotique : $\Gamma_n \rightarrow I_n(\theta)^{-1}$ (Borne de Cramer-Rao)

Normalité asymptotique : $\sqrt{(n)}(\theta_n - \theta_*) \rightarrow \mathcal{N}(0, \Gamma)$ (Tendance vers une loi normale, théorème de limite centrale)

Pour les méthodes générales d'estimation statistique, voir [Dacunha-Castelle 82] et [Monfort 80].

1.5.4. Le lien entre point de vue déterministe et probabiliste : l'ergodisme.

On espère que l'algorithme adaptatif, version stochastique d'un algorithme déterministe, atteindra asymptotiquement le même comportement optimal pour le filtrage des données. Ceci ne peut se produire que si les *moyennes temporelles* réalisées par les itérations de l'algorithme convergent vers le vrai gradient, en réalisant l'opérateur E d'espérance mathématique qu'on a supprimé. Ce sont donc les propriétés d'*ergodisme* des signaux aléatoires traités, généralisant la **loi des grands nombres** chère au statisticien, qui seront déterminantes. Nous en résumons ici les principales variantes, qui seront utilisées dans les études de performances des algorithmes développées dans la suite de l'ouvrage. On se limite ici au cas de signaux scalaires et stationnaires, ceci se généralise aux cas vectoriels et non-stationnaires.

Ergodisme de la moyenne

Le signal traité est modélisé par un processus aléatoire $x : \mathbb{Z} \times \Omega \rightarrow \mathbb{C}$ défini sur l'espace de probabilité Ω . On le suppose ici centré, et stationnaire au sens large (invariance par translation du moment d'ordre 2 : $\Gamma_x(n, n-p) = \gamma_x(p)$), ou au sens fort (invariance par translation de la loi du processus). On a alors le théorème de base [Dacunha-Castelle 82]

THÉORÈME.— *Ergodisme au premier ordre*

L'estimateur empirique de la moyenne $\hat{m}_x(n) = \frac{1}{n} \sum_1^n x_k$ est non biaisé. Il est convergent en m.q. si et seulement si la densité spectrale $\overset{\circ}{\gamma}_x(\nu)$ ne présente pas de raie en 0, et alors on a

$$\text{var}(\hat{m}_x(n, \omega)) \approx \frac{1}{n} \sum_p \gamma_x(p) = \frac{1}{n} \overset{\circ}{\gamma}_x(0) \quad \text{quand } n \rightarrow +\infty. \quad (1.38)$$

Ergodisme du second ordre

On a vu que la plupart des traitements optimaux envisagés réalisent de l'ELMQ, et utilisent les statistiques d'ordre deux des signaux. Donc la propriété minimale d'ergodisme requise pour la convergence des algorithmes adaptatifs est l'ergodisme au second ordre (ou ergodisme de la covariance) au sens fort ou faible défini par

$$\frac{1}{n} \sum_{k=1}^n x_k x_{k-p}^* \rightarrow \text{E}(x_k x_{k-p}^*) = \gamma_x(p) \quad \text{p.s. ou m.q.} \quad (1.39)$$

Pour obtenir ce résultat, il suffit d'appliquer le théorème précédent sur le processus produit $y_n = x_n x_{n-p}^*$. Il faut donc connaître ses propriétés à l'ordre deux, c'est-à-dire les moments d'ordre quatre de x . Un théorème plus fort de [Doob 52] s'applique aux signaux à corrélation décroissant vers 0.

THÉORÈME.— [Doob 52]

(i) Si la corrélation de y_n tend vers 0 à l'infini assez vite :

$$|\gamma_y(m)| \leq \frac{C}{m^\alpha} \text{ pour un } \alpha > 0$$

l'estimateur de la corrélation $\hat{\gamma}_x(p) = \frac{1}{n} \sum_1^n y_k$ est consistant

$$\hat{\gamma}_x(p) \rightarrow \gamma_x(p) \quad \text{p.s. et m.q.} \quad (1.40)$$

(ii) si de plus (x_n) est gaussien, l'ergodisme p.s. équivaut à

$$\frac{1}{n} \sum_1^n |\gamma_x(k)|^2 \rightarrow 0 \Leftrightarrow \text{la dsp } \overset{\circ}{\gamma}_x(\nu) \text{ ne possède aucune raie.} \quad (1.41)$$

On vérifie facilement la propriété (1.39) sur des signaux quasi-périodiques (à amplitude non aléatoire) tandis que les conditions de (1.40) sont vérifiées par les signaux AR, ARMA (filtrés de bruit blanc par un filtre rationnel stable), et plus généralement les processus à représentation markovienne stable (du type de Kalman (1.6)).

Processus ergodique

On peut définir une notion d'ergodisme plus forte, liée à la loi de probabilité complète du processus et non plus à ses moments. Pour un processus aléatoire strictement stationnaire $(x_n)_{n \in \mathbb{Z}}$, le point de vue général de la théorie ergodique s'applique à l'espace de probabilité $\{\Omega, P\}$ associé au processus (espace des suites numériques $\omega = (x_n)_{n \in \mathbb{Z}}$), avec la transformation S qui est le décalage temporel (le retard) : $Sx_n = x_{n-1}$ pour laquelle la probabilité P est invariante. On a alors le théorème ergodique.

THÉORÈME.— [Birkhoff 1931]

Pour toute fonction $f \in L^1(\Omega)$, les moyennes temporelles existent : soit $\omega = (x_n) \in \Omega$

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n f(x_k) = \bar{f}(\omega) \quad \text{existe p.s.} \quad (1.42)$$

et l'on dira que le processus est *ergodique* si ces moyennes temporelles sont constantes, indépendantes de la réalisation ω du processus. On montre que ceci équivaut à la propriété suivante pour les événements A de Ω :

$$A \text{ invariant} : \forall n, P(A \Delta S^n A) = 0 \Rightarrow P(A) = 0 \text{ ou } 1.$$

L'ergodisme est lié à des propriétés d'indépendance asymptotique ou de mélange.

Indépendance asymptotique, mélange

La loi forte des grands nombres (LGN) s'applique évidemment à un processus indépendant, ou bruit blanc fort. Elle s'étend comme on l'a vu ci-dessus à des cas de dépendance décroissant à l'infini. Diverses caractérisations peuvent en être données, qui peuvent d'ailleurs s'appliquer à des processus non-stationnaires. Citons les principales utilisées dans l'étude de convergence des algorithmes adaptatifs. Elles sont toujours liées à une mesure de la dépendance entre les tribus du passé $\mathcal{B}_n = \sigma(x_k, k \leq n)$ et du futur $\mathcal{F}_n = \sigma(x_k, k \geq n)$. Pour un événement B , on note $B^{(n)} = S^{-n}B$ l'événement décalé dans le futur de n .

M-indépendance :

$$n \geq M \Rightarrow \mathcal{B}_k \text{ est indépendante de } \mathcal{F}_{n+k}. \quad (1.43)$$

Indépendance asymptotique :

$$\rho(n) = \sup\{\rho_{uv}, u \in L^2(\mathcal{B}_0), v \in L^2(\mathcal{F}_n)\} \rightarrow 0 \text{ si } n \rightarrow \infty. \quad (1.44)$$

Mélange :

$$\forall A, B \in \mathcal{B}_\infty : \quad P(A \cap B^{(n)}) \rightarrow P(A)P(B) \quad \text{si } n \rightarrow \infty. \quad (1.45)$$

Notes

Nous avons abordé les grandes classes de problèmes d'optimisation qui vont intéresser le traiteur de signal, soit dans un but de *filtrage optimal* des signaux, soit dans un but de *modélisation paramétrique* des signaux.

La démarche la plus couramment utilisée est celle du *filtrage de Wiener transverse (RIF)*, car elle conduit au problème simple de minimisation d'une fonction quadratique, équivalent à la résolution d'un système linéaire. Son traitement adaptatif fait l'objet du chapitre 2.

Nous avons vu cependant d'autres types de représentation des filtres et des signaux, qui peuvent apporter des performances meilleures avec une économie de paramètres ou d'intéressantes propriétés de découplage des paramètres : ces méthodes conduisent à la minimisation de fonctions *non quadratiques* par rapport aux paramètres. Le traitement adaptatif de ces situations générales fait l'objet du chapitre 3.

Pour un panorama large des problèmes et méthodes optimales en traitement du signal, on se reportera à [Orfanidis 85]. Des développements sur les méthodes adaptatives en signal sont dans [Bellanger 85], [Widrow85], et une référence essentielle pour l'identification adaptative dans des contextes très généraux (modélisation, automatique et contrôle) est [Ljung 83].

Les outils essentiels de probabilités et statistique utiles sont dans [Bremaud 84], [Dacunha-Castelle 82] et [Monfort 80]. Pour les algorithmes d'optimisation en Analyse Numérique, on trouvera le point de vue algorithmique dans [Minoux 83], et une approche plus mathématique dans [Ciarlet 85]. Les fondements de la théorie des processus aléatoires, notamment stationnaires, sont dans les références classiques [Cramer 67] [Doob 53] et [Gikhman]. On y retrouvera les principaux résultats concernant l'ergodisme, dont un résumé est fait dans [Crepel 81].

1.6. Bibliographie

- BELLANGER, 1987, Adaptive digital filters and signal analysis, Dekker, New York.
 BREMAUD, 1984, Introduction aux Probabilités, Springer Verlag, Berlin.
 CHUI et CHEN, 1987, Kalman Filtering with real-time applications, Springer-Verlag, Berlin.
 CIARLET, 1985, Analyse Numérique Matricielle et Optimisation, Masson, Paris.
 CRAMER et LEADBETTER, 1967, Stationary and related stochastic processes, Wiley, New York.
 CREPEL, 1981, « Variables aléatoires dépendantes », in [Outils et Modèles Mathématiques pour l'Automatique et le Traitement du Signal (OMMATS) vol1 81], Editions du CNRS, Paris.
 DACUNHA-CASTELLE, 1982, Probabilités et Statistique, vol 1 et 2, Masson, Paris.
 DOOB, 1953, Stochastic Processes, Wiley, New York.
 LJUNG et SODERSTRÖM, 1983, Theory and Practice of Recursive Identification, MIT Press, Cambridge.
 MINOUX, 1983, Programmation mathématique, vol.1 , Dunod, Paris.
 MONFORT, 1980, Cours de Statistique Mathématique, Economica, Paris.
 ORFANIDIS, 1985, Optimum Signal Processing, MacMillan , New York.
 WIDROW and HOFF, 1960, "Adaptive switching circuits", IRE WESCON Conv. Rec.,Part4.
 WIDROW et al., 1967, "Adaptive antenna systems", Proc. IEEE vol. 55, p2143.
 WIDROW et al., 1975, "Adaptive noise cancelling", Proc. IEEE vol. 63, p1962.
 WIDROW et STEARNS, 1985, Adaptive Signal Processing, Prentice Hall, New Jersey.

Chapitre 2

Algorithmes pour le filtrage adaptatif RIF

CONTENU.— Ce chapitre développe les deux grandes classes d'algorithmes adaptatifs pour traiter en ligne le problème de base du filtrage de Wiener transverse ou RIF. Celles-ci sont issues des deux approches des méthodes numériques d'optimisation rappelées au chapitre 1 : méthodes de gradient, qui donneront ici l'algorithme LMS et ses variantes, méthodes newtoniennes qu'on obtiendra ici par la méthode des moindres carrés et le RLS. Les variantes intermédiaires (blocs, treillis, projection) sont développées. On étudie aussi la convergence des algorithmes et leurs performances comparées dans des contextes stationnaires. La capacité de poursuite en situation non-stationnaire sera abordée dans le contexte général du Chapitre 3.

2.1. Gradient stochastique

Le problème à traiter est le filtrage de Wiener d'ordre fini M (voir section 1.3.1). On cherche le vecteur du filtre transverse d'ordre M défini par l'EQM minimale : $w^* = \text{Arg min } J(w)$ avec

$$J(w) = \text{E}[e_n^2] = \sigma_x^2 - 2p^T w + w^T R w \quad (2.1)$$

L'algorithme LMS est une approximation stochastique de l'algorithme *du gradient déterministe (GD)* (1.22) appliqué à la minimisation de la fonction de coût quadratique $J(w)$: celui-ci a été étudié en détail au Chapitre 1 (voir section 1.4.1).

Remarque : l'appellation LMS consacrée dans la littérature, issue de l'anglais (*Least Mean Square*), se réfère uniquement au but de l'algorithme (minimiser l'écart

quadratique moyen $J(w)$ et pas du tout à sa nature. L'appellation correcte est donc celle de Gradient stochastique (GS) qui explicite la double origine de l'algorithme (méthode de gradient et approximation stochastique).

La démarche et la dénomination même d'*approximation stochastique*, ont été introduites au départ par [Robbins, Monro 51] sur un problème statistique d'estimation de la moyenne d'une fonction de variable aléatoire monodimensionnelle. L'application au contexte du filtrage optimal de Wiener est sensiblement plus complexe à analyser : c'est un aspect paradoxal de cet algorithme, dont la mise en œuvre effective et l'efficacité pratique ont été développées de longue date par les ingénieurs alors que l'obtention de résultats mathématiques de convergence est plus tardive et nécessite des techniques difficiles.

2.1.1. Le gradient stochastique : LMS

Le passage de l'algorithme du (GD) (1.22) à un algorithme adaptatif (ou stochastique) de traitement en ligne des données passe par l'élimination de l'opérateur E d'espérance mathématique : c'est l'*approximation stochastique du gradient*, qui revient aussi à remplacer le gradient de l'EQM par le gradient de l'erreur quadratique instantanée $\nabla e_n^2 = -2e_n U(n)$. L'exemple très simple d'estimation d'une moyenne (Chapitre 1, p34) met en évidence le lien entre cette démarche récursive et la loi des grands nombres utilisée par le statisticien pour une estimation en bloc de l'espérance : ici c'est l'algorithme stochastique qui réalisera au cours des itérations le moyennage des données.

Estimation récursive de $E(X_n)$

Soit une suite de variables aléatoires de même loi $(X_n)_{n>0}$. Sous des conditions d'ergodisme (indépendance asymptotique entre X_m et X_{m+n} si $n \rightarrow \infty$), l'estimateur empirique de la moyenne sera convergent :

$$M_n = \frac{1}{n} \sum_{k=1}^n X_k \rightarrow a = E(X_n) \text{ p.s. si } n \rightarrow \infty$$

On peut aussi l'estimer en minimisant le coût quadratique $J(\theta) = E[(X_n - \theta)^2]$. On résout ce problème d'optimisation par la méthode du gradient stochastique : ici le gradient est la dérivée $J'(\theta) = -2E(X_n - \theta)$, on se déplacera de $d_k = X_k - \theta$, ce qui donne l'algorithme à pas décroissant $\mu_n = \frac{1}{n}$

$$\theta_n = \theta_{n-1} + \mu_n (X_n - \theta_{n-1}) = \frac{n-1}{n} \theta_{n-1} + \frac{X_n}{n} = M_n \text{ si } \theta_0 = 0 \quad (2.2)$$

On voit sur cet exemple que le gradient stochastique redonne ici le meilleur estimateur possible à savoir la *moyenne empirique d'échantillon*.

L'algorithme LMS

On met le (GD) (1.22) sous forme d'espérance, à partir des définitions de R et p :

$$\frac{1}{2} \nabla J(w) = Rw - p = E((U_n^T w - x_n)U_n) = -E(e_n U_n)$$

et on obtient par approximation stochastique du gradient

$$\mathbf{LMS} \quad w_n = w_{n-1} + \mu_n e_n U_n \quad (2.3)$$

Cet algorithme est d'une grande simplicité, ce qui en fait le plus populaire des algorithmes adaptatifs. Sa *complexité* est de M multiplications par itération (soit autant que le filtrage lui-même). On distinguera deux mises en œuvre distinctes :

- Algorithme à pas décroissant : $\mu_n \rightarrow 0$ lorsque $n \rightarrow \infty$
- Algorithme à pas constant : $\forall n, \mu_n = \mu > 0$

Illustrons le LMS sur un exemple d'application [ALE Adaptive Line Enhancer] : c'est le problème de l'extraction d'un signal périodique dans un signal à bande large . Dans la simulation présentée Figure 2.1, le signal est le suivant :

$$x_n = \sin(0.1 * n) + b_n \quad b_n \text{ bruit blanc de variance } \sigma^2 = 2, \quad u_n = x_{n-1}$$

et on calcule le filtre optimal de Wiener H d'ordre $M = 100$. Les résultats en terme de puissances des erreurs d'estimation sont les suivants :

$$\begin{aligned} \text{Erreur Wiener } E(s_n - z_n)^2 &= 0.04 & \text{Gain en RSB} &: 50 \\ \text{Erreur LMS } E(x_{lms} - s_n)^2 &= 0.08 & \text{Gain en RSB} &: 25 \end{aligned}$$

Le pas de calcul pour le LMS est $\mu = 10^{-4}$. Le gain en RSB de Wiener est donc de 50, tandis que celui du LMS est de 25 : ceci pourrait évidemment être amélioré en diminuant le pas μ et augmentant la taille n pour mieux faire converger le LMS vers l'optimum w^* de Wiener.

2.1.2. Convergence du LMS

Pour analyser la convergence du LMS, on introduit l'erreur de filtre $V_n = w_n - w^*$, solution de la récurrence linéaire (à coefficients matriciels, aléatoires, non stationnaires) :

$$V_n = (I - \mu_n U_n U_n^T) V_{n-1} + \mu_n b_n U_n \quad (2.4)$$

On peut interpréter ce système d'erreur du LMS comme un système dynamique linéaire, de matrices de transition $A_n = I - \mu_n U_n U_n^T$, excité par l'entrée externe

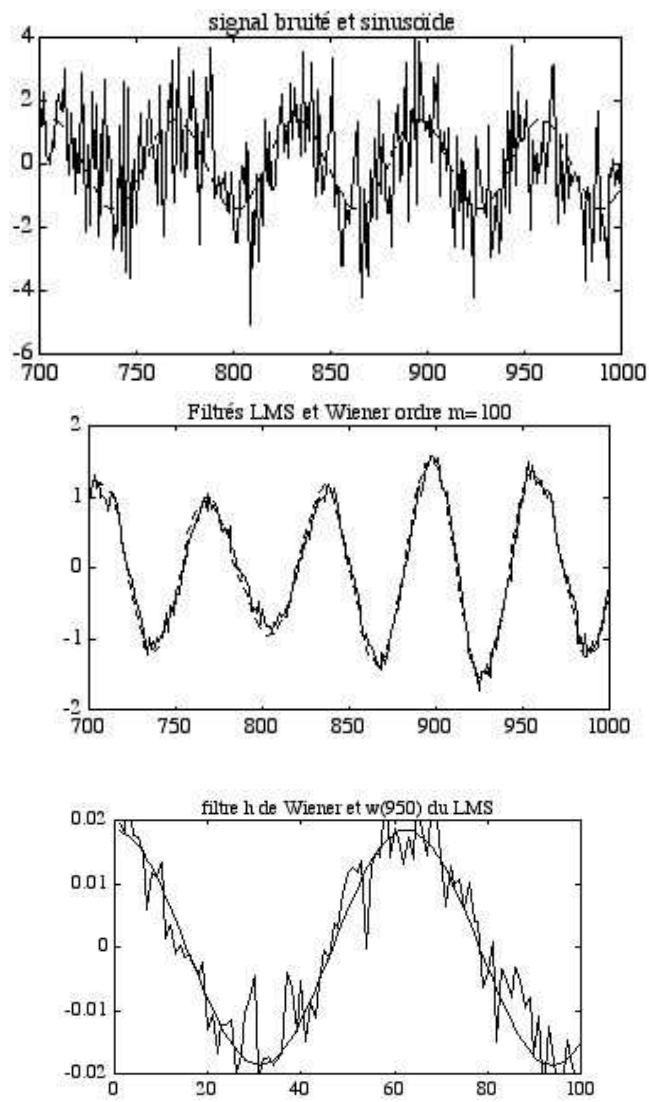


Figure 2.1. Extraction de sinusoïde par Wiener et LMS $M = 100$

vectorielle aléatoire $C_n = b_n U_n$ pondérée par μ_n . Rappelons qu'ici b_n est le bruit de sortie du filtre optimal de Wiener, caractérisé par la décorrélation $E(b_n U_n) = 0$, donc

$$V_n = A_n V_{n-1} + \mu_n C_n \quad \text{avec } E(A_n) = R \quad E(C_n) = 0 \quad (2.5)$$

La solution de cette récurrence linéaire s'écrit à l'aide des matrices de transition itérées $\Pi_{n,k} = A_n A_{n-1} \dots A_{k+1} = \prod_{j=k+1}^n A_j$ si $k \leq n$, et $\Pi_{n,k} = I$ si $k \geq n$:

$$V_n = \Pi_{n,0} V_0 + \Lambda_n \quad \text{avec } \Lambda_n = \sum_1^n \Pi_{n,k} \mu_k C_k \quad (2.6)$$

Ceci fait apparaître clairement les deux composantes de cette erreur :

– le *transitoire* $\Pi_{n,0} V_0$, effet de mémoire des conditions initiales, solution homogène de (2.4)

– la *fluctuation* Λ_n , effet cumulé de l'excitation aléatoire C_n , solution à conditions initiales nulles de (2.4)

Dans l'hypothèse d'indépendance des processus (u_n) et (b_n) , on déduit de (2.6) que $E(\Lambda_n) = 0$: la fluctuation est de moyenne nulle, après le transitoire le LMS est *non-biaisé*. Dans le cas général, il y a un biais du LMS et le lien entre trajectoire déterministe du (GD) et trajectoire aléatoire du LMS sera précisé au Chapitre 3.

On ne peut espérer une convergence vers 0 de l'erreur V_n que si la puissance moyenne d'excitation $P_n = E(\|C_n\|^2) = \mu_n^2 J_{\min} \text{Tr}(R) \rightarrow 0$ lorsque $n \rightarrow \infty$, c'est-à-dire pour l'algorithme à pas décroissant. Nous étudierons d'abord celui-ci, permettant d'atteindre strictement la performance optimale de Wiener dans un contexte stationnaire.

Convergence p.s. du LMS à pas décroissant

Des résultats généraux de convergence du LMS, sous des hypothèses réalistes de dépendance des signaux traités, ont été obtenus par [Farden 81] puis [Eweda-Macchi 84]. Nous donnerons le second résultat, en élargissant ses hypothèses applicables à des signaux stationnaires.

THÉORÈME. – (Eweda-Macchi 84)

Si le processus conjoint $\{u_n, b_n\}$ est strictement stationnaire et ergodique, possède des moments finis jusqu'à l'ordre 4 avec une matrice de covariance $R = E(U_n U_n^T) > 0$, on a pour la suite (w_n) du LMS (2.3) à pas décroissant $\mu_n = \frac{A}{B+n}$

$$w_n \rightarrow w^* \quad \text{avec probabilité 1} \quad (2.7)$$

L'hypothèse ergodique signifie que la tribu asymptotique (événements invariants par translation temporelle) est constituée d'événements de probabilité 0 ou 1 : elle est impliquée par les hypothèses de corrélation décroissante $\gamma_y(p) = O(|p|^{-\alpha})$ données dans [Eweda 84], mais vaut plus largement aussi pour des processus quasi-déterministes dont la corrélation est infinie.

La condition requise est en fait que tous les processus d'ordre deux sur les données $\{x_n u_{n-i}, \|x_n U_n\|, u_n u_{n-i}\}$ vérifient la loi forte des grands nombres.

En conclusion, pour toutes les classes de signaux rencontrées en pratique, l'algorithme LMS converge en contexte stationnaire vers l'optimum du filtre de Wiener w^* . La vitesse de cette convergence est donnée par celle du (GD), étudiée en section 1.4.1.

Stabilité exponentielle du transitoire du LMS à pas fixe μ

On se place maintenant dans le cas du LMS à pas fixe $\mu > 0$: on sait qu'il ne permettra plus une convergence stricte vers l'optimum w^* , mais lui seul permettra l'adaptativité du système, la capacité de suivre en permanence les variations temporelles lentes des signaux ou systèmes traités. C'est donc toujours cet algorithme qui sera mis en œuvre dans les applications réelles. On sépare l'étude du comportement entre le transitoire et la fluctuation résiduelle.

On développe $\Pi_{n,m} = \prod_{j=m+1}^n (I - \mu U_j U_j^T)$ en notant $F_j = U_j U_j^T$:

$$\Pi_{n,m} = I - \sum_j \mu F_j + \sum_{q=2}^{n-m+1} (\mu^q \sum_{i_1 < i_2 < \dots < i_q} F_{i_1} \dots F_{i_q}) \quad (2.8)$$

on montre ensuite que les termes d'ordre 2 et plus en μ sont négligeables devant les deux premiers : la convergence est donc guidée par $\|I - \sum_j \mu F_j\| = 1 - \mu \lambda_{\min}(\sum_j F_j)$ pour des pas suffisamment petits. On voit que la stabilité exponentielle sera liée à une propriété d'excitation persistante du signal u_n , s'exprimant sous une forme déterministe ou probabiliste du type

$$\forall m \quad \lambda_{\min}(\sum_{j=m+1}^{m+N} U_j U_j^T) \geq \delta > 0 \quad \text{ou} \quad \lambda_{\min}(E(U_j U_j^T)) \geq \delta \quad (2.9)$$

Par exemple pour le cas déterministe [Solo 95 p142]), si on suppose le signal borné selon $\|F_i\| < M$, on déduit de (2.8) que

$$\|\Pi_{k+N,k}\| \leq \|I - \mu \sum_{k+1}^{k+N} F_i\| + \sum_2^N \frac{(\mu M N)^j}{j!} \leq 1 - \mu \frac{\delta}{2}$$

à condition de prendre $\mu < \min(1/MN, 1/\delta, \delta/(2M^2N^2))$. Il en résulte la convergence exponentielle du transitoire du LMS avec un taux $\beta = (1 - \mu\frac{\delta}{2})^{1/N}$.

Pour des signaux aléatoires, ces propriétés sont des conséquences de la stationnarité, de la positivité de la covariance $R = E(U_n U_n^T) > 0$, et de l'ergodisme. On obtient alors le théorème suivant (cf [Bitmead 83], [Bitmead, Anderson 86]).

THÉORÈME.— Stabilité exponentielle du transitoire LMS

Si (u_n) est strictement stationnaire et ergodique, de covariance $E(U_n U_n^T) = R > 0$, si on a l'une des deux hypothèses

(H1) Signal borné $\|U_n\| < B$ et $\mu < \mu_0 = 2/B^2$ ou

(H2) (u_n) possède des moments finis de tous ordres et $\mu < \mu_0$

il y a stabilité exponentielle du transitoire, p.s. de taux $1 - \mu\lambda_{\min}(R)$ et en m.q. :

$$\begin{aligned} \forall \mu < \mu_0 \quad \forall \alpha < \lambda_{\min}(R), \quad \exists K > 0 : \quad \|\Pi_{n,0}\| \leq K(1 - \mu\alpha)^n \quad \text{p.s.} \\ \exists C_2, \delta_2 > 0 : \quad \forall \mu < 1/\delta_2 \quad E(\|\Pi_{n,0} U_n\|^2)^{1/2} \leq C_2(1 - \mu\delta_2)^n \end{aligned} \quad (2.10)$$

Ceci montre que le transitoire du LMS, pour μ assez petit, convergera encore vers 0 en suivant d'assez près la trajectoire du (GD) associé. Par ailleurs, des résultats de stabilité exponentielle pour les moments de $\|\Pi_{n,0}\|$ ont été obtenus par [Eweda-Macchi 83]. Ces majorations s'étendent aux matrices $\Pi_{n,k}$ et servent ensuite à contrôler la fluctuation du LMS.

Fluctuation résiduelle du LMS et EQMR

On se place ici dans les hypothèses de stabilité exponentielle du transitoire. Il est naturel de s'attendre à ce que la fluctuation Λ_n du LMS donnée par (2.6) converge vers une loi asymptotique stationnaire de variance finie proportionnelle au pas de calcul μ : c'est le meilleur résultat que l'on puisse attendre, puisqu'il ne peut y avoir de convergence stricte en raison de l'excitation aléatoire permanente du système. On notera dans la suite $\|X\|_2 = E(\|X\|^2)^{1/2}$ la norme quadratique du vecteur aléatoire X . On réécrit la fluctuation sous la forme

$$\Lambda_n = \sum_1^n \Pi_{n,k} d_k = \sum_0^{n-1} \Pi_{n,n-k} d_{n-k} \quad \text{avec } d_k = \mu b_k U_k$$

La version stationnaire de la fluctuation sera obtenue par la limite (à n fixé)

$$Z_n^m = \sum_0^m \Pi_{n,n-k} d_{n-k} \rightarrow Z_n = \sum_0^\infty \Pi_{n,n-k} d_{n-k} \quad \text{p.s. et en m.q.} \quad (2.11)$$

On a par (2.10) $\|\Pi_{n,n-k}d_{n-k}\|_2 \leq \mu C_2(1 - \mu\delta_2)^k$ d'où

$$\sum_0^\infty \|\Pi_{n,n-k}d_{n-k}\|_2 \leq \frac{C_2}{\delta_2} \Rightarrow Z_n^m \rightarrow Z_n \quad \text{m.q.}$$

de plus $\|Z_n^m - Z_n\|_2 \leq \frac{C_2}{\delta_2}(1 - \mu\delta_2)^m$, donc

$$\sum_{m=0}^\infty \|Z_n^m - Z_n\|_2^2 < \infty \Rightarrow Z_n^m \rightarrow Z_n \quad \text{p.s.}$$

Par le même raisonnement sur l'écart $\|\Lambda_n - Z_n\|_2$, on obtient le résultat suivant.

THÉORÈME.— [Bitmead 83],[Solo 89]

S'il y a stabilité exponentielle du transitoire (p.s. et en m.q.), si l'excitation $b_n U_n$ possède des moments d'ordre 2, la fluctuation Λ_n du LMS donnée par (2.6) converge p.s., en m.q. et donc en loi vers la **fluctuation stationnaire**

$$\lim_{n \rightarrow \infty} \|\Lambda_n - Z_n\| = 0 \quad \text{p.s. , m.q. avec } Z_n = \sum_0^\infty \Pi_{n,n-k}d_{n-k} \quad (2.12)$$

ERREUR D'AJUSTEMENT DU LMS.— Il est pratiquement utile de pouvoir mesurer l'effet de la fluctuation sur l'erreur quadratique du LMS. Pour le faire facilement à partir des résultats ci-dessus, on partira de [Macchi-Bellanger 87] avec une hypothèse :

(H3) "Régression linéaire" : le bruit (b_n) est blanc et indépendant du signal (u_n) (2.13)

Ceci est vrai dans les applications du type Annulation d'écho, Identification de système linéaire. On a alors orthogonalité de la suite $\{\Pi_{n,k}b_k U_k\}_k$ de la fluctuation (2.6). On en déduit, par stationnarité :

$$E(\|\Lambda_n\|^2) = \mu^2 J_{\min} \sum_{k=1}^n E(\|\Pi_{n,k}U_k\|^2) = \mu^2 J_{\min} \sum_{j=0}^{n-1} E(\|\Pi_{j,0}U_0\|^2)$$

d'où, par croissance de $E(\|\Lambda_n\|^2)$ et majoration exponentielle du moment d'ordre 2

$$E(\|\Lambda_n\|^2) \rightarrow F(\mu) \leq \mu K_2 \quad (2.14)$$

On a donc stabilité en m.q. de la fluctuation du LMS, pour un pas $\mu < \mu_0 = 1/\delta_2$, et une erreur quadratique de fluctuation stationnaire proportionnelle au pas de calcul μ .

Une fois passé le transitoire, cette fluctuation résiduelle stationnaire du filtre LMS va entraîner un *excès d'erreur quadratique moyenne* pour les signaux filtrés :

$$e_n = x_n - w_{n-1}^T U_n = x_n - U_n^T w^* - \Lambda_{n-1}^T U_n = b_n - f_n$$

où $f_n = \Lambda_{n-1}^T U_n$ est le signal d'erreur de fluctuation.

DÉFINITION.— L'erreur quadratique moyenne résiduelle (EQMR) du LMS est

$$EQMR \quad \delta^2 = \lim_{n \rightarrow \infty} E(f_n^2) = E[(U_n^T Z_{n-1})^2] \quad (2.15)$$

et représente l'excès d'eqm obtenu en stationnaire par le LMS par rapport au J_{\min} de Wiener. L'erreur d'ajustement est le nombre $\mathcal{M} = \frac{EQMR}{J_{\min}} < 1$, il représente en pourcentage la perte d'optimalité par rapport à Wiener.

La combinaison de la convergence en m.q. (2.14) et du calcul récurrent de $E(\|\Lambda_n\|^2)$ à partir de (2.6), associé aux hypothèses d'indépendance entre b_n et $\{u_n\}$, donne l'évaluation asymptotique

$$\delta^2 = \frac{\mu}{2} J_{\min} \text{Tr}(R) + \mu E(f_n^2 | \|U_n\|^2) \quad (2.16)$$

Cette équation définit implicitement la valeur de l'EQMR. On peut en tirer une valeur ou majoration explicite dans les cas suivants :

– Signal borné : $|u_n|^2 < B$, u_n de loi uniforme sur $[-\sqrt{B}, +\sqrt{B}]$ et donc $MB = 3M\sigma_u^2 = 3 \text{Tr}(R)$. Alors

$$\delta^2 \leq \mu \frac{J_{\min} \text{Tr}(R)}{2 - \mu MB} \quad \text{pour } \mu < \frac{2}{MB} = \frac{2}{3 \text{Tr}(R)}$$

– Cas général : on majore le second terme de (2.16) par l'inégalité de Schwarz, on obtient les mêmes expressions que ci-dessus en prenant $M^2 B^2 = E(\|U_n\|^4) \cdot m_4 / \sigma^4$ (moments de la loi asymptotique stationnaire de f_n)

– Cas du petit pas : $\mu \rightarrow 0$, on obtient par D.L.

$$\delta^2 = \frac{\mu}{2} J_{\min} \text{Tr}(R) + o(\mu) \quad (2.17)$$

Pour la mise en œuvre de l'algorithme, on devra gérer un compromis pour le choix de μ : une adaptation rapide (transitoire court) exige μ le plus grand possible (compatible avec la stabilité), alors que la précision liée à la réduction de l'EQMR nécessite $\mu \rightarrow 0$.

Remarque : la borne supérieure μ_0 admissible pour la stabilité en m.q. de la fluctuation n'est établie de façon exacte que dans le cas du signal borné, où elle a la forme $\mu_0 = \frac{2}{3 \text{Tr}(R)}$: elle est nettement plus contraignante que celle établie pour le (GD) (1.27). On trouve en général dans la littérature la borne $\mu_0 = \frac{2}{\text{Tr}(R)}$, qui ne peut s'établir que sous "l'hypothèse classique d'indépendance" (toujours fautive pour le filtrage

adaptatif d'ordre $M > 1$) [Widrow 76]. La vraie borne μ_0 dépend de façon complexe de la statistique des signaux, on garantira en pratique la stabilité m.q. par un pas $\mu < \frac{2}{\alpha \text{Tr}(R)}$ avec $\alpha \geq 3$.

Estimation déterministe asymptotique sur la fluctuation du LMS

Pour s'affranchir de l'hypothèse (H3) donnée en (2.13), qui est fautive dans beaucoup de cas du filtrage de Wiener, le résultat le plus général a été établi par [Solo 89] en se restreignant au cas du signal borné, et en adoptant une démarche d'estimation déterministe sur une trajectoire (plutôt que probabiliste moyenne) conforme au point de vue de l'utilisateur (qui ne dispose que d'une réalisation). De plus cette démarche de "moyennage déterministe" s'étend à des algorithmes très généraux (cf Chapitre 3).

On note $e_0(n) = b_n - U_n^T Z_n$ l'erreur d'estimation stationnaire du LMS. On obtient le théorème suivant, généralisant (2.17).

THÉORÈME.— [Solo 89]

On suppose les signaux (x_n, u_n) strictement stationnaires et vérifiant les hypothèses :

(S1) Signal borné $\|U_n\|^2 < B$ et $\mu < 2/B$

(S2) U_n est un processus "purement non déterministe" (puissance de l'innovation à passé infini $\sigma_\varepsilon^2 > 0$)

Alors l'erreur quadratique moyenne (déterministe) du LMS converge p.s. :

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_1^N e_k^2 = E(e_0(n)^2) = J_{\min} \left(1 + \frac{\mu}{2} m f \text{Tr}(R)\right) + O(\mu^2) \quad (2.18)$$

avec $M = E(b_n^2 \|U_n\|^2)$ et

$$f = \frac{M}{J_{\min} \text{Tr}(R)} \quad m = 1 - \frac{2}{M} \sum_1^\infty E(b_{k+1} b_0 U_{k+1}^T \Pi_{k,0} U_0)$$

On notera que sous l'hypothèse (H3), on retrouve $m = f = 1$ et donc l'évaluation (2.17).

Simulations et compromis vitesse-précision du LMS

Les simulations ci-dessous illustrent le compromis vitesse-précision et la borne de stabilité en m.q. du LMS, en évaluant l'EQMR selon μ , sur 50 réalisations. Le modèle simulé est l'AR2, de paramètres

$$a_1 = 0.5562, a_2 = -0.81, J_{\min} = 0.3114, \text{Tr}(R) = 2, nstat = 50$$

Les résultats sont donnés dans le tableau (2.1) et la figure 2.2. On y constate

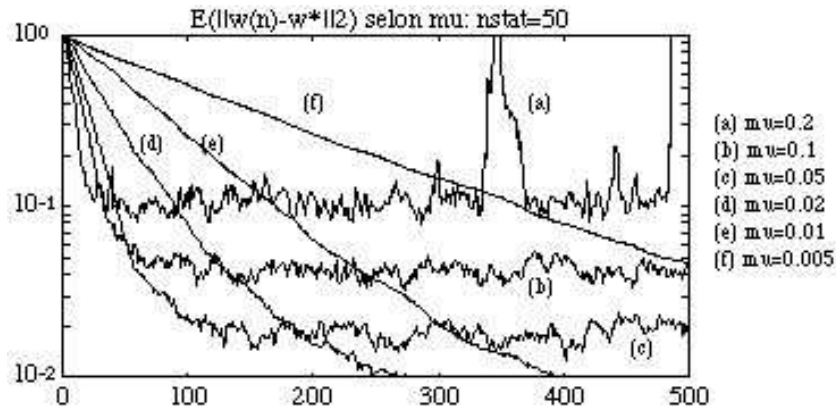


Figure 2.2. LMS : vitesse et EQMR selon μ

– divergence à partir de $\mu = 0.35$, correspondant à $\alpha = 3$

– puis amélioration de l'estimation de w^* jusqu'à $\mu = 0.02$, avec stabilisation de l'eqm sur W ($\lim \text{var}W$) à 6×10^{-3} et de l'eqm à 0.316 : l'erreur d'ajustement est alors de l'ordre de 2×10^{-2} correspondant à l'évaluation théorique du "petit pas" $\mathcal{M} = \frac{\mu}{2} M \sigma_u^2 = 2 \times 10^{-2}$ pour $\mu = 0.02$ ($M = 2, \sigma_u^2 = 1$),

– pour $\mu < 0.02$, $n = 500$ est inférieur au temps du transitoire et l'erreur augmente.

On remarquera aussi la robustesse du LMS et sa tendance à l'auto-stabilisation en cas de début d'instabilité liée à un pas μ trop grand : voir sur la figure le cas (a). Cependant, en implémentation virgule fixe, ces bouffées d'instabilité risquent de conduire à un débordement de capacité, et sont donc à éviter.

μ	$E(w_1)$	$E(w_2)$	$\lim \text{var}W$	EQM
0.4	5.1699	-1.2438	4.710^4	1.02710^5
0.35	1.7587	-0.7156	9.210^3	1.84710^4
0.30	0.4948	-0.6338	11	27.85
0.20	0.5284	-0.6926	0.88	2.983
0.10	0.5284	-0.6926	4.110^{-2}	3.45910^{-1}
0.05	0.5438	-0.7575	1.910^{-2}	3.3010^{-1}
0.02	0.5640	-0.7923	6.110^{-3}	3.16810^{-1}
0.01	0.5243	-0.7641	7.410^{-3}	3.16510^{-1}
0.005	0.4054	-0.6325	6.110^{-2}	3.46910^{-1}

Tableau 2.1.

En conclusion, le tableau ci-dessous résume les propriétés principales du LMS à pas fixe μ , utiles pour les choix d'implémentation de l'algorithme.

Stabilité en m.q.	$0 < \mu < \frac{2}{\alpha \text{Tr}(R)} \quad \alpha \geq 3$
Stabilité exponentielle du transitoire	Taux $\beta(\mu) = 1 - \mu \lambda_{\min}(R) < 1$
Erreur quadratique stationnaire	$\lim E(\ \Lambda_n\ ^2) \leq \mu J_{\min} C$
EQMR	$\delta^2 = \mu \frac{J_{\min} \text{Tr}(R)}{2 - \mu \alpha \text{Tr}(R)}$
Cas du petit pas : $\mu \rightarrow 0$	$\delta^2 = \frac{\mu}{2} J_{\min} \text{Tr}(R) + o(\mu)$

Tableau 2.2. Propriétés du LMS à pas fixe μ

2.1.3. Variantes du LMS : NLMS, formes par blocs

L'algorithme du gradient stochastique normalisé et le NLMS

Le tableau 2.2 rappelle que la stabilité en MQ de l'algorithme LMS est dépendante de la trace de la matrice de covariance R du signal d'excitation u_n . Cette propriété a une incidence pratique importante car le filtrage adaptatif a souvent à traiter des signaux d'excitation de puissance non constante, comme l'est par exemple le signal de parole. Il en résulte qu'assurer la stabilité en MQ de l'algorithme pour ce type de signaux nécessite le réglage permanent du pas μ afin que celui-ci soit maintenu dans le domaine adéquat.

Un moyen simple d'obtenir ce résultat est de normaliser le pas par la variance du signal d'excitation, supposée connue *a priori* ou estimée sur les échantillons de ce signal. On définit ainsi l'*algorithme du gradient stochastique normalisé* par l'équation d'adaptation suivante :

$$w_n = w_{n-1} + \frac{\mu}{\beta \text{var}(u_n)} e_n U(n) \quad (2.19)$$

où $\text{var}(u_n)$ représente la variance de l'excitation. β est un coefficient constant ; on notera que d'après l'analyse précédente du LMS, le choix $\beta \geq \frac{3M}{2}$ garantit la stabilité en MQ de l'algorithme au moins pour une excitation de distribution uniforme.

Dans la pratique $\text{var}(u_n)$ est une estimation empirique de la variance. Deux estimateurs, désignés par la fenêtre de pondération des données, sont traditionnellement employés :

L'estimateur à fenêtre rectangulaire

$$\text{var}(u_n) = \frac{1}{N} \sum_{i=1}^N u_{n-i+1}^2 \quad (2.20)$$

L'estimateur à fenêtre exponentielle

$$\text{var}(u_n) = (1 - \lambda) \sum_{i=1}^N \lambda^{n-i} u_i^2 \quad (2.21)$$

Sous l'hypothèse que l'excitation est centrée, ces quantités sont des estimations non biaisées (asymptotiquement non biaisée pour (2.21)) de la variance théorique $E(u_n^2)$. On notera que le choix approprié de la taille de fenêtre N ou du facteur d'oubli λ donne à ces estimateurs une "mémoire" ajustable aux caractéristiques du signal d'excitation lorsque celui-ci est non stationnaire. On notera aussi qu'il est de pratique commune d'ajouter au terme $\text{var}(u_n)$ une petite constante positive de "régularisation" qui aide à contrôler le comportement de l'algorithme lors des périodes de très faible énergie d'excitation.

Faisant l'hypothèse implicite d'une excitation non corrélée, l'algorithme (2.19) peut être considéré comme une version "dégénérée" de l'algorithme des moindres carrés récursifs (RLS) qui fera l'objet du paragraphe 2.2 : en effet, il est élémentaire de vérifier qu'en négligeant les termes non diagonaux de la matrice de covariance déterministe du RLS, proche d'une matrice diagonale sous cette hypothèse, on obtient exactement l'algorithme (2.19) en appliquant l'estimateur approprié de la variance $\text{var}(u_n)$.

L'algorithme (2.19) muni de l'estimateur de variance à fenêtre rectangulaire avec $N = M$ est un "algorithme de projection" d'ordre 1, comme ceci est discuté au paragraphe 2.3. Historiquement, c'est cet algorithme qui a été dénommé NLMS (pour Normalized Least Mean Squares). On observera que le choix $\beta = M$ et $\mu = 1$, cf. (2.61) conduit à l'annulation de l'erreur *a posteriori*.

Comportement de l'algorithme du gradient stochastique normalisé

L'analyse théorique du comportement de l'algorithme du gradient stochastique normalisé est plus délicate que celle du LMS en raison de la présence du terme de normalisation du pas, qui dépend de l'excitation. On rappelle ici les résultats d'une analyse classique du comportement du NLMS [Slock 93].

Les vecteurs d'excitation $\{U_n\}$ sont supposés centrés, identiquement distribués et indépendants (hypothèse classique d'indépendance). Pour modéliser une large classe

de signaux d'excitation du monde réel, considérons la décomposition de la matrice du signal d'excitation suivant : $R = E [U_n U_n^T] = S \Sigma S^T$ avec $\Sigma = \text{diag}(\lambda_1, \dots, \lambda_M)$ et $S = (\nu_1, \dots, \nu_M)$, les paramètres $\{\lambda_i\}$ et $\{\nu_i\}$, $i = 1, \dots, M$ représentant respectivement les valeurs propres et les vecteurs propres orthonormés de la matrice R .

Suivant le modèle (dénommé MI), le vecteur U_n est supposé formé du produit de trois variables aléatoires indépendantes : $U_n = s r V$ où

$$\begin{aligned} Pr(s = \pm 1) &= 1/2 \\ Pr(V = \nu_i) &= \lambda_i / \text{Tr}(R), i = 1, \dots, M \\ r &\approx \|U_n\| \end{aligned} \quad (2.22)$$

Avec l'hypothèse suivante sur l'observation y_n (hypothèse classique d'un schéma d'identification) : $y_n = w_n^* U_n + \epsilon^*$, $\{\epsilon^*\}$ étant une suite d'échantillons i.i.d. centrés et de variance σ_b^2 , on obtient les résultats suivants :

- convergence en M.Q. ssi $0 < \mu < 2$
- désajustement $\frac{J_\infty - \sigma_b^2}{\sigma_b^2} = \frac{\mu}{2-\mu} \text{Tr}\{R\} E \left[\frac{1}{r^2} \right]$.

La courbe de convergence de l'erreur quadratique $J(n)$ est donnée par $J(n) = \sigma_b^2 + \sum_{i=1}^M \lambda_i \tilde{\lambda}_{n,i}$ avec :

$$\tilde{\lambda}_{n,i} = [1 - \mu(2 - \mu) p_i] \tilde{\lambda}_{n-1,i} + \mu^2 \sigma_b^2 E \left(\frac{1}{r^2} \right) p_i \quad (2.23)$$

où $p_i = \lambda_i / \text{Tr}(R)$. Ce résultat est un cas particulier (pour $P = 1$) de celui établi avec l'algorithme APA (2.61) qui sera présenté dans la partie 2.3.4. La convergence la plus rapide est obtenue pour $\mu = 1$. On notera qu'alors la convergence du NLMS peut être plus rapide que celle du LMS avec $\mu = \frac{1}{\text{Tr}(R)}$; ceci a été montré par exemple sur des signaux d'excitation gaussiens i.i.d. [Tarrab 88].

La dépendance directe du terme de normalisation du pas $U_n^T U_n$ par rapport aux échantillons du signal d'excitation introduit un facteur de fluctuation supplémentaire par rapport à l'algorithme LMS à pas normalisé par $\text{Tr}(R)$, ce qui conduit à un désajustement plus élevé du NLMS et à un risque de déstabilisation locale. Ce dernier peut être contrôlé par l'emploi d'une constante de régularisation, comme il a été dit plus haut (ce point sera étudié plus en détails dans la partie sur l'algorithme APA).

Compatibilité de la fenêtre d'estimation avec la taille M du filtre

Dans l'algorithme NLMS, qui utilise la forme (2.20) de l'estimateur de variance avec $N = M$, la taille de la fenêtre sur les données d'excitation est égale à la longueur M du filtre. Dans le cas de l'algorithme du gradient stochastique normalisé par l'estimateur (2.21), la fenêtre sur les données est asymptotiquement de taille infinie, cependant sa "taille efficace" dépend de la valeur du facteur d'oubli λ .

Pour établir une forme d'équivalence statistique entre les algorithmes normalisés par les estimateurs (2.20) et (2.21), raisonnons en terme de variance de chaque estimateur normalisée par sa moyenne élevée au carré. Sous l'hypothèse d'excitation centrée i.i.d. stationnaire d'ordre 4 de variance $E(u_n^2) = \sigma^2$ et de moment non centré d'ordre quatre m_4 , on obtient ainsi pour l'estimateur (2.20) :

$$\frac{E \left[(\text{var}(u_n) - E[\text{var}(u_n)])^2 \right]}{E[\text{var}(u_n)]^2} = \frac{1}{M} \left(\frac{m_4}{\sigma^4} - 1 \right) \quad (2.24)$$

et asymptotiquement pour l'estimateur (2.21)

$$\lim_{n \rightarrow +\infty} \frac{E \left[(\text{var}(u_n) - E[\text{var}(u_n)])^2 \right]}{E[\text{var}(u_n)]^2} = \frac{1 - \lambda}{1 + \lambda} \left(\frac{m_4}{\sigma^4} - 1 \right) \quad (2.25)$$

Le choix $\lambda = \frac{M-1}{M+1}$ conduit à une valeur de l'expression (2.25) égale à celle de (2.24), c'est-à-dire que les deux estimateurs ont (asymptotiquement) la même variance. On peut alors parler de fenêtre exponentielle compatible avec la taille M du filtre, et prévoir un comportement asymptotique de l'algorithme du gradient stochastique normalisé avec oubli exponentiel voisin de celui du NLMS. Quant au régime transitoire, il dépend très fortement des données locales qui conditionnent le comportement initial des estimateurs (2.20) et (2.21). Une comparaison plus facile entre les deux formes d'algorithmes normalisés se fera en supposant que la fenêtre rectangulaire ou exponentielle est "remplie" préalablement" par suffisamment de données. Dans le cas (2.21) une règle empirique sera par exemple d'attendre que le nombre d'échantillons d'excitation dépasse $\frac{3}{1-\lambda}$ avant de démarrer l'adaptation suivant (2.19).

D'après (2.24) et (2.25), on observera que pour $M \gg 1$, un choix de fenêtre exponentielle compatible et un signal d'excitation stationnaire, les estimateurs (2.20) et (2.21) ont une faible variance et peuvent être considérés comme constants : le comportement asymptotique de l'algorithme du gradient stochastique normalisé est alors quasiment le même que celui du LMS avec le pas constant $\mu/\text{Tr}(R)$.

Exploitation de la redondance des calculs : forme bloc-exacte du LMS ; autres formes "par blocs"

Si la simplicité calculatoire du LMS en terme de nombre d'opérations arithmétiques par échantillon est déterminante dans des applications où le nombre M de coefficients du filtre est peu élevé, ce nombre d'opérations peut être rédhibitoire dans le cas de très longs filtres rencontrés par exemple en annulation d'écho acoustique (voir chapitre sur ce sujet dans la seconde partie de l'ouvrage). Cette contrainte de complexité peut être contournée par l'exploitation de la redondance algorithmique intrinsèque au LMS, dans lequel les vecteurs de données U_n et U_{n-1} possèdent $M - 1$ composantes communes. En corollaire, la redondance algorithmique peut être réduite par une transformation du LMS pour que le filtre soit maintenu constant sur un bloc de taille N , ce qui lorsque $N \gg 1$ permet notamment le calcul efficace de N convolutions successives par l'emploi d'une transformée orthogonale telle que la DFT.

Il existe une forme bloc-exacte du LMS [Benesty 92] que nous rappelons brièvement ici. Soit N la taille du bloc. On peut écrire :

$$w_n = w_{n-N} + \mu [U_n U_{n-1} \cdots U_{n-N+1}] E_n = w_{n-N} + \mu V^T(n) E_n \quad (2.26)$$

avec $E_n^T = [e_n e_{n-1} \cdots e_{n-N+1}]$ le vecteur des N dernières erreurs successives du LMS. Appelons $Y_n' = V(n) w_{n-N}$ le vecteur des N dernières convolutions successives des vecteurs d'entrée avec le filtre constant sur le bloc w_{n-N} et $Y_n = [y_n y_{n-1} \cdots y_{n-N+1}]^T$ le vecteur des N dernières observations. Nous posons :

$$E_n = G_n (Y_n - Y_n') = G_n \varepsilon_n \quad (2.27)$$

G_n est la matrice de correction permettant de passer des erreurs calculables avec le filtre constant w_{n-N} aux erreurs exactes du LMS ; on montre qu'elle s'explique comme suit :

$$G_n = [\mu S_n + I_n]^{-1} \quad (2.28)$$

où I_n est la matrice identité $N \times N$ et S_n s'écrit :

$$S_n = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 \\ U_{n-N+2}^T U_{n-N+1} & 0 & \cdots & \cdots & 0 \\ U_{n-N+3}^T U_{n-N+1} & U_{n-N+3}^T U_{n-N+2} & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ U_n^T U_{n-N+1} & U_n^T U_{n-N+2} & \cdots & U_n^T U_{n-1} & 0 \end{bmatrix} \quad (2.29)$$

Compte tenu de la forme triangulaire de la matrice S_n , l'équation (2.27) se résout par substitutions ; les termes de cette matrice peuvent se déduire les uns des autres par récursivité. On obtient ainsi une forme bloc-exacte du LMS puisque la suite des erreurs produites est exactement celle de l'algorithme initial. La contrainte évidente est que le pas μ doit rester constant sur le bloc. Les conditions de stabilité établies précédemment pour le LMS s'appliquent ici puisque l'algorithme bloc-exact lui est équivalent.

L'avantage de cette forme bloc du LMS est la réduction du nombre d'opérations arithmétiques : en effet, l'équation (2.26) peut être résolue avec un nombre de multiplications inférieur à $M \cdot N$ au moyen du réordonnement des éléments de $V(n)$ en composantes polyphases ; cependant, le gain le plus significatif est obtenu par l'emploi de techniques de convolution rapide par transformée orthogonale. Le calcul des éléments de la matrice G_n et la résolution de l'équation (2.27) coûtent quant à eux $N(N-1)$ multiplications et $2.5 N(N-1)$ additions. On obtient ainsi pour $M = 1024$ et $N = 128$, les coûts suivants en nombre d'opérations par échantillon, la forme bloc-exacte utilisant une transformée de Fourier rapide "split radix" :

- LMS "standard" : 2048 additions et 2048 multiplications
- Forme bloc-exacte : 678 additions et 349 multiplications.

Cette formulation par blocs du LMS permet d'introduire un ensemble de variantes d'algorithmes adaptatifs par blocs : par exemple, en posant $N = M$ et $S_n = 0$ on obtient le "Block-LMS" traditionnel [Clark 81]. Il est à noter que la vitesse de convergence de ce dernier est inférieure à celle du LMS (et donc du bloc-exact LMS), sauf dans le cas d'une excitation blanche : en effet, si $M \gg 1$, les termes de la matrice S_n , assimilables à des corrélations, multipliés par μ , sont alors faibles devant l'unité et par conséquent le bloc-exact LMS et le "Block-LMS" se rejoignent. Dans le cas $M = KN$, avec K entier, il est à noter une parenté du bloc-exact LMS intégrant la FFT avec le "Multi-delay Adaptive Filter" [Soo90] qui sera évoqué dans le chapitre de la seconde partie de l'ouvrage consacré à l'annulation d'écho.

Enfin, le passage des équations du LMS par bloc dans le domaine transformé (typiquement Fourier) permet de moduler le pas d'adaptation appliqué à chaque coefficient dans l'équation équivalente à (2.26) de telle sorte que la vitesse de convergence soit

considérablement augmentée. Cette possibilité sera aussi explorée dans le même chapitre de la seconde partie de l'ouvrage.

2.2. Moindres carrés récursifs : RLS

Introduction

La méthode des Moindres carrés s'applique à des situations très générales, où on cherche à approcher un vecteur de données par un modèle linéaire en un vecteur de paramètres θ . Cette méthode est d'usage courant en Analyse Numérique et en Statistique (approximation de fonction, lissage de courbes, régression multilinéaire). Ses très bonnes propriétés statistiques pour éliminer l'influence de "bruits" perturbant des données "utiles" en font un très bon candidat pour le traitement optimal des signaux bruités.

Ici nous l'utilisons comme méthode systématique de construction d'un algorithme adaptatif : on remplace le critère statistique d'eqm $J(\theta)$ par un critère géométrique de moindres carrés, que l'on minimise par un algorithme déterministe. Il s'agit au départ d'une méthode bloc de traitement des données, qui a été étudiée au Chapitre 1 (cf section 1.3.2). Nous allons lui donner une forme récursive exacte, d'où le nom de l'algorithme : Moindres carrés récursifs, RLS (*Recursive Least Squares*). On étudie ensuite les propriétés de convergence de cet algorithme adaptatif.

2.2.1. L'algorithme récursif : RLS

Le passage de la forme "bloc" des équations (1.21) à une forme récursive découle directement de la récurrence sur les matrices $\Gamma_n = nR_n = \Phi^T \Phi$ et vecteurs $c_n = nr_n = \Phi^T X$ définis en (1.19) :

$$\Gamma_n = \Gamma_{n-1} + \varphi_n \varphi_n^T \quad c_n = c_{n-1} + x_n \varphi_n \quad (2.30)$$

On part de l'hypothèse de récurrence $\Gamma_{n-1} \theta_{n-1} = c_{n-1}$ et on veut résoudre, avec $\theta_n = \theta_{n-1} + \delta$:

$$(\Gamma_{n-1} + \varphi_n \varphi_n^T)(\theta_{n-1} + \delta) = c_{n-1} + x_n \varphi_n$$

d'où après simplification $\Gamma_n \delta = e_n \varphi_n$ et l'algorithme récursif

RLS

$$\begin{aligned} \theta_n &= \theta_{n-1} + K_n e_n \varphi_n \\ e_n &= x_n - \varphi_n^T \theta_{n-1} && \text{erreur de prédiction "a priori"} \\ K_n &= \Gamma_n^{-1} && \text{gain matriciel} \end{aligned}$$

Le calcul récursif direct de K_n à l'aide de K_{n-1} est fait par le *Lemme d'inversion matricielle* : pour A matrice symétrique, b un vecteur, on a

$$(A + bb^T)^{-1} = A^{-1} - sA^{-1}bb^T A^{-1} \quad \text{avec } s = 1/(1 + b^T A^{-1}b) \quad (2.31)$$

qu'on applique ici avec $A = \Gamma_{n-1}$ et $b = \varphi_n$. On obtient alors la forme complète d'implémentation de l'algorithme.

RLS

$$\theta_n = \theta_{n-1} + K_n e_n \varphi_n \quad (2.32)$$

avec

$$\begin{aligned} e_n &= x_n - \varphi_n^T \theta_{n-1} \\ v_n &= K_{n-1} \varphi_n \quad s = 1/(1 + \varphi_n^T v_n) \\ K_n &= K_{n-1} - s(v_n v_n^T) \end{aligned}$$

COMPLEXITÉ DU RLS.– Les opérations de calcul de v_n, K_n, θ_n nécessitent chacune M^2 multiplications, et il faut mémoriser des vecteurs de taille M et une matrice de taille M^2 . On a donc une complexité temporelle et spatiale en $O(M^2)$, à comparer défavorablement à la complexité linéaire en $O(M)$ du LMS.

INITIALISATION DU RLS.– En principe la matrice R_n de corrélation empirique ne pourra être inversible que pour $n > M$. Il faudrait donc calculer au départ R_M et le paramètre θ_M associé, comme point de départ de la récurrence du RLS. En pratique, on évitera cette phase préalable en initialisant arbitrairement la suite K_n par $K_0 = \frac{1}{\delta}I$, avec $\delta > 0$. Ceci revient à remplacer la suite R_n par la suite "régularisée"

$$R_n = \frac{1}{n} \left(\sum_1^n \varphi_n \varphi_n^T + \delta I \right)$$

et on voit que pour $n \rightarrow \infty$ l'influence de la perturbation initiale sera négligeable. Cette méthode présente l'intérêt de construire par le RLS une suite croissante de matrices $\Gamma_{n+1} \geq \Gamma_n \geq \delta I > 0$: on est ainsi sûr de l'inversibilité de toutes les matrices R_n . On constate en pratique que le choix de δ n'a aucune influence sur la vitesse de convergence de l'algorithme.

Interprétation : Quasi-Newton stochastique

Nous avons montré au Chapitre 1, section 1.4.2, que l'algorithme RLS peut être obtenu en appliquant la méthode Newton sur l'écart des Moindres carrés $J_n(\theta)$: voir (1.33). On l'a aussi interprété comme algorithme de quasi-Newton stochastique. Compte tenu des résultats de vitesse de convergence (quadratique) vus au Chapitre 1

sur les méthodes newtoniennes, cette interprétation expliquera la supériorité du RLS sur le LMS du point de vue vitesse de convergence. Nous le réécrivons sous la forme

$$\begin{aligned} \text{RLS1} \quad \theta_n &= \theta_{n-1} + \mu_n R_n^{-1} e_n \varphi_n & \mu_n &= \frac{1}{n} \\ R_n &= R_{n-1} + \mu_n (\varphi_n \varphi_n^T - R_{n-1}) \end{aligned}$$

Par rapport au critère d'erreur quadratique moyenne $J(\theta)$, on a

$$\frac{1}{2} \nabla J(\theta) = -E(e_n \varphi_n), \quad \frac{1}{2} \nabla^2 J(\theta) = R$$

et R_{n-1} est une suite convergeant vers R^{-1} , $e_n \varphi_n$ est l'approximation stochastique du gradient.

L'algorithme RLS fonctionne uniquement en contexte **stationnaire**, il correspond à un algorithme quasi-Newton à pas décroissant $\mu_n = 1/n$. Il converge strictement vers le paramètre optimal θ^* : cela signifie qu'il n'évoluera plus, une fois terminé sa phase de convergence, donc il n'a pas de *capacité d'évolution permanente* pour suivre les variations du contexte. Nous allons maintenant le modifier légèrement pour obtenir cette adaptativité permanente.

2.2.2. Algorithme adaptatif : facteur d'oubli

Il faut passer d'un algorithme à *pas décroissant* à un algorithme à *pas constant* μ . Dans le contexte des Moindres carrés, il se trouve que l'algorithme à pas constant va encore s'interpréter comme la résolution d'une minimisation exacte, mais pour un nouveau critère *de moindres carrés pondérés* : on se fixe un *facteur d'oubli* $\lambda \in]0, 1]$, et on prend le critère

$$J_n(\theta) = \mu \sum_{k=1}^n \lambda^{n-k} e_n^2 \quad \text{avec } \mu = 1 - \lambda \quad (2.33)$$

INTERPRÉTATION.— On oublie les données anciennes à la vitesse exponentielle λ^{n-k} . Remarquons que cette pondération des données revient aussi à travailler dans le même espace géométrique $H = \mathbb{R}^n$, mais avec le nouveau produit scalaire pondéré

$$\langle X, Y \rangle = \mu \sum_{k=1}^n \lambda^{n-k} x_k y_k \quad (2.34)$$

et la matrice de Gram R_n s'écrit alors

$$R_n = \Phi^T \Lambda \Phi = \mu \sum_{k=1}^n \lambda^{n-k} \varphi_k \varphi_k^T \quad \text{avec } \Lambda = \mu \text{Diag}(\lambda^{n-k}) \quad (2.35)$$

tandis que le vecteur projeté devient $r_n = \Phi^T \Lambda X = \mu \sum_1^n \lambda^{n-k} x_k \varphi_k$.

On obtient les equations normales pondérées

$$R_n \theta_n = r_n \quad (2.36)$$

Les relations de récurrence deviennent

$$R_n = \lambda R_{n-1} + \mu \varphi_n \varphi_n^T \quad r_n = \lambda r_{n-1} + \mu x_n \varphi_n \quad (2.37)$$

et on établit exactement comme précédemment l'algorithme récursif RLS- λ : notons que l'algorithme sans facteur d'oubli s'obtient en posant $\lambda = 1$, et en substituant à R_n la matrice $\Gamma_n = n R_n$ définie en (2.30).

On pourra donc mettre en oeuvre de façon unique les deux versions du RLS, le choix se faisant par la valeur de λ . D'autre part, le paramètre $\mu = 1 - \lambda$ jouera ici le même rôle que le pas μ de l'algorithme LMS de gradient stochastique.

$$\begin{aligned} \text{RLS-}\lambda \quad \theta_n &= \theta_{n-1} + K_n e_n \varphi_n \\ e_n &= x_n - \varphi_n^T \theta_{n-1} \\ v_n &= K_{n-1} \varphi_n \quad s = 1/(\lambda + \varphi_n^T v_n) \\ K_n &= \frac{1}{\lambda} (K_{n-1} - s(v_n v_n^T)) \end{aligned}$$

On peut interpréter cet algorithme comme une méthode quasi-Newton stochastique à *pas constant* : cette interprétation est visible sur la réécriture suivante de l'algorithme :

$$\begin{aligned} \text{RLS2} \quad \theta_n &= \theta_{n-1} + \mu R_{n-1}^{-1} e_n \varphi_n \quad \mu = 1 - \lambda \\ R_n &= R_{n-1} + \mu (\varphi_n \varphi_n^T - R_{n-1}) \end{aligned}$$

2.2.3. Convergence et propriétés statistiques

La méthode des moindres carrés permet de construire une suite $(\theta_n)_{n \in \mathbb{N}}$ d'estimateurs de θ^* , basée sur les signaux observés (x_1, \dots, x_n) et (u_1, \dots, u_n) . L'étude de la convergence de ces estimateurs est basée sur le modèle statistique des données traitées, donné en (1.14), on distinguera les deux cas suivants :

$$\mathbf{M1} \quad \text{Modèle de régression linéaire } b_n \text{ indépendant de } (u_k)_{k \in \mathbb{Z}} \quad (2.38)$$

et

$$\mathbf{M2} \quad \text{Modèle de Wiener } b_n \text{ décorrélé de } \varphi_n \quad (2.39)$$

Dans le premier cas, on peut raisonner conditionnellement au signal (u_n) , sur le seul bruit de modèle (b_n) . Ce n'est plus possible dans le cas (M2) général de Wiener.

Convergence p.s. du RLS

On se place en contexte stationnaire, avec l'algorithme à pas décroissant, c'est-à-dire le RLS1 (cf p.62). Les hypothèses générales du modèle sont (M2) (2.39).

On peut représenter vectoriellement les équations du modèle et de l'estimateur à l'aide des vecteurs $X = (x_1, \dots, x_n)$ et $B = (b_1, \dots, b_n)$.

$$X = \Phi\theta^* + B \quad (2.40)$$

et l'équation (1.21) se réécrit alors

$$\Phi^T \Phi (\theta_n - \theta^*) = \Phi^T B \quad (2.41)$$

d'où

$$\theta_n = \theta^* + R_n^{-1} d_n \quad \text{avec } d_n = \frac{1}{n} \Phi^T B = \frac{1}{n} \sum_{k=1}^n b_k \varphi_k \quad (2.42)$$

Comme pour le LMS, ce sont les propriétés d'ergodisme des signaux (x_n, φ_n) qui permettront la convergence du RLS : on les suppose stationnaires et mutuellement ergodiques du second ordre au sens fort, avec les convergences presque sûres des matrices empiriques d'auto et d'intercorrélations

$$R_n \rightarrow R \quad \text{p.s. si } n \rightarrow \infty, \quad R > 0 \quad r_n \rightarrow p \quad \text{p.s.} \quad (2.43)$$

Alors R^{-1} existe et $R_n^{-1} \rightarrow R^{-1}$, donc $\|R_n^{-1}\| \leq M$. Par l'ergodisme, on a $d_n \rightarrow E(b_n \varphi_n) = 0$, d'où par (2.42) le théorème suivant.

THÉORÈME.— Convergence p.s. du RLS

Si le processus conjoint $\{u_n, b_n\}$ est strictement stationnaire et ergodique au second ordre avec une matrice de covariance $R = E(U_n U_n^T) > 0$, on a pour la suite (θ_n) du RLS (2.32)

$$\theta_n \rightarrow \theta^* \quad \text{avec probabilité 1} \quad (2.44)$$

Contrairement au cas du LMS, cette convergence est très simple à établir ici. Elle se complète aussi facilement par des résultats statistiques importants caractérisant les estimateurs de Moindres carrés.

Biais et variance : cas de la régression linéaire

Ayant une expression explicite de θ_n pour tout n , on peut aussi s'intéresser aux propriétés de cet estimateur à n fixé. On traite ici le cas simple de la *régression linéaire* au sens défini par le modèle (M1) (2.38). Les vecteurs φ_k sont connus, non

aléatoires, et b_n est une suite centrée de covariance Γ_b : on raisonne conditionnellement au processus u_n , le seul aspect aléatoire étant contenu dans le bruit (b_n) du modèle.

Il en résulte alors que les matrices Φ sont non aléatoires. Cette situation décrit par exemple le cas de l'annulation d'écho. On déduit alors de (2.42) que $E(\theta_n) = \theta^*$ (estimateur non-biaisé), et $\text{cov}(\theta_n) = \Gamma_n^{-1} \Phi^T \Gamma_b \Phi \Gamma_n^{-1}$

CAS DU BRUIT BLANC .-

$$\Gamma_b = \sigma^2 I \Rightarrow \text{cov}(\theta_n) = \frac{\sigma^2}{n} R_n^{-1} \quad (2.45)$$

Cette dernière propriété est importante d'un point de vue pratique : $\text{cov}(\theta_n) \rightarrow 0$, donc l'estimateur des Moindres carrés est convergent en m.q., et la vitesse de convergence est donnée par (2.45). Comme la matrice $R_n^{-1} = K_n$ est une donnée disponible à chaque itération du RLS, on a donc aussi la covariance de θ_n et par conséquent la précision atteinte par l'algorithme à chaque instant, en estimant σ^2 par son estimateur empirique non biaisé $\|E\|^2/(n - M)$ (où $E = (e_1, \dots, e_n)^T$ est l'erreur d'estimation vectorielle).

La supériorité sur le LMS est une vitesse de convergence beaucoup plus grande et indépendante du conditionnement de l'autocorrélation. Nous illustrerons ici cette vitesse par des simulations. Pour une mesure précise et comparative des vitesses des algorithmes stochastiques, nous renvoyons à l'approche générale du chapitre 3.

Comportement asymptotique du RLS- λ

L'introduction du facteur d'oubli $\lambda < 1$ fait perdre la convergence de l'estimateur θ_n . Ceci est dû au fait que les matrices R_n et r_n construites par (2.35) n'ont plus de propriété de convergence p.s. et restent aléatoires quand $n \rightarrow \infty$.

Comme pour le LMS, on aura deux phases de convergence : le transitoire, durée d'oubli de l'influence de la condition initiale loin de θ^* , puis une phase de fluctuation stationnaire liée au pas $\mu = 1 - \lambda$.

Vitesse transitoire et constante de temps

Nous déterminerons cette vitesse par celle de l'*algorithme déterministe* (2.47), version moyennée de RLS2 (p.63) : on y remplace $e_n \varphi_n$ et $\varphi_n \varphi_n^T$ par leur espérance mathématique (cf Chapitre 3 pour le lien entre les trajectoires des deux algorithmes lorsque $\mu \rightarrow 0$), ce qui donne les deux équations

$$DRLS \quad \theta_n = \theta_{n-1} + \mu R_n^{-1} R (\theta_{n-1} - \theta^*) \quad (2.46)$$

$$R_n = R_{n-1} + \mu (R - R_{n-1}) \quad (2.47)$$

L'étude de convergence, comme pour le gradient déterministe (GD), se fait en introduisant les écarts $V_n = \theta_n - \theta^*$ et $\Delta_n = R_n - R$, pour lesquels la récurrence devient

$$V_n = (I - \mu R_n^{-1} R) V_{n-1} \quad \Delta_n = \lambda \Delta_{n-1} \quad (2.48)$$

On initialise à $\Delta_0 = \delta I - R$, on a donc $\Delta_n = \lambda^n \Delta_0$, et le taux de compression est

$$\beta(\mu, n) = \|I - \mu R_n^{-1} R\| = \|I - \mu(I + \lambda^n R^{-1} \Delta_0)^{-1}\| = 1 - \frac{\mu}{1 + \lambda^n (\delta/\lambda_{\min} - 1)}$$

donc les deux erreurs convergent exponentiellement vers 0 au taux $\lambda = 1 - \mu$. On en déduit la constante de temps de ce transitoire RLS

$$\tau = \frac{-1}{\log \lambda} \approx \frac{1}{1 - \lambda} \quad \text{lorsque } \lambda \rightarrow 1 \quad (2.49)$$

Ce temps traduit la *largeur de fenêtre équivalente* pour la mémoire du RLS- λ sur les données, qui augmente infiniment lorsque $\lambda \rightarrow 1$. Par contre, elle ne traduit pas correctement la durée effective du transitoire du RLS lorsque la mémoire est longue : on s'intéresse en effet ici au début de la convergence, loin de θ^* , dans la phase de rapprochement, alors on a

$$\delta/\lambda_{\min} \ll 1, \lambda \approx 1 \Rightarrow \beta(\mu, n) \approx 1 - \frac{\mu}{1 - \lambda^n} \rightarrow 0 \quad \text{si } n \rightarrow 1 \quad (2.50)$$

ce qui donne une convergence *superlinéaire* pour les petites valeurs de n .

La comparaison de ce résultat avec le cas du LMS est intéressante : on a ici un taux exponentiel uniforme, indépendant de la direction et du conditionnement de R . On notera aussi que la stabilité de ce DRLS est assurée pour tout $\mu \in]0, 1]$. On aura évidemment des contraintes plus fortes pour l'algorithme stochastique, mais la convergence rapide donnée par (2.50) lorsque $\lambda \rightarrow 1$ évitera l'allongement des durées transitoires au "petit pas", contrairement au LMS (cf simulations ci-dessous).

Fluctuation résiduelle du RLS

La convergence vers une loi asymptotique stationnaire de la suite θ_n résulte de la convergence correspondante pour les matrices et vecteurs R_n et r_n (cf. [Eweda 87], où cette convergence est utilisée pour obtenir une borne de $\|\theta_n - \theta^*\|$ avec probabilité $> 1 - \varepsilon$). On les interprète comme des *filtrés linéaires* des suites aléatoires (supposées stationnaires) $\varphi_n \varphi_n^T$ et $x_n \varphi_n$ par le filtre $H_\lambda(z) = \frac{\mu}{1 - \lambda z^{-1}}$ et la stabilité de ce filtre pour $\lambda < 1$ entraîne la convergence vers une loi stationnaire, qui sera celle des matrices asymptotiques

$$R_n^* = \mu \sum_0^\infty \lambda^k \varphi_{n-k} \varphi_{n-k}^T \quad r_n^* = \mu \sum_0^\infty \lambda^k x_{n-k} \varphi_{n-k} \quad (2.51)$$

La convergence p.s. et en m.q. de ces séries est assurée dès que les matrices ou vecteurs filtrés sont stationnaires et possèdent une norme quadratique finie (ce qui suppose des moments d'ordre 4 finis pour (u_n)). La difficulté pour passer à l'erreur quadratique stationnaire $E(\|\theta_n - \theta^*\|^2)$ vient de la matrice inverse R_n^{-1} , dont il faut arriver à borner les moments. Le résultat principal a été obtenu par [Macchi-Eweda 88]. Il est basé sur une borne inférieure des valeurs propres des matrices $A_h(n) = \sum_{i=n+1}^{n+h} \varphi_i \varphi_i^T$, et s'énonce ainsi.

THÉORÈME. – Erreur quadratique du RLS- λ

Si le processus conjoint $\{u_n, x_n\}$ est strictement stationnaire avec une matrice de covariance $R = E(U_n U_n^T) > 0$, et vérifie les hypothèses suivantes :

(S1) Toutes les suites $Y_k = x_k u_{k-i}$ ou $Y_k = u_k u_{k-i}$ ont des moments finis à l'ordre 8 avec une corrélation à décroissance exponentielle

$$\exists C > 0, \rho < 1 : \quad |\gamma_Y(m)| \leq C \rho^m$$

(S2) Il existe un entier $h > 0$ tel que

$$\sup_{n>0} E[\lambda_{\min}^{-p}(A_h(n))] < \infty \quad \text{pour } p = 8 \quad (2.52)$$

alors il existe des constantes C_1 et μ_0 telles que

$$\forall \mu < \mu_0 \quad \limsup_{n \rightarrow \infty} E(\|\theta_n - \theta^*\|^2) \leq \frac{C_1}{\lambda_{\min}^2(R)} \frac{1 + \rho}{1 - \rho} \mu \quad (2.53)$$

Ce résultat est un équivalent pour RLS de celui du LMS (2.14). Il a une conséquence, importante du point de vue pratique et qui sera confirmée par les simulations : pour un niveau donné δ de fluctuations de θ_n , on devra choisir un pas μ d'autant plus faible que le taux de corrélation des données ρ sera proche de 1.

Il a été précisé par [Ljung 91], [Guo, Ljung, Priouret 93] : une condition suffisante pour (2.52) est donnée dans le cas où la suite φ_n est i.i.d. elle équivaut à l'existence de $K > 0, \gamma > 0, x_0 > 0$ tels que

$$\forall v \in \mathbb{R}^d, \|v\| = 1, 0 \leq x \leq x_0 \Rightarrow P(|v^T \varphi_1| \leq x) \leq K x^\gamma$$

et la propriété s'étend ensuite à des signaux générés à partir d'un tel bruit blanc (M-dépendants, ou modèle d'état observable). Par une méthode d'approximation générale (qui sera présentée au Chapitre 3), on obtient pour des signaux à dépendance faible à l'infini, le résultat asymptotique comparable à (2.17) obtenu pour LMS : en posant $\Sigma_b = \text{var}(b_n | u_n, u_k, b_k \ k < n)$ (qui vaut J_{\min} dans le cas (M1) de la régression linéaire),

$$E(\|\theta_n - \theta^*\|^2) \sim \frac{\mu}{2} \Sigma_b \text{Tr}(R^{-1}) \quad \text{lorsque } \mu \rightarrow 0 \quad (2.54)$$

Leurs résultats sont plus généraux et s'appliquent à la poursuite d'un filtre variable, qu'on abordera à la fin du Chapitre.

En conclusion, le comportement de fluctuation du RLS donne une erreur quadratique proportionnelle à μ , avec un compromis nécessaire entre vitesse transitoire et réduction de la fluctuation, et est comparable à celui du LMS.

Propriétés statistiques du RLS

Revenons aux Moindres carrés "purs" RLS, dont on a établi ci-dessus la consistance et les propriétés de biais et variance. On peut compléter dans le cas de la régression linéaire (M1)(2.38) par plusieurs résultats d'optimalité des Moindres carrés. On raisonne d'abord à signal (u_n) connu, non aléatoire, en partant des équations (??) et (2.42).

PROPRIÉTÉ DE VARIANCE MINIMALE.— On a vu que $\theta_n = (\Phi^T \Phi)^{-1} \Phi^T X$ est un estimateur convergent, non biaisé, de covariance $\Gamma(\theta_n) = \sigma^2 (\Phi^T \Phi)^{-1}$. Considérons alors un autre estimateur linéaire non biaisé T basé sur la même observation X : il existe une matrice A telle que $T = AX$ avec $E(T) = A\Phi\theta^* = \theta^*$. Ceci implique $A\Phi = I$. Alors

$$\Gamma(T, \theta_n) = E(AXX^T\Phi(\Phi^T\Phi)^{-1}) = \sigma^2 A\Phi(\Phi^T\Phi)^{-1} = \sigma^2(\Phi^T\Phi)^{-1} = \Gamma(\theta_n)$$

d'où l'orthogonalité $\Gamma(T - \theta_n, \theta_n) = 0$ et

$$\Gamma(T) = \Gamma((T - \theta_n) + \theta_n) = \Gamma(T - \theta_n) + \Gamma(\theta_n) \geq \Gamma(\theta_n)$$

ce qui prouve le théorème ci-dessous.

THÉORÈME.— Gauss-Markov

Dans le modèle de régression linéaire (M1), l'estimateur des Moindres carrés θ_n est à variance minimale parmi tous les estimateurs linéaires $T = AX$ non biaisés basés sur la même observation X . Cette variance minimale est donnée par

$$\Gamma(\theta_n) = \sigma^2 (\Phi^T \Phi)^{-1} = \frac{\sigma^2}{n} R_n^{-1} \quad (2.55)$$

CAS GAUSSIEN : MAXIMUM DE VRAISEMBLANCE ET EFFICACITÉ.— On suppose ici que (u_n, b_n) sont des processus gaussiens indépendants, de covariances R et $\sigma^2 I$. On note $f(x, u; \theta)$ la fonction de vraisemblance (ou densité de probabilité) des données observées $(x, u) = (x_1, \dots, x_n, u_1, \dots, u_n) \in \mathbb{R}^{2n}$ pour la valeur θ du paramètre. D'après (2.40), la loi conditionnelle de $(x|u)$ est gaussienne de moyenne $\Phi\theta$ et de covariance $\Gamma_b = \sigma^2 I$, on a donc $f(x, u; \theta) = f(u)f(x|u)$ et la Log-vraisemblance $\Lambda_n(\theta) = -\log f(x, u; \theta)$ vaut

$$\Lambda_n(\theta) = K + u^T R^{-1} u + \frac{1}{2\sigma^2} \|X - \Phi\theta\|^2 = K_1 + \frac{n}{2\sigma^2} J_n(\theta) \quad (2.56)$$

donc l'estimateur du *maximum de vraisemblance*, qui minimise $\Lambda_n(\theta)$, coïncide avec l'estimateur des *Moindres carrés* qui minimise $J_n(\theta)$.

L'information de Fisher en θ de l'observation (x, u) se déduit de (2.56)

$$I_n(\theta) = E\{\nabla^2 \Lambda_n(\theta)\} = \frac{n}{\sigma^2} R \quad (2.57)$$

la borne de Cramer-Rao pour la covariance d'un estimateur de θ est $I_n(\theta)^{-1}$. L'estimateur des Moindres carrés θ_n a d'après (2.45) la covariance complète (par rapport à la loi de probabilité de (u_n, b_n))

$$\Gamma(\theta_n) = \frac{\sigma^2}{n} E(R_n^{-1}) \quad (2.58)$$

et lorsque $n \rightarrow \infty$ on a par les propriétés d'ergodisme déjà vues

$$R_n \rightarrow R \quad \Rightarrow \quad E(R_n^{-1}) \rightarrow R^{-1} \Rightarrow \Gamma(\theta_n) \sim I_n(\theta)^{-1} \quad (2.59)$$

on peut donc affirmer que l'estimateur θ_n est *asymptotiquement efficace* : sa covariance converge vers la borne de Cramer-Rao $I_n(\theta)^{-1}$.

On peut enfin préciser ce résultat par un Théorème de limite-centrale : si $n \rightarrow \infty$, θ_n convergera en loi vers la loi gaussienne . Ce résultat se démontre par des calculs élémentaires.

THÉORÈME.– Normalité asymptotique

*Pour le modèle de régression linéaire (M1), si (u_n) et (b_n) sont gaussiens indépendants de covariances R et $\sigma^2 I$, l'estimateur des Moindres carrés θ_n est l'estimateur du **Maximum de vraisemblance**, et on a*

$$\sqrt{n}(\theta_n - \theta^*) \rightarrow \mathcal{N}(0, \sigma^2 R^{-1}) \quad (2.60)$$

(borne de Cramer-Rao)

REMARQUE.– Ces propriétés de maximum de vraisemblance, borne de Cramer-Rao et normalité asymptotique s'étendent à des modèles gaussiens plus généraux que (M1), notamment au cas de la modélisation AR(p). On se reportera à ce sujet à [Duflo 90] et [Ljung 79].

2.2.4. Simulations et comparaisons RLS-LMS

On reprend le cas de la prédiction linéaire d'un modèle AR2.

Comparaison du RLS et LMS à pas décroissant

Le modèle est un AR2 dont les pôles complexes sont définis par $\rho = 0.95$ et $\theta = 0.04\pi$. Les paramètres sont

$$J_{min} = 0.0034, \text{cond}(R) = 216, \quad n = 500 \quad W_{opt} = [1.8850, -0.9025]$$

Le mauvais conditionnement de R entraîne une lenteur désespérante du LMS (sur ces données, après 4000 itérations le $w_1(n)$ n'aura pas dépassé 1.2), tandis que le RLS termine à nouveau sa convergence en 150 itérations.

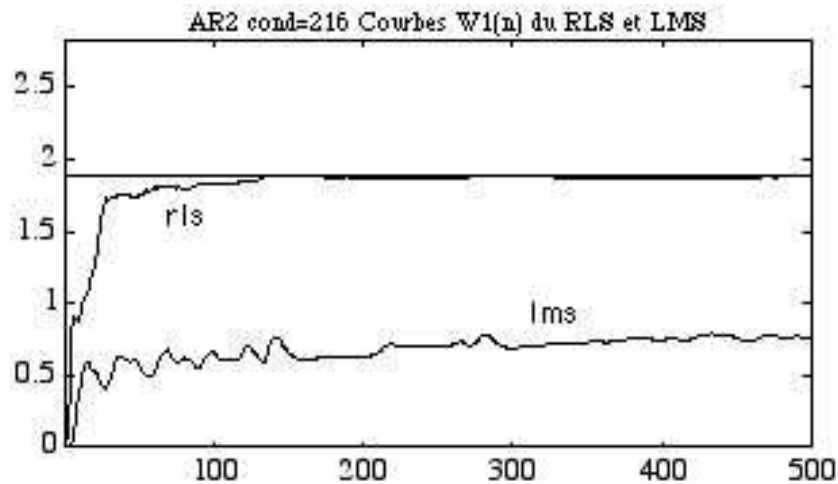


Figure 2.3. LMS et RLS à pas décroissant

RLS à pas fixe : erreur quadratique selon μ

Le modèle est un AR2 de pôles $\rho = 0.9$ et $\theta = 0.4\pi$, déjà utilisé pour le LMS. On réalise pour différents pas μ des moyennes sur 50 réalisations de séquences de taille $n = 200$.

Les valeurs stationnaires affichées sont obtenues par moyennes statistiques puis moyennes temporelles sur $n = [100, 200]$. On compare le LMS à pas μ (cf. résultats et courbes) et le RLS à facteur d'oubli $\lambda = 1 - \mu$: ceci est cohérent du point de vue normalisation, puisque l'on a les bornes de stabilité approchées

$$\begin{array}{ll} \text{RLS} & 0 < \lambda \leq 1 \Leftrightarrow 0 \leq \mu < 1 \\ \text{LMS} & 0 < \mu < \frac{2}{\text{Tr}(R)} = 1 \quad \text{ici car } \sigma_u^2 = 1, m = 2 \end{array}$$

Dans le cas général, la comparaison logique sera entre $\lambda = 1 - \mu'$ et LMS au pas μ avec $\mu' = \frac{\text{Tr}(R)}{2} \mu$: ceci revient à comparer à pas identique le RLS et le LMS normalisé par $E(\|U_n\|^2)$. Les données sont

$$w^* = [0.5562, -0.81] \quad J_{\min} = 0.3114 \quad \text{cond}(R) = 1.88 \quad n_{\text{stat}} = 50 \quad n = 200$$

μ	$E(w_1)$	$E(w_2)$	$\lim \text{var}W$	EQM
0.4	0.5073	-0.7782	0.2554	0.5052
0.35	0.5064	-0.7692	0.2126	0.4713
0.30	0.5178	-0.7682	0.1696	0.4390
0.20	0.5112	-0.7625	0.1102	0.3885
0.10	0.5391	-0.7845	0.0431	0.3417
0.05	0.5481	-0.7864	0.0217	0.3262
0.02	0.5407	-0.8048	0.0097	0.3169
0.01	0.5559	-0.8002	0.0054	0.3138
0.005	0.5593	-0.8001	0.0054	0.3206

Tableau 2.3. Simulation du RLS à oubli $\lambda = 1 - \mu$ sur 50 réalisations

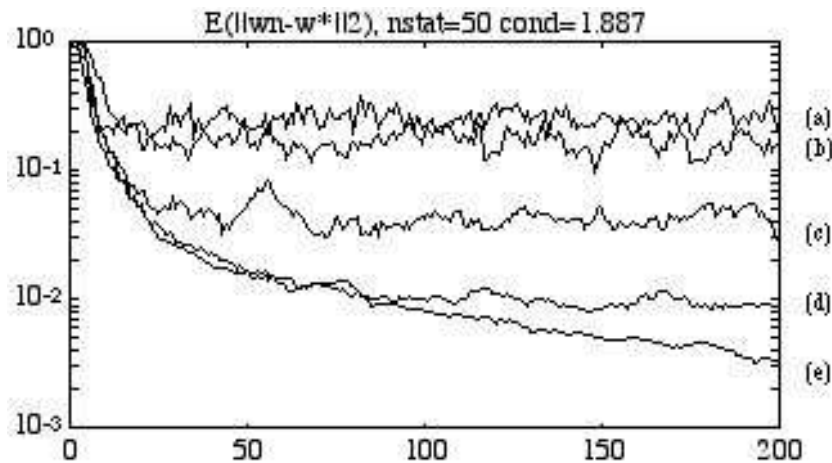


Figure 2.4. Erreur quadratique du RLS selon μ : (a) 0.4 (b) 0.3 (c) 0.1 (d) 0.02 (e) 0.005

Erreur quadratique stationnaire, EQMR et stabilité

On constate la stabilisation de $E(\|w(n) - w^*\|^2)$ à un niveau égal approximativement à $\mu/2$ pour $\mu = 0.4$ à 0.01 , conforme à l'estimation théorique (2.54). Le temps de stabilisation en m.q. augmente quand le pas μ diminue, et pour $\mu = 0.005$ la durée

$n = 200$ est insuffisante pour atteindre ce niveau stationnaire : mais contrairement au LMS, la phase initiale de descente (le transitoire) se fait à pente constante, il n'est pas ralenti par la diminution du pas.

La comparaison avec le LMS sur les mêmes données (cf. figure 2.2) montre une convergence plus rapide du RLS ; un niveau d'erreur quadratique et d'EQMR inférieur pour le RLS à pas identique, et surtout *une meilleure résistance du RLS à l'instabilité*, puisque la stabilisation a encore lieu aux grands pas 0.4 et 0.35 pour lesquels le LMS divergeait nettement.

Vitesse transitoire du RLS- λ

Les courbes de la figure (2.4) confirment une relative indépendance de la durée transitoire du RLS par rapport au facteur d'oubli lorsque $\lambda \rightarrow 1$, et le maintien dans ces conditions d'une convergence rapide selon (2.50). Les courbes de la figure (2.5) montrent ce transitoire rapide, même pour un pas $\mu = 10^{-5}$: on a représenté en échelle logarithmique la droite de pente asymptotique $\log \lambda$ quasi-nulle qui décrit la convergence de $\|R_n - R\|$, la courbe d'erreur du DRLS (exhibant la convergence superlinéaire de (2.50)) et la courbe moyennée $\|\theta_n - \theta^*\|$ d'erreur du RLS qui, après un transitoire rapide suivant le DRLS, se stabilise à un niveau fixe lié à l'erreur quadratique stationnaire.

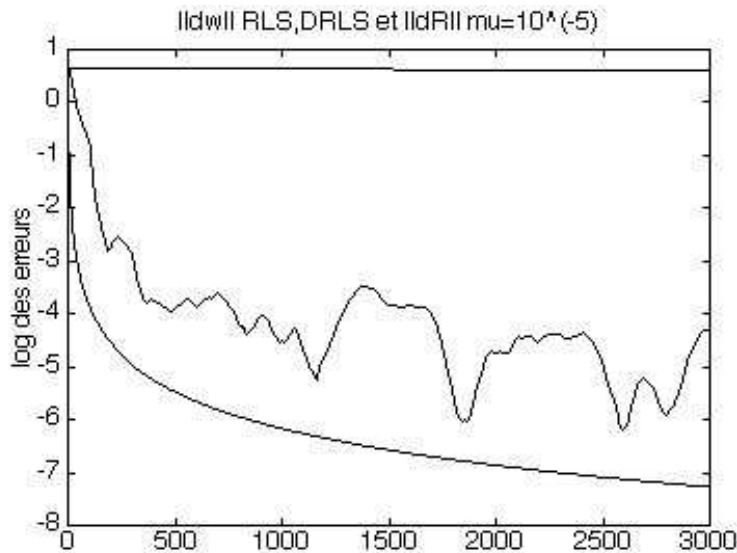


Figure 2.5. Transitoire du RLS $\mu = 10^{-5}$

Conclusion sur RLS

L'algorithme newtonien RLS présente donc l'avantage d'une vitesse de convergence rapide et indépendante du conditionnement des données. Il est de plus peu sensible, pour sa vitesse, au choix du pas de calcul (ou du facteur d'oubli λ).

A facteur d'oubli λ (correspondant à un pas fixe $\mu = 1 - \lambda$), il se traduit par une fluctuation résiduelle proportionnelle à μ , comme le LMS. Mais on peut choisir le pas très petit sans allonger la durée transitoire de convergence : ce choix dépendra donc uniquement de la "largeur de fenêtre équivalente" $W = \frac{1}{1-\lambda}$, représentant la durée de stationnarité supposée des données traitées.

Sa limitation est la complexité de calcul en $O(M^2)$ opérations par itération, qui empêche l'implémentation temps réel pour des valeurs élevées de la taille M du filtre. Pour remédier à cette difficulté, on construira des *algorithmes de Moindres Carrés rapides* : c'est l'objet du chapitre 4.

2.3. Algorithmes de projection affine

OBJECTIF.— On a vu les limitations intrinsèques des deux classes d'algorithmes précédentes : les gradients LMS, par la lenteur de convergence dans les cas mal conditionnés, et les Newton RLS par la complexité de calcul. Les variantes étudiées ici (APA puis treillis) visent à construire un compromis intermédiaire entre les deux, plus rapide que le LMS et moins complexe que le RLS.

De façon similaire à la méthode des Moindres Carrés où le critère statistique de la fonction de coût $J(w)$ est remplacé par un critère géométrique des moindres carrés, l'algorithme de projection affine correspond à une méthode bloc de traitement des données. Il possède un caractère déterministe dépendant des données car il minimise, à chaque itération, un critère géométrique.

2.3.1. Introduction

Dérivation de l'algorithme APA d'ordre P

Considérons la relation suivante $w_n = w_{n-1} + \Delta w_n$ correspondant à la mise à jour des M coefficients d'un filtre adaptatif. On souhaite trouver l'incrément Δw_n permettant d'annuler les P erreurs *a posteriori* les plus récentes, soit :

$$Y_n - \Lambda_n^T w_n = 0 \Leftrightarrow \Lambda_n^T \Delta w_n = E_n \quad (2.61)$$

où $Y_n = (y(n) \cdots y(n - P + 1))^T$ représente le vecteur des P dernières observations, $\Lambda_n = (U_n \cdots U_{n-P+1})$ représente la matrice des P derniers vecteurs d'observation et $E_n = (y(n) - U_n^T w_{n-1} \cdots y(n - P + 1) - U_{n-P+1}^T w_{n-1})^T$ désigne le vecteur des P erreurs *a priori* à l'instant n .

En considérant que $P < M$ (où M est l'ordre du filtre adaptatif), le système à résoudre de P équations à M inconnues est un système sous-déterminé pour lequel, parmi l'ensemble des solutions Δw_n admissibles, la solution à norme minimale conduit à un incrément unique donné par $\Delta w_n = \Lambda_n [\Lambda_n^T \Lambda_n]^{-1} E_n$ où la matrice carrée $\Lambda_n^T \Lambda_n$, définie non-négative, est inversible lorsqu'elle est de rang P . L'algorithme ainsi obtenu est donné par :

Algorithme APA

$$\begin{aligned} E_n &= Y_n - \Lambda_n^T w_{n-1} \\ w_n &= w_{n-1} + \Lambda_n [\Lambda_n^T \Lambda_n]^{-1} E_n \end{aligned} \quad (2.62)$$

Relaxation de l'algorithme APA

En pratique, un paramètre de relaxation μ est introduit afin de pondérer l'incrément Δw_n ce qui conduit à la relation de mise à jour suivante $w_n = w_{n-1} + \mu \Delta w_n$. On obtient alors l'algorithme suivant :

APA relaxé

$$\begin{aligned} E_n &= Y_n - \Lambda_n^T w_{n-1} \\ Z_n &= [\Lambda_n^T \Lambda_n]^{-1} E_n \\ w_n &= w_{n-1} + \mu \Lambda_n Z_n \end{aligned} \quad (2.63)$$

En supposant que le vecteur d'observation est donné par $Y_n = \Lambda_n^T w^* + B_n$, où B_n représente le vecteur des P derniers échantillons de bruit, on montre que le vecteur d'écart à l'optimum des coefficients du filtre $V_n = w_n - w^*$ suit la règle

$$V_n = P_n^\perp V_{n-1} + Q_n B_n \quad (2.64)$$

où $P_n^\perp = I - \mu \Lambda_n [\Lambda_n^T \Lambda_n]^{-1} \Lambda_n^T$ et $Q_n = \mu \Lambda_n [\Lambda_n^T \Lambda_n]^{-1}$. La matrice $P_n = \Lambda_n [\Lambda_n^T \Lambda_n]^{-1} \Lambda_n^T$ correspond à un opérateur de projection sur un sous-espace linéaire constitué par les vecteurs colonnes $\{U_n, U_{n-1}, \cdots, U_{n-P+1}\}$ de la matrice Λ_n . La matrice $P_n^\perp = I - P_n$ représente alors une matrice de projection orthogonale à P_n et la relation (2.64) définit une projection (relaxée) du vecteur V_{n-1} sur un sous-espace affine. En conséquence, l'algorithme donné par (2.62), initialement proposé dans [Ozeki 84], est dénommé algorithme de projection affine d'ordre P . Lorsque cette projection est relaxée, les relations (2.63) définissent la version relaxée de cet algorithme.

Interprétation géométrique de l'algorithme APA

En considérant l'hyperplan $\Pi_n = \{w \in \mathbb{R}^L / y(n) - U_n^T w = 0\}$, l'algorithme NLMS permet au cours des itérations successives de construire une suite de vecteurs $\{w_n, n \geq 0\}$ qui converge vers le filtre optimal w^* . Par définition, ce filtre optimal appartient à l'intersection de tous les hyperplans $\{\Pi_n, n \geq 0\}$. Ainsi, l'algorithme NLMS projette le vecteur w_{n-1} sur Π_n colinéairement à U_n pour obtenir le vecteur w_n qui vérifie l'inégalité $\|V_{n+1}\| \leq \|V_n\|$ (cf. figure (2.6)). On note alors que la vitesse de convergence du NLMS est fortement liée à la valeur de l'angle ϕ entre les hyperplans Π_n et Π_{n-1} . Cet angle, défini par $\cos(\phi) = \frac{\langle U_n, U_{n-1} \rangle}{\|U_n\| \|U_{n-1}\|}$, est gouverné par le degré de corrélation existant entre les vecteurs successifs U_n et U_{n-1} .

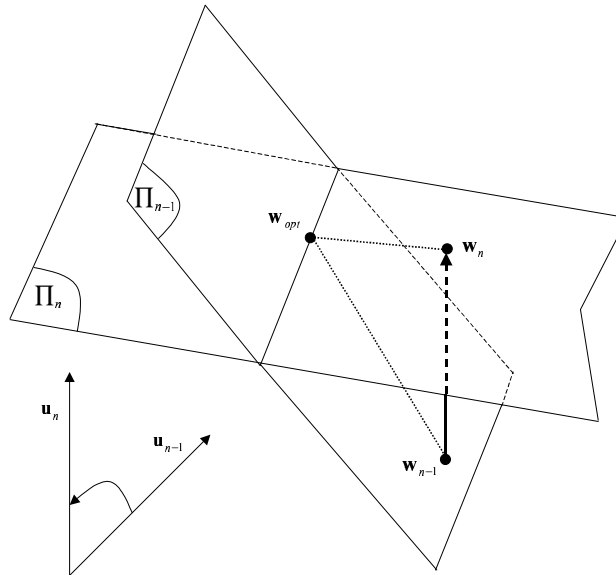


Figure 2.6. Interprétation géométrique de l'algorithme NLMS $M = 3$

Pour améliorer la vitesse de convergence de l'algorithme NLMS en présence de signaux corrélés, il est envisageable de modifier la direction d'adaptation des coefficients du filtre. L'idée de base de l'algorithme APA consiste à utiliser une projection, non plus colinéairement à U_n , mais dans la direction orthogonale à $\bigcap_{k=0, \dots, P-1} \Pi_{n-k}$.

Cette projection, dite d'ordre P , permet de réduire la norme $\|V_n\|$ et donc d'accroître la vitesse de convergence de l'algorithme en comparaison à celle du NLMS. La figure (2.7) illustre la mise à jour des $M = 3$ coefficients du filtre pour l'algorithme APA d'ordre $P = 2$.

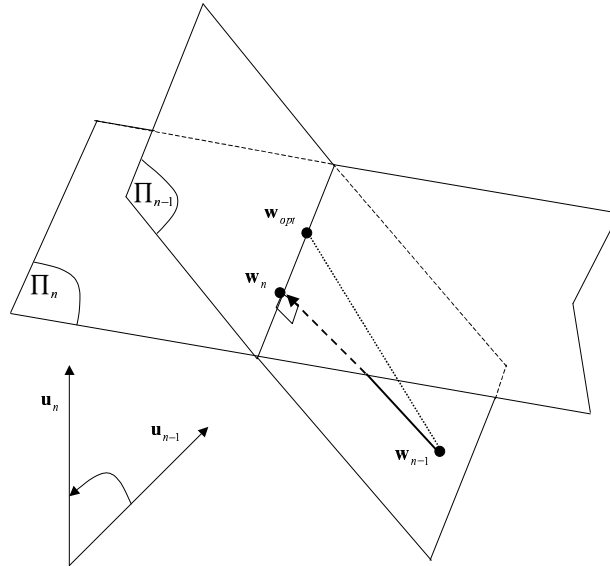


Figure 2.7. Interprétation géométrique de l'algorithme APA $M = 3$ et $P = 2$

2.3.2. Liens avec les autres familles d'algorithmes

Lien avec l'algorithme NLMS

Il est évident que les relations décrivant l'algorithme APA conduisent, lorsque l'ordre de projection P est égal à l'unité, aux relations suivantes :

$$\begin{aligned} e(n) &= y(n) - U_n^T w_{n-1} \\ w_n &= w_{n-1} + \mu [U_n^T U_n]^{-1} U_n e(n) \end{aligned} \quad (2.65)$$

Cette relation correspond à la définition de l'algorithme NLMS qui permet, lorsque le pas d'adaptation est égal à l'unité, d'annuler à chaque itération n l'erreur d'estimation *a posteriori* définie par : $\varepsilon(n) = y(n) - U_n^T w_n$.

Lien avec l'algorithme des moindres carrés

Dans le cas où $P \geq M$, le problème précédent qui consiste à trouver l'incrément Δw_n pour annuler les P erreurs *a posteriori* les plus récentes [soit $\Lambda_n^T \Delta w_n = E_n$] conduit à un système sur-déterminé. L'unique solution au sens des moindres carrés est donnée par $\Delta w_n = [\Lambda_n \Lambda_n^T]^{-1} \Lambda_n E_n$. En pratique, la mise à jour des coefficients du filtre est réalisée par un algorithme de type RLS.

Lien avec l'algorithme de transformation à faible rang

On peut interpréter l'algorithme APA comme un cas particulier des algorithmes de transformation à faible rang [Raghotaman 2000] où les propriétés de décorrélation de la matrice de transformation sont utilisées pour approcher l'inverse de la matrice de corrélation des données. Pour cela, considérons une matrice de transformation Q , matrice carrée unitaire de rang plein ($Q Q^T = I = Q^T Q$). Il existe deux possibilités d'introduire cette matrice Q au sein de la relation (2.61) : $Q Q^T \Lambda_n^T \Delta w_n = E_n$ ou bien $\Lambda_n^T Q Q^T \Delta w_n = E_n$. Soit $Q = [Q_1 \ Q_2]$ où Q_1 est une matrice de dimension $P \times M$ telle que $Q_1^T \Lambda_n^T \neq 0$. Le système (2.61) peut être transformé suivant $Q_1^T \Lambda_n^T \Delta w_n = Q_1^T E_n$.

Si l'on considère le système d'origine (2.61) sur-déterminé ($P \geq M$), la transformation Q_1 le convertit en un système sous-déterminé, $1 \leq K < M$. En choisissant $Q = I_{P \times P}$ et $Q_1 = [I_{K \times K} \ 0_{K \times (P-K)}]^T$, il est possible de convertir (2.61) de façon à obtenir comme solution l'algorithme NLMS, puis l'algorithme APA, puis l'algorithme RLS en modifiant simplement la valeur du paramètre K . Si $Q_1 = [1 \ 0_{1 \times (P-1)}]^T$, l'application de la transformation à faible rang extrait uniquement la première équation du système (2.61), ce qui correspond à l'opération réalisée par l'algorithme NLMS. Si $Q_1 = \begin{bmatrix} 1 & 0 & 0_{1 \times (P-2)} \\ 0 & 1 & 0_{1 \times (P-2)} \end{bmatrix}^T$, la transformation extrait uniquement les 2 premières équations ce qui correspond à l'algorithme APA d'ordre 2. Cette propriété peut-être facilement étendue à des ordres supérieurs à 2. En étendant la transformation à $Q_1 = I_{P \times P}$ seules les P dernières équations du système (2.61) sont considérées ce qui est équivalent à l'algorithme RLS (pour $P \geq M$) à fenêtre rectangulaire.

2.3.3. Régularisation directe de la matrice de covariance des données

La relation (2.64) montre que les propriétés de l'algorithme APA sont directement liées à celles de la matrice des données $\Lambda_n^T \Lambda_n$. Il apparaît évident que le terme de bruit $Q_n B_n$ de la relation (2.64) se trouve amplifié lorsque cette matrice est mal conditionnée ce qui conduit, durant ces périodes, à un désajustement important du filtre identifié. Ce problème peut être résolu en introduisant une procédure de régularisation directe de la matrice $\Lambda_n^T \Lambda_n$ qui consiste à introduire, au sein de la diagonale de la matrice de covariance des données, un paramètre $\delta \in \mathbb{R}^{*+}$ ce qui conduit à l'équation de mise à jour des coefficients du filtre $w_n = w_{n-1} + \mu \Lambda_n [\Lambda_n^T \Lambda_n + \delta I]^{-1} E_n$. Soit la décomposition en valeur singulière $\Lambda_n = S_n \Sigma_n T_n^T$ de la matrice $\Lambda_n \in \mathbb{R}^{M \times P}$, où $S_n \in \mathbb{R}^{M \times M}$ et $T_n \in \mathbb{R}^{P \times P}$ sont deux matrices unitaires ($S_n S_n^T = I$ et $T_n T_n^T = I$)

et où $\Sigma_n \in \mathbb{R}^{M \times P}$ est donnée, lorsque $P \leq M$, par

$$\Sigma_n = \begin{bmatrix} \lambda_{n,1} & & & & \\ & \lambda_{n,2} & & & \\ & & \ddots & & \\ & & & & \lambda_{n,P} \\ \hline & & & & 0_{(M-P) \times P} \end{bmatrix} \quad (2.66)$$

on montre alors que la matrice de projection $P_n^\perp = \mathbf{I} - \mu \Lambda_n [\Lambda_n^T \Lambda_n + \delta \mathbf{I}]^{-1} \Lambda_n^T$ de l'algorithme APA relâché admet la décomposition suivante :

$$P_n^\perp = S_n \text{diag}(1 - \mu \frac{\lambda_{n,1}^2}{\lambda_{n,1}^2 + \delta}, \dots, 1 - \mu \frac{\lambda_{n,P}^2}{\lambda_{n,P}^2 + \delta}, 1_{M-P}) S_n^T. \quad (2.67)$$

Le facteur d'amplification du bruit d'observation est gouverné par la structure de la matrice $Q_n = \mu \Lambda_n [\Lambda_n^T \Lambda_n + \delta \mathbf{I}]^{-1}$ de valeurs valeurs singulières $\mu \frac{\lambda_{n,k}}{\lambda_{n,k}^2 + \delta}$, $\forall k = 1, \dots, P$. La limitation du facteur d'amplification du bruit de fond est réalisée à la fois par la procédure de relaxation et par celle de régularisation. En effet, les valeurs singulières de Q_n sont d'autant plus faibles que le paramètre de relaxation μ est faible et que le paramètre de régularisation δ est fort. Un choix judicieux de ce dernier paramètre consiste à prendre $\delta = E[b^2(n)]$. Ainsi, la mise à jour des coefficients du filtre n'est effectuée que suivant les directions pour lesquelles le rapport signal à bruit est important [soit $\frac{\lambda_{n,k}^2}{\lambda_{n,k}^2 + \delta} \approx 1$], tandis que l'adaptation est minimisée dans celles pour lesquelles le rapport signal à bruit est faible [soit $\frac{\lambda_{n,k}^2}{\lambda_{n,k}^2 + \delta} \approx \frac{\lambda_{n,k}^2}{\delta}$].

L'introduction du facteur de relaxation μ conduit à un vecteur d'erreur *a posteriori* $\varepsilon_n = Y_n - \Lambda_n^T w_n$ donné par $\varepsilon_n = [\mathbf{I} - \mu \Lambda_n^T \Lambda_n (\Lambda_n^T \Lambda_n + \delta \mathbf{I})^{-1}] E_n$. Dans le cas particulier où δ est négligeable devant les valeurs propres $\lambda_{n,k}^2$, $\forall k = 1, \dots, P$ de la matrice $\Lambda_n \Lambda_n^T$, cette équation conduit à l'approximation

$$\varepsilon_n \simeq (1 - \mu) E_n. \quad (2.68)$$

Cette dernière relation se révèle être une égalité lorsque l'on utilise la version non-régularisée de l'algorithme APA. Dans ce dernier cas, il est évident d'après (2.68) que le choix d'un pas d'adaptation égal à l'unité permet d'annuler simultanément P erreurs *a posteriori* ce qui correspond à la solution de (2.61).

2.3.4. Convergence de l'algorithme APA

Contrairement à l'algorithme NLMS dont les propriétés de convergence ont été étudiées de façon intensive, peu d'études ont porté sur la convergence de l'algorithme de projection affine. Des résultats de convergence de l'algorithme APA non régularisé avec un pas d'adaptation égal à 1 ont été obtenus par [De Almeida 03] en modélisant le signal d'entrée par un processus gaussien auto-régressif. Des expressions analytiques de la courbe d'apprentissage et du comportement asymptotique de l'erreur quadratique moyenne de l'algorithme APA régularisé (méthode directe) sont présentés par [Shin 04] sans modèle particulier des signaux d'entrée et sans restreindre l'analyse à des signaux gaussiens. Nous donnerons ici les résultats présentés par [Sankaran 00] qui, bien que de portée moins large, ont l'avantage de pouvoir exprimer les principales propriétés de convergence de l'algorithme APA non-régularisé.

Convergence en moyenne quadratique

Supposons [Slock, 1993] que les vecteurs d'entrée $\{U_n\}$ sont centrés, indépendants et identiquement distribués, et que la matrice de covariance est donnée par $R = E[U_n U_n^T] = S \Sigma S^T$ avec $\Sigma = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M)$ et $S = \{\nu_1, \dots, \nu_M\}$, les paramètres $\{\lambda_i\}$ et $\{\nu_i\}$ représentant les valeurs propres et vecteurs propres de R . On suppose de plus que l'observation est donnée par $y(n) = U_n^T w_{opt} + b(n)$ avec $b(n)$ bruit blanc centré de variance σ_b^2 et indépendant de $\{U_n\}$. Dans ces conditions, l'erreur quadratique moyenne de l'algorithme APA non-régularisé est donnée par $J(n) = \sigma_b^2 + \sum_{i=1}^M \lambda_i \tilde{\lambda}_{n,i}$ avec :

$$\begin{cases} \tilde{\lambda}_{n,i} = [1 - \mu(2 - \mu) \beta_i] \tilde{\lambda}_{n-1,i} + \mu^2 \sigma_b^2 E\left(\frac{1}{r^2}\right) \beta_i \\ \beta_i = 1 - (1 - p_i)^P \quad \text{et} \quad p_i = \lambda_i / \text{Tr}(R) \end{cases} \quad (2.69)$$

Ainsi, le choix $\mu \in [0, 2]$ constitue une condition suffisante pour assurer la convergence de l'erreur quadratique moyenne vers l'erreur asymptotique $J(\infty) = \sigma_b^2 [1 + \frac{\mu}{2-\mu} E(\frac{1}{r^2}) \text{Tr}(R)]$. De la relation (2.69), nous pouvons constater que la convergence de l'erreur quadratique moyenne est gouvernée par la valeur du produit $\mu(2 - \mu)\beta_i$ avec $\mu(2 - \mu) \leq 1$ (car l'algorithme est supposé stable) et $\beta_i = 1 - (1 - p_i)^P \leq 1$. La convergence sera donc d'autant plus rapide que $(1 - \mu(2 - \mu)\beta_i)$ est proche de zéro soit donc lorsque les paramètres μ et β_i sont proches de l'unité.

Dans (2.69), les quantités μ et $(2 - \mu)$ conduisent à la même vitesse de convergence. Par contre, le compromis entre vitesse de convergence et erreur en excès peut être optimisé en sélectionnant le pas d'adaptation dans l'intervalle $[0, 1]$ ce qui limite l'erreur quadratique moyenne en excès qui est proportionnelle à la valeur de μ . Si l'on analyse l'influence de l'ordre P de la projection affine, on constate qu'un ordre élevé

améliore la vitesse de convergence puisque β_i tend vers la valeur unité. L'algorithme APA converge donc plus rapidement que l'algorithme NLMS (cas $P = 1$) quelque soit la structure du signal d'entrée. Dans le cas où le signal d'entrée correspond à un bruit blanc, l'erreur quadratique moyenne (exprimée en décibels) décroît linéairement d'environ 20 dB tous les $5M/P$ échantillons pour un pas d'adaptation égal à l'unité. Ceci est à comparer avec la décroissance de 20 dB tous les $5M$ échantillons pour le NLMS. Ces éléments suggèrent l'idée d'améliorer la vitesse de convergence de ces algorithmes en travaillant avec un nombre limité de coefficients durant la phase de convergence initiale de l'algorithme, puis en augmentant progressivement l'ordre du filtre M jusqu'à atteindre la valeur désirée.

Convergence en moyenne

Considérons la décomposition de la valeur moyenne $E[V_n]$ du vecteur d'écart à l'optimum V_n sur la base orthonormée des vecteurs propres $\{\nu_1 \nu_2 \cdots \nu_M\}$, cette décomposition est donnée par $\{\rho_{n,i} = \nu_i^T E[V_n], i = 1, 2, \dots, M\}$. Avec cette notation, la convergence en moyenne de ce vecteur d'écart est gouvernée par la relation $\rho_{n+1,i} = (1 - \mu\beta_i) \rho_{n,i}$.

La convergence vers zéro de $\rho_{n,i}$ est assurée si et seulement si $|1 - \mu\beta_i| < 1$. Pour des signaux suffisamment riches ($p_i \neq 0, \forall i$), on a $0 \leq \beta_i \leq 1$. Ainsi, une condition suffisante pour assurer la convergence du vecteur $\rho_n = [\rho_{n,1} \rho_{n,2} \cdots \rho_{n,M}]^T$ est $\mu \in [0, 2]$. Inversement, lorsque $\mu \in [0, 2]$, le vecteur ρ_n tend asymptotiquement vers zéro de façon exponentielle et donc $E[V_n]$ converge vers zéro puisque l'ensemble $\{\nu_1, \nu_2, \dots, \nu_M\}$ forme une base orthonormée. L'algorithme APA constitue donc un estimateur asymptotiquement non-biaisé des coefficients optimaux w^* .

2.3.5. Complexité algorithmique et algorithmes rapides

Dans sa version non-régularisée (2.63), la complexité de l'algorithme APA est de $2MP + KP^2$ multiplications par échantillon, où K désigne une constante dépendant de la stratégie choisie pour inverser la matrice $\Lambda_n^T \Lambda_n$. Une valeur de $K = 7$ est obtenue lorsque l'algorithme de Levinson généralisé est utilisé pour résoudre le système d'équations $Z_n = [\Lambda_n^T \Lambda_n]^{-1} E_n$ présent au sein de l'algorithme APA. Pour diminuer cette complexité, il est possible de ne mettre à jour les coefficients qu'une seule fois tous les P échantillons successifs ce qui conduit à l'algorithme PRA (pour Partial Rank Algorithm) d'une complexité de $2M + KP$ multiplications par échantillon.

L'algorithme FAP (pour Fast AP algorithm) constitue une version rapide approchée de l'algorithme de projection affine régularisé par la méthode directe [Gay 93b], [Gay 95] et [Tanaka 95]. La complexité de cet algorithme est de $2M + 20P$ multiplications par échantillon mais peut être réduite à $2M + 14P$ multiplications par

échantillon lorsque $\delta = 0$, soit une complexité proche de celle de l'algorithme NLMS. Cependant, pour limiter les problèmes d'instabilité numérique inhérents à l'algorithme FAP, il est nécessaire d'introduire des procédures de stabilisation ce qui conduit à une complexité de $2M + 30P$ (resp. $2M + 20P$) pour la version FAP avec relaxation (resp. sans relaxation).

Pour réduire la complexité en comparaison du FAP, il est nécessaire d'utiliser des procédures optimisées de filtrage transverse à base de transformées de Fourier rapides. De tels algorithmes correspondent à une version exacte par blocs de l'algorithme FAP (dénommée Block Exact Fast Affine Projection Algorithm ou BEFAP)[Tanaka, 99]. Pour $M = 2000$ et $P = 8$, la complexité de l'algorithme BEFAP est de 2160 multiplications par échantillon ce qui est sensiblement équivalent à celle de l'algorithme FAP pour un filtre de taille $M = 1000$. Cet avantage est obtenu au prix d'une augmentation du retard de traitement introduit par la structure bloc de l'algorithme BEFAP.

Enfin, une version exacte par blocs de l'algorithme APA a été proposée [Rombouts 02]. L'algorithme Bloc Exact Affine Projection Algorithm (BEAPA) ainsi obtenu repose sur les mêmes principes généraux que l'algorithme BEFAP mais calcule, à chaque itération, l'ensemble des composantes du vecteur d'erreur *a priori* et non plus une approximation de celles-ci.

2.4. Structures en treillis

Alors que les algorithmes pour les structures adaptatives transversales sont principalement basés sur des récurrences temporelles, les structures en treillis font appel à des récurrences temporelles et sur l'ordre. Même si de telles structures demandent davantage d'opérations que leur équivalent sous forme transverse, les filtres en treillis possèdent des propriétés particulièrement intéressantes en vue de leur implémentation, en raison notamment de leur structure modulaire et de leur moindre sensibilité aux effets de quantification et de représentation en précision finie.

2.4.1. Introduction

Prédiction linéaire directe et rétrograde

Considérons le problème de la prédiction linéaire appliqué à un processus aléatoire stationnaire $u(n)$ à valeurs dans \mathbb{R} et de moyenne nulle, on montre à partir de l'algorithme de Levinson que les erreurs de prédiction linéaire directe $f_m(n)$ et rétrograde $r_m(n)$ d'ordre m et $m - 1$ sont liées par la relation suivante

$$\begin{aligned} f_m(n) &= f_{m-1}(n) - \Gamma_m r_{m-1}(n-1) \\ r_m(n) &= r_{m-1}(n-1) - \Gamma_m f_{m-1}(n) \end{aligned} \quad (2.70)$$

où Γ_m est un coefficient réel. Ces relations définissent une procédure permettant, par récurrence sur l'ordre m du prédicteur, de calculer les erreurs de prédiction directe $f_m(n)$ et rétrograde $r_m(n)$, ceci sans résoudre directement le système des équations de normales. A l'initialisation pour $m = 0$, soit lorsqu'aucune prédiction n'est réalisée, on choisit les valeurs initiales $f_0(n) = r_0(n) = u(n)$. Cette procédure, itérée pour $i = 0, \dots, m$ conduit à la structure symétrique en treillis de la figure (2.8).

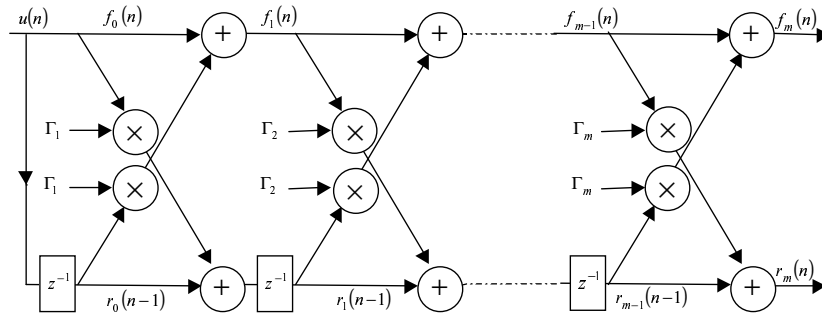


Figure 2.8. Filtre en treillis de prédiction d'erreurs à l'ordre m

Les coefficients Γ_m sont appelés les *coefficients de réflexion* par analogie avec la théorie des lignes de transmission, ou encore *coefficients de corrélation partielle* (PARCOR). Leur valeur optimale Γ_m^* est obtenue à partir de (2.70) en minimisant la puissance moyenne des erreurs directe et rétrograde, soit

$$\Gamma_m^* = E[r_{m-1}(n-1) f_{m-1}(n)] / E[r_{m-1}^2(n-1)] \quad (2.71)$$

Interprétation géométrique

En appliquant le théorème de projection à l'erreur de prédiction linéaire rétrograde obtenue à l'ordre j , on obtient les relations $E[r_j(n) u(n-j+i)] = 0$ pour $i = 0, \dots, j$ ce qui mène à la relation

$$E[r_j(n) r_k(n)] = E[r_j^2(n)] \delta_{jk} \quad (2.72)$$

où δ_{jk} représente le symbole de Kronecker. Dans l'espace de Hilbert $H = L^2(\Omega)$ muni du produit scalaire $\langle X, Y \rangle = E[XY]$, la relation (2.72) exprime l'orthogonalité des erreurs de prédiction linéaire rétrograde disponibles, à un même instant n , aux différents étages du treillis. Les erreurs de prédiction linéaire rétrograde forment ainsi une base orthogonale $B_m = [r_0(n), \dots, r_{m-1}(n)]$ et la construction du treillis correspond à l'orthogonalisation de l'espace des observations $[u(n), \dots, u(n-m+1)]$ à partir de la procédure de Gram-Schmidt. Les coefficients du prédicteur rétrograde et le signal d'entrée sont liés par $r^{(m)}(n) = L U_m(n)$ où la matrice triangulaire inférieure $L = [b_0(n), \dots, b_{m-1}(n)]^T$ est constituée des prédicteurs linéaires rétrogrades $b_i(n)$

obtenus à l'instant n pour des ordres $i < m$. La matrice de covariance des erreurs de prédiction rétrogrades admet alors la décomposition LU (*lower upper*) de Cholesky suivante

$$\mathbb{E} \left[r^{(m)}(n) r^{(m)}(n)^T \right] = L \mathbb{E} \left[U_m(n) U_m^T(n) \right] L^T = \Lambda \quad (2.73)$$

où Λ est une matrice diagonale telle que $\lambda_{jk} = \mathbb{E} [r_j^2(n)] \delta_{ij}$. En un certain sens, cette dernière relation exprime l'orthogonalité des filtres d'erreurs de prédiction à l'instant n en raison de l'égalité $b_i^T(n) \mathbb{E} [U_m(n) U_m^T(n)] b_j^T(n) = \mathbb{E} [r_i^2(n)] \delta_{ij}$. Par suite, la matrice inverse de Λ est également diagonalisable ce qui conduit à la décomposition LU de l'inverse de la matrice de covariance des données

$$\mathbb{E} \left[U_m(n) U_m^T(n) \right]^{-1} = \left(\Lambda^{-\frac{1}{2}} L \right)^T \left(\Lambda^{-\frac{1}{2}} L \right). \quad (2.74)$$

2.4.2. Le treillis gradient et variantes

Prédiction linéaire adaptative sous forme treillis

Nous nous intéressons ci-dessous à l'implémentation sous forme adaptative de la structure symétrique en treillis stationnaire de la figure 2.8 et nous cherchons les coefficients optimaux Γ_m^* permettant de minimiser la fonction de coût donnée par $J_m(n) = \mathbb{E} [f_m^2(n) + r_m^2(n)]$. A partir de la récurrence sur l'ordre du treillis (2.70), la solution optimale est donnée par

$$\Gamma_m^* = 2 \frac{\mathbb{E} [f_{m-1}(n) r_{m-1}(n-1)]}{\mathbb{E} [f_{m-1}^2(n) + r_{m-1}^2(n-1)]} \quad (2.75)$$

L'algorithme du treillis adaptatif correspond à la recherche de la solution (2.75) par une méthode itérative basée sur l'algorithme du gradient. Il est obtenu en introduisant dans (2.70) des coefficients de réflexion $\Gamma_m(n)$ variant au cours du temps, et en évaluant le gradient de la fonction de coût ce qui conduit à la relation de mise à jour

$$\Gamma_m(n+1) = \Gamma_m(n) + 2\mu_m \mathbb{E} [f_m(n) r_{m-1}(n-1) + f_{m-1}(n) r_m(n)] \quad (2.76)$$

où μ_m est une constante positive. L'algorithme ainsi obtenu possède des propriétés de convergence spécifiques en raison de la non-linéarité de la récurrence par rapport aux Γ_m , mais aussi du caractère non-quadratique de la fonction de coût. Asymptotiquement, on montre que la stabilité de l'algorithme est assurée si $|1 - 2\mu_m D_m| < 1$ où $D_m = \mathbb{E} [f_{m-1}^2(n) + r_{m-1}^2(n-1)] = 2 \mathbb{E} [f_{m-1}^2(n)]$ par stationnarité et symétrie du treillis. Etant donné que $D_0 \geq D_1 \geq \dots \geq D_m$, les pas d'adaptation des étages d'ordres élevés doivent être choisis plus grands que ceux des étages inférieurs.

L'implémentation de l'algorithme du *treillis LMS* consiste à utiliser une approximation stochastique du gradient, ce qui conduit à

$$\Gamma_m(n+1) = \Gamma_m(n) + \mu_m (f_m(n) r_{m-1}(n-1) + f_{m-1}(n) r_m(n)) \quad (2.77)$$

En choisissant des pas d'adaptation μ_m adéquats, cet algorithme possède une vitesse de convergence plus rapide en général que son équivalent sous forme transversale en raison des propriétés d'orthogonalisation des étages successifs du treillis et de la diagonalisation de la matrice de covariance (2.73). Cette vitesse de convergence est moins dépendante du conditionnement de la matrice de covariance du processus d'entrée. Cependant, son implémentation se heurte à la nécessité de connaître *a priori* la statistique des D_m ce qui se révèle particulièrement délicat en environnement non-stationnaire. Nous verrons que ce problème peut être résolu en utilisant une version "normalisée" de l'algorithme d'adaptation.

Forme treillis adaptatif d'un filtre de Wiener RIF

Il est possible d'exploiter les propriétés d'orthogonalité intrinsèques aux structures en treillis dans un cadre plus général du filtrage linéaire sous forme transversale. La figure 2.9 représente la réalisation sous forme treillis d'un filtre de Wiener. Elle est constituée de deux étages distincts : un premier étage produit les erreurs de prédiction linéaire directe $f_i(n)$ et rétrograde $r_i(n)$ pour chaque ordre. Le second étage utilise, au sein d'une structure transversale, les erreurs de prédiction linéaire rétrograde $r_i(n)$ calculées aux ordres $i = 1, \dots, m$ de manière à minimiser la puissance de l'erreur $e_m(n) = d(n) - y(n)$.

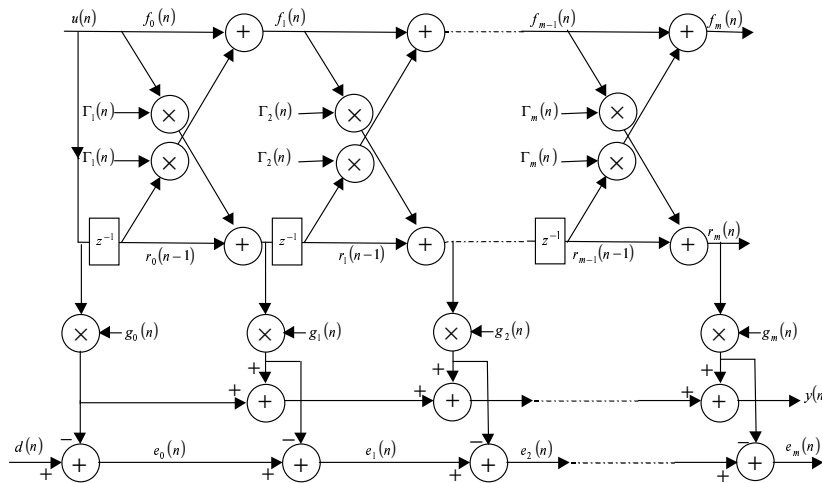


Figure 2.9. Réalisation sous forme treillis d'un filtre de Wiener

Dans le cas stationnaire, lorsque le critère d'optimisation correspond à la minimisation de l'erreur quadratique moyenne $J_m(n) = E[e_m^2(n)]$, le caractère diagonal

de la matrice de covariance des erreurs de prédiction rétrograde donné par (2.73) rend indépendant chaque élément de g , soit

$$g_i^* = \frac{\text{E} [d(n)r_i(n)]}{\text{E} [r_i^2(n)]}, \quad i = 0, 1, \dots, m \quad (2.78)$$

Ces coefficients peuvent être estimés de façon itérative par un algorithme de type LMS déterministe soit $g(n+1) = g(n) - P \nabla [J_m(n)]$, où $P = \text{diag}[\mu_0, \mu_1, \dots, \mu_m]$ est une matrice diagonale définie positive. La convergence de chacun des coefficients $\{g_i(n), i = 0, \dots, m\}$ sera gouvernée par la quantité $(1 - 2\mu_i \text{E} [r_i^2(n)])$. Le choix $2\mu_i = \frac{\alpha}{\text{E} [r_i^2(n)]}$ avec $0 < \alpha < 1$ impose une vitesse de convergence identique pour chacun des coefficients $g_i(n)$, ceci quelle que soit la corrélation du processus d'entrée $u(n)$. Par approximation stochastique du gradient, on obtient

$$\begin{aligned} e_m(n) &= d(n) - g^T(n)r^{(m)}(n) \\ g_i(n+1) &= g_i(n) + \alpha e_m(n)r_i(n)/\text{E} [r_i^2(n)] \quad , i = 0, 1, \dots, m \end{aligned} \quad (2.79)$$

Deux jeux de coefficients sont adaptés en parallèle : les coefficients de réflexion orthogonalisent le signal d'entrée $u(n)$ et les coefficients du filtre transverse $g(n)$ ne sont plus dépendants du conditionnement de la matrice de covariance du processus d'entrée $u(n)$. Cette indépendance n'est cependant valide que lorsque les coefficients de réflexion ont convergé vers leur valeur optimale. Durant la phase initiale de convergence, il y aura donc une certaine dépendance entre les erreurs de prédiction linéaire rétrograde $r_i(n)$. Par ailleurs, les changements dans la statistique du signal de référence $d(n)$ ne viennent affecter que uniquement la seconde section de la structure conjointe de la figure 2.9, la partie prédiction linéaire rétrograde restant inchangée.

En prenant en compte les propriétés d'orthogonalité (2.72) entre les erreurs de prédiction rétrograde, l'erreur $e_m(n)$ dans (2.79) peut être remplacée par $e_i(n) = d(n) - g_i(n)r_i(n)$. La minimisation de la puissance de l'erreur globale $e_m(n)$ est remplacée par la minimisation locale à chaque étage de $\text{E} [e_i^2(n)]$ par rapport à $g_i(n)$.

2.4.3. Le treillis Moindres Carrés

Moindres carrés exacts

On considère maintenant que le critère d'optimisation statistique à chaque étage du treillis est remplacé par une estimation empirique des moindres carrés. Ceci revient à travailler, non plus avec le produit scalaire défini précédemment, mais avec $\langle X, Y \rangle = \frac{1}{n} \sum_{k=1}^n x(k)y(k)$ ce qui conduit à la méthode de Burg. La minimisation de la fonction de coût $J_m(n) = \frac{1}{n} \sum_{k=1}^n [f_m^2(k) + r_m^2(k)]$ est obtenue pour la solution optimale

$$\Gamma_m^*(n) = 2 \frac{\sum_{k=1}^n f_{m-1}(k) r_{m-1}(k-1)}{\sum_{k=1}^n [f_{m-1}^2(k) + r_{m-1}^2(k-1)]} \quad (2.80)$$

En posant $\mu_n = \frac{1}{n}$ et en faisant apparaître des récurrences sur le temps, on obtient l'algorithme de *Burg adaptatif* défini ci-après où l'estimée des coefficients de réflexion $\Gamma_m(n)$ minimisent exactement la fonction de coût $J_m(n)$:

$$\begin{aligned} c_m(n) &= (1 - \mu_n) c_m(n-1) + \mu_n f_{m-1}(n) r_{m-1}(n) \\ d_m(n) &= (1 - \mu_n) d_m(n-1) + \mu_n [f_m^2(n) + r_m^2(n-1)] \\ \Gamma_m(n) &= 2c_m(n)/d_m(n) \end{aligned} \quad (2.81)$$

L'algorithme obtenu correspond à une approximation stochastique d'un algorithme de type quasi-Newton à pas décroissant μ_n . Il possède une propriété de convergence p.s. des $\Gamma_m(n)$ vers les coefficients optimaux (2.75). Par ailleurs, en phase de convergence asymptotique, l'écart quadratique moyen des coefficients à l'optimum est proportionnel à la valeur du pas d'adaptation de l'algorithme.

Moindres carrés à oubli exponentiel

De manière à optimiser le fonctionnement de l'algorithme *Burg adaptatif* en contexte non-stationnaire, il est préférable de considérer un critère de moindres carrés à facteur d'oubli exponentiel λ . Considérons la version stochastique de (2.76) avec le choix $2\mu_m = \alpha/D_m$ (où $0 < \alpha < 1$) ce qui rend l'adaptation indépendante du paramètre m et conduit à une moindre sensibilité de la vitesse de convergence de l'algorithme vis à vis de la dispersion des valeurs propres de la matrice de covariance du processus d'entrée. L'implémentation de la version *normalisée* de l'algorithme du *treillis LMS* consiste à utiliser une approximation stochastique du gradient, soit

$$\Gamma_m(n+1) = \Gamma_m(n) + \frac{\alpha}{\hat{D}_m(n)} [f_m(n) r_{m-1}(n-1) + f_{m-1}(n) r_m(n)] \quad (2.82)$$

combinée à une approximation itérative de la quantité D_m par la méthode des moindres carrés avec facteur d'oubli exponentiel λ

$$\hat{D}_m(n) = (1 - \lambda) \sum_{k=1}^n \lambda^{n-k} [f_{m-1}^2(k) + r_{m-1}^2(k-1)] \quad (2.83)$$

où $0 < \lambda < 1$. Lorsque $\alpha = 1 - \lambda$, l'algorithme en treillis adaptatif obtenu représente une version récursive de la méthode de Burg conduisant à l'estimateur suivant

$$\Gamma_m(n+1) = 2 \frac{\sum_{k=1}^n \lambda^{n-k} f_{m-1}(k) r_{m-1}(k-1)}{\sum_{k=1}^n \lambda^{n-k} [f_{m-1}^2(k) + r_{m-1}^2(k-1)]} \quad (2.84)$$

On retrouve une solution similaire à (2.80) lorsque le critère des moindres carrés est remplacé par un critère des moindres carrés à facteur d'oubli exponentiel λ .

2.5. Performances comparées des algorithmes adaptatifs

Nous présentons ici des résultats de simulations, sur des signaux stationnaires, permettant de comparer en termes de vitesse de convergence (durée du transitoire) et de performance asymptotiques (fluctuation résiduelle), les algorithmes APA et en treillis aux LMS et RLS déjà vus : l'objectif de ces algorithmes était en effet de réaliser un compromis de complexité et de vitesse de convergence entre les gradients LMS et les algorithmes newtoniens RLS.

2.5.1. Performances de APA et NLMS

Comparaisons en convergence initiale

On considère le problème de l'identification de systèmes où l'observation bruitée $y(n) = U_n^T w^* + b(n)$ désigne la réponse d'un système inconnu (modélisé par la réponse impulsionnelle w^* et comportant $M = 30$ coefficients) à un signal d'entrée $u(n)$ modélisé par un modèle AR1 de pôle ρ variable. Le bruit d'observation est gaussien, blanc, centré de variance σ_b^2 . On se place dans les conditions suivantes : pas d'adaptation des algorithmes $\mu = 0.5$, rapport signal à bruit d'observation de 30 dB, valeur initiale des coefficients du filtre $w_{-1} = 0$. Le filtre optimal w^* est tel que sa décomposition sur la base des vecteurs propres de la matrice de corrélation du signal d'entrée $R = E[U_n U_n^T] = V \Sigma V^T$ admet des composantes égales en amplitude (hypothèse de maximum d'entropie) soit

$$w^* = \sqrt{\frac{E[y^2(n)] - \sigma_b^2}{\text{Tr}(R)}} 1_M V^T \quad (2.85)$$

avec $1_M = [1 \dots 1]$. Pour différentes valeurs de l'ordre de projection P de l'algorithme APA, on réalise des moyennes temporelles sur 50 échantillons successifs et des moyennes statistiques sur 1000 réalisations de l'erreur *a priori* $e(n)$, chacune de taille $n = 2000$. Afin de contrôler le risque de déstabilisation locale de l'algorithme

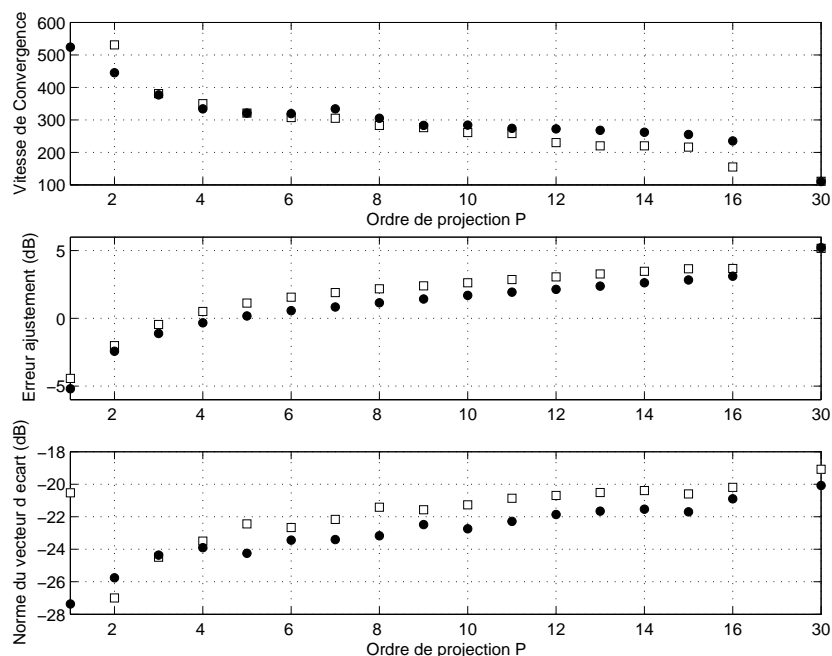


Figure 2.10. Comportement de l'algorithme APA en fonction de l'ordre P (courbes \bullet) pour un signal d'entrée peu corrélé ($\text{cond}(R) = 1.49$) : (haut) Vitesse de convergence, (milieu) Erreur d'ajustement normalisée, (bas) Norme du vecteur d'écart des coefficients à l'optimum

APA, une constante de régularisation directe $\delta = \sigma_b^2$ est ajoutée sur la diagonale de la matrice covariante des données $\Lambda_n^T \Lambda_n$ de l'algorithme APA.

Nous analysons tout d'abord le comportement des algorithmes lorsque le signal d'entrée est faiblement corrélé pour un modèle AR1 de pole $\rho = 0.1$ soit $\text{cond}(R) = 1.49$. La figure (2.10) représente, pour différentes valeurs de l'ordre de projection $P \in [1, 30]$ de l'algorithme APA, la *vitesse de convergence* T_c de l'algorithme (figure du haut) mesurée telle que $J_n > J_\infty \forall n < T_c$, l'*erreur d'ajustement normalisée* $(J_\infty - \sigma_b^2) / \sigma_b^2$ (figure du milieu) exprimée en décibels et la *norme de la différence des coefficients par rapport à l'optimum* évaluée asymptotiquement (figure du bas) soit $E[\|w_\infty - w^*\|^2]$ également exprimée en décibels. La vitesse de convergence permet d'apprécier le comportement en convergence initiale des algorithmes.

Nous comparons le cas de l'algorithme NLMS (équivalent à l'algorithme APA avec un ordre de projection $P = 1$) avec l'algorithme APA d'ordre $P \geq 2$. Nous constatons sur la figure (2.10) que, quelque soit l'ordre de projection $P \geq 2$, l'algorithme APA possède une vitesse de convergence initiale plus rapide que l'algorithme NLMS. Cette vitesse de convergence est d'autant plus rapide (comparativement au

NLMS) que les signaux d'entrée sont corrélés. En effet, la figure (2.12) obtenue pour un modèle AR1 de pole $\rho = 0.9$ soit $\text{cond}(R) = 256.57$ confirme cette tendance puisque le NLMS atteint le régime asymptotique au bout de 3000 itérations alors que l'algorithme APA l'atteint en moins de 500 itérations. Cette propriété est également illustrée sur la figure 2.11 qui représente la convergence initiale d'un des coefficients du filtre pour les algorithmes NLMS, APA d'ordre $P = 2$ et APA d'ordre $P = M = 30$ sur des signaux faiblement (gauche) et fortement (droite) corrélés.

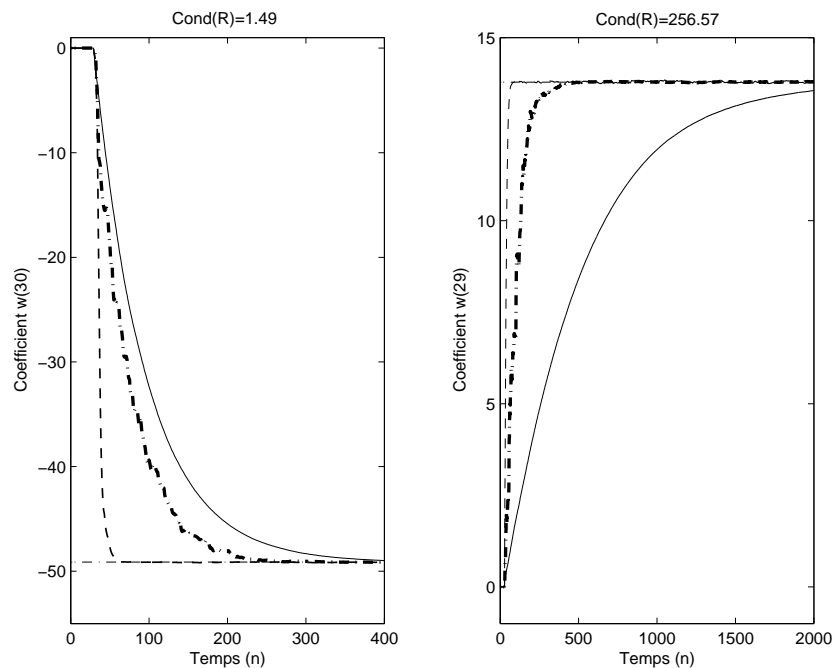


Figure 2.11. Convergence initiale des coefficients pour le NLMS (—), l'APA d'ordre $P = 2$ (-.-) et l'APA d'ordre $P = 30$ (- -) pour $\mu = 0.5$ et $\delta = \sigma_a^2$. Signal d'entrée faiblement corrélé $\text{cond}(R) = 1.49$ (gauche) et fortement corrélé $\text{cond}(R) = 257.57$ (droit)

Comparaisons en régime asymptotique

Afin de comparer les algorithmes NLMS et APA en régime asymptotique, nous commentons les figures (2.10) et (2.12) en terme d'erreur d'ajustement normalisée $(J_\infty - \sigma_b^2)/\sigma_b^2$ et de norme du vecteur d'écart des coefficients $E[\|w_\infty - w^*\|^2]$, ces grandeurs étant exprimées en décibels. On constate que l'algorithme NLMS possède un comportement asymptotique qui dépend de la nature du signal d'entrée puisque, pour une erreur d'ajustement normalisée de -5 dB, la norme du vecteur d'écart des

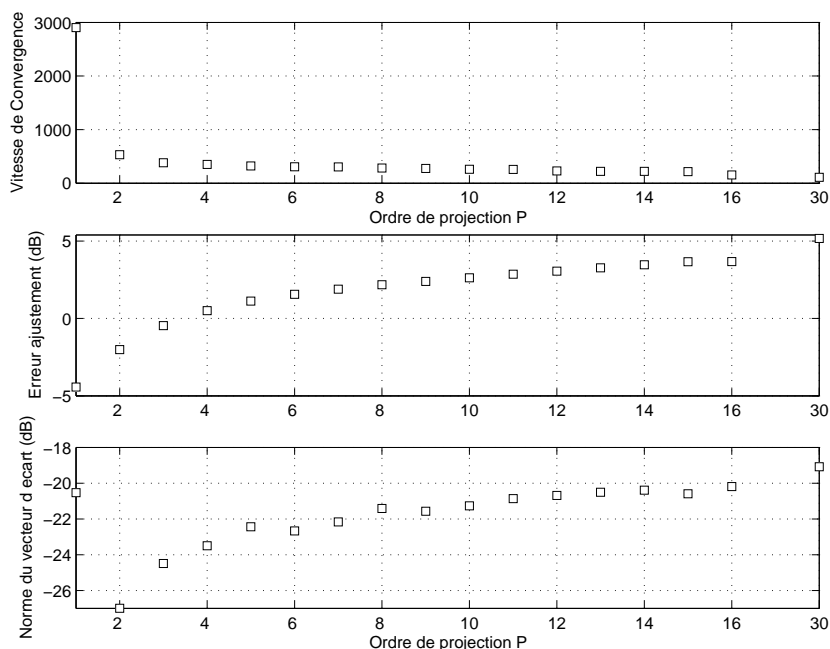


Figure 2.12. Comportement de l'algorithme APA en fonction de l'ordre P pour un signal d'entrée fortement corrélé ($\text{cond}(R) = 256.57$) : (haut) Vitesse de convergence, (milieu) Erreur d'ajustement normalisée, (bas) Norme du vecteur d'écart des coefficients à l'optimum

coefficients à l'optimum obtenue avec le NLMS est de -27 dB ou de -21.5 dB suivant que le signal d'entrée est fortement ou faiblement corrélé. Ceci est conforme à la relation $\frac{J_{\infty} - J_{min}}{J_{min}} = \frac{\mu}{2 - \mu} \text{Tr}\{R\} E\left[\frac{1}{r^2}\right]$ vue précédemment lors de l'analyse du NLMS. Par contre, l'algorithme APA présente une erreur d'ajustement normalisée et une norme du vecteur d'écart relativement indépendantes de la corrélation du signal d'entrée puisque l'on retrouve sensiblement les mêmes valeurs sur les courbes (2.10) et (2.12), soit lorsque le signal d'entrée est faiblement ou fortement corrélé. Ceci peut s'observer sur la figure (2.10) où sont superposés les résultats obtenus pour un signal d'entrée fortement (\square) et faiblement (\bullet) corrélé.

Il faut remarquer que les bonnes propriétés de vitesse de convergence initiale de l'algorithme APA ainsi que sa robustesse vis à vis de la coloration du signal d'entrée sont obtenues au prix d'une augmentation de l'erreur quadratique moyenne (ou EQM) asymptotique puisque l'EQM obtenue pour l'algorithme APA est toujours plus importante que celle obtenue pour le NLMS (on observe 10 dB de différence entre l'erreur d'ajustement normalisée obtenue pour le NLMS et l'APA d'ordre $P = 20$). En ce qui concerne la précision de la convergence asymptotique, nous constatons que

l'algorithme APA produit asymptotiquement une norme du vecteur d'écart des coefficients à l'optimum dont l'ordre de grandeur est sensiblement équivalent, que le signal d'entrée soit corrélé ou non.

2.5.2. Performances des algorithmes en treillis

L'étude de la convergence ne relève plus de la méthode directe qui avait servi pour le LMS standard. Elle relève par contre des méthodes générales d'étude des algorithmes stochastiques qui seront abordées dans le chapitre 3.

Qualitativement, on peut résumer les résultats de la façon suivante : les propriétés de convergence p.s. vues pour le LMS à pas décroissant sont conservées, mais la vitesse de convergence pourra être meilleure et surtout *moins dépendante du conditionnement de R*. Pour le détail de cette étude, on se reportera à [Michaut 92, Ch5].

LMS-Treillis : implémentation, choix des pas μ_i

En pratique, pour une implémentation adaptative, on choisira des pas constants $\mu_i(n) = \mu_i$. On a par contre un degré de liberté supplémentaire par rapport au LMS dans le choix d'un pas adapté à chaque étage du treillis.

Comme pour le LMS standard, la convergence n'aura pas lieu au sens strict dans ce cas : on peut seulement espérer une stabilisation de la fluctuation des $\Gamma_i(n)$ autour des Γ_i^* , et celle-ci supposera des pas μ_i assez petits. On peut avoir une idée qualitative des bornes nécessaires en étudiant la convergence du gradient déterministe (2.76), où nous choisissons des pas μ_i constants.

La version linéarisée autour de l'optimum Γ^* donne pour l'écart $\delta_i(n) = \Gamma_i(n) - \Gamma_i^*$ la récurrence homogène

$$\delta_i(n) = (1 - 2\mu_i D_{i-1})\delta_i(n-1) \quad (2.86)$$

et donc la condition de stabilité locale à l'optimum sera

$$|1 - 2\mu_i D_{i-1}| < 1 \quad \iff \quad 0 < \mu_i < \frac{1}{D_{i-1}} \quad (2.87)$$

Une bonne adaptation des pas μ_i peut permettre d'atteindre une vitesse de convergence plus grande que le LMS, et on peut s'attendre à une moindre dépendance de la vitesse par rapport au conditionnement (taux de corrélation des données) en liaison avec la décorrélation et l'orthogonalisation des étages successifs qui revient à diagonaliser la covariance.

Vitesse comparée LMS et LMS-Treillis

Le modèle du signal est un AR2. On établit que, pour les algorithmes à pas constant *unique*, le choix du pas optimal donne un taux de convergence de LMS-Treillis toujours meilleur que LMS. On accentue la supériorité du LMS-Treillis en choisissant des pas μ_i adaptés à chaque étage du treillis,

La figure 2.13 montre l'évolution des deux algorithmes, sur une réalisation, dans le cas des petits pas du LMS $\mu = 0.01$, et du LMS-Treillis de $\mu_1 = 0.02$, $\mu_2 = 0.04$ en rapport 1/50 avec les bornes déterministes $1/\lambda_{max}$, $1/D_0$ et $1/D_1$. On voit une vitesse nettement accrue du LMS-treillis en transitoire, mais au prix d'une fluctuation résiduelle sensiblement plus forte que pour le LMS.

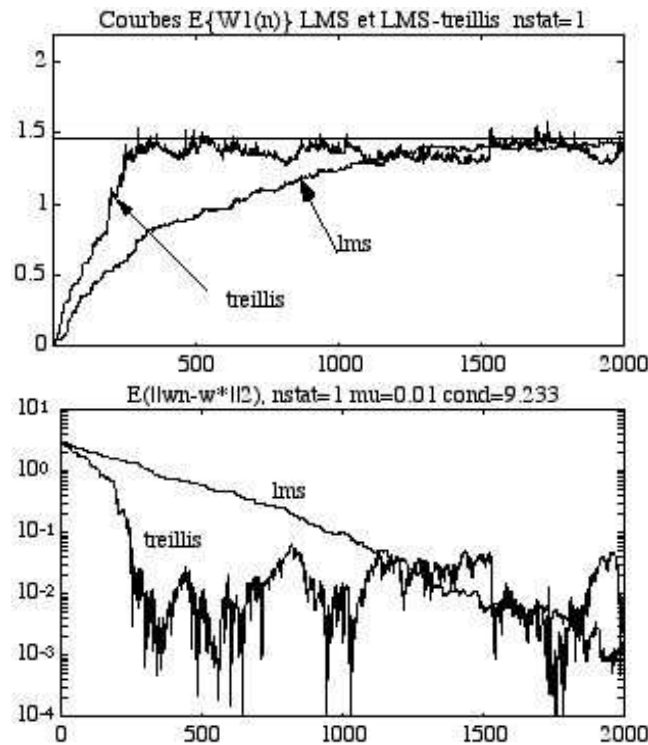


Figure 2.13. Comparaison LMS et LMS-Treillis

Vitesse comparée du Moindres carrés en treillis (Burg-adaptatif)

La figure 2.14 montre la simulation des trois algorithmes Burg, RLS, LMS-Treillis à pas décroissant $\mu_n = 1/n$. Le modèle est un AR2 avec des pôles définis par $\rho = 0.9$ et $\nu = 0.01$, de très mauvais conditionnement $\text{cond}(R) = 266$.

On constate la supériorité absolue de Burg en vitesse et qualité de convergence sur le RLS, même si les deux algorithmes ont un comportement proche avec une vitesse de convergence indépendante du mauvais conditionnement.

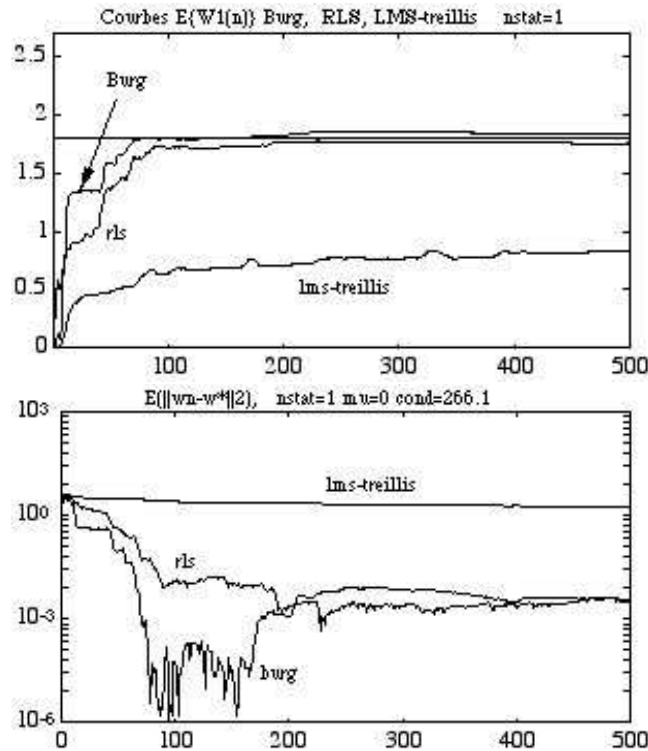


Figure 2.14. Comparaison de RLS, Burg et LMS-Trellis

Au contraire le LMS-trellis montre clairement les limitations des algorithmes de gradient face au conditionnement : notons toutefois que nous avons pris ici *le même pas* μ_n pour les deux étages du treillis. On pourrait donc améliorer la performance du LMS-Trellis (sans toutefois se rapprocher de la vitesse des Moindres carrés) en adaptant les pas à chaque étage du treillis.

Conclusions sur le treillis adaptatif

La représentation en treillis apporte de nombreux avantages, principalement la décorrélation des étages successifs et la modularité de mise en œuvre : on peut ajouter un étage au treillis *sans remettre en cause les paramètres* k_i déjà optimisés dans les étages précédents. On peut dire d'un point de vue analyse numérique qu'elle remplace le traitement en bloc d'une optimisation M-dimensionnelle par une suite de problèmes

d'optimisation *unidimensionnels*, et ceci grâce à l'orthogonalisation de l'espace d'observation.

La conséquence du point de vue de la vitesse de convergence des algorithmes est la suivante :

- le LMS-Treillis associé à un choix adapté des pas de calcul par étage est intermédiaire entre le LMS et le RLS,
- il y a donc dans le cas du treillis moins de différence entre l'approche gradient et l'approche Newton-Moindres carrés,
- chacun des algorithmes en treillis surpasse en vitesse et en qualité de convergence l'algorithme correspondant de la structure transversale de la prédiction linéaire,
- la complexité de calcul de l'approche Moindres carrés du Burg-récurif est *du même ordre de grandeur que pour le LMS* en $O(M)$, par opposition à la complexité en $O(M^2)$ du RLS. Ce gain fondamental est lié directement à la *diagonalisation* du problème d'optimisation.

Ces bonnes propriétés de la structure en treillis expliquent pourquoi elle sera aussi à la base de la construction des algorithmes rapides de Moindres carrés, étudiés au chapitre 4.

Notes

L'étude théorique de la convergence du LMS à pas décroissant a été faite par [Farden 81] et [Eweda-Macchi 83 84, 87]. Pour l'étude du comportement asymptotique du LMS à pas constant (en situation stationnaire et en poursuite), une méthodologie spécifique a été développée dans [Macchi 81, 87]. L'intérêt de cette approche est d'obtenir des résultats de stabilité inconditionnelle, ou des estimations d'erreur quadratique non asymptotiques (valables pour tout $\mu < \mu_0$), mais elle ne se généralise pas à d'autres algorithmes adaptatifs.

Les résultats les plus complets pour le LMS à pas fixe, en stationnaire et en poursuite de filtre variable, sont ceux de [Solo 89], [Solo 92]. Ils sont de nature asymptotique, et basés sur des méthodes de moyennage (déterministe ou stochastique) qui ont ensuite été généralisées à tous les algorithmes stochastiques dans l'ouvrage de référence [Solo 95] (cf Chapitre 3).

Les outils théoriques sur les propriétés d'excitation persistante et la stabilité exponentielle du LMS sont développés dans [Green, Moore 86] et [Bitmead 81,84], [Bitmead, Anderson 80], [Bitmead 80 à 86].

Pour des résultats statistiques classiques sur les Moindres carrés, on se reportera à [Monfort 80]. Des résultats approfondis dans des contextes généraux d'estimation

réursive de processus aléatoires sont développés dans [Duflo 90]. L'étude du RLS à facteur d'oubli a été menée d'abord par [Macchi-Eweda, 83 et 88], et approfondie y compris en situation non stationnaires par [Ljung, Priouret 91, 93].

2.6. Bibliographie

- BENESTY et DUHAMEL, 1992, "A fast exact least mean square adaptive algorithm", IEEE Trans. on Signal Processing, vol. 40, no 12, décembre 1992, p 2904.
- BERSHAD, 1986, "Analysis of the normalized LMS with gaussian inputs", IEEE Trans. ASSP, vol ASSP-34, août 1986, p 793.
- BITMEAD, 1981, "Convergence properties of LMS adaptive estimators with unbounded dependent inputs.", Proc. 20ème IEEE Conf. Decision Contr., San Diego, CA, 1981, p607
- BITMEAD, 1983, "Convergence in distribution of LMS-Type adaptive parameter Estimates", IEEE Trans. AC-28, n°1.
- BITMEAD, 1984, "Convergence properties of LMS adaptive estimators with unbounded dependent inputs", IEEE Trans. AC-29, n°5 p 477.
- BITMEAD, 1986, "Convergence rate determination for gradient-type algorithms", Automatica vol. 22, 1986, p185.
- BITMEAD et ANDERSON, 1980-a, "Lyapunov techniques for exponential stability of linear difference equations with random coefficients.", IEEE Trans. AC25, p782, aout 80
- BITMEAD et ANDERSON, 1980-b, "Performances of adaptive estimation algorithms in dependant random environments", IEEE Trans. AC25, p788, Août1980.
- CLARK et al., 1981, "Block implementation of adaptive digital filters", IEEE Trans. Circuits Syst., vol. CAS-28, juin 1981, p 584.
- DE ALMEIDA et al., 2003, "A stochastic model for the convergence behavior of the affine projection algorithm for gaussian inputs", Int. Conf Acoustics, Speech, and Signal Processing, 2003, p313.
- DUFLO, 1990, Méthodes récurives aléatoires, Masson, Paris
- EWEDA et MACCHI, 1984, "Convergence of an adaptive linear estimation algorithm", IEEE Trans. AC-29 n°2.
- EWEDA et MACCHI, 1987, "Convergence of the RLS and LMS adaptive filters", IEEE Trans. Circ.Syst. vol .34 n°7.
- FARDEN, 1981, "Stochastic approximation with correlated data", IEEE Trans. IT-27 n°1.
- FURUKAWA et al., 1989, "The orthogonal projection algorithm for block adaptive signal processing", Int. Conf. on Acoustics, Speech, and Signal Processing, vol.2, pp. 1059.
- GAY, 2003, "A fast converging, low complexity adaptive filtering algorithm", 3rd Int. Workshop on Acoustic Echo Control, Plestin-les-grèves, France, pp. 223.
- GAY et TAVATHIA, 1995, "The fast affine projection algorithm", Int. Conf. on Acoustics, Speech, and Signal Processing, 1995, p. 3023.

- GREEN et MOORE, 1986, "*Persistence of excitation in linear systems*", Systems and Control Letters vol7 n°5 Sept 86
- GUO, LJUNG et PRIOURET, 1993, "*Performance analysis of the forgetting factor RLS algorithm*", International Journal of Adaptive Control and Signal Processing, vol.7, p525.
- LJUNG et PRIOURET, 1991-a, "*Remarks on the Mean-square Tracking Error*", International Journal of Adaptive Control and Signal Processing vol. 5, p 231.
- LJUNG et PRIOURET, 1991-b, "*A result on the Mean-square Error obtained using general tracking algorithms*", International Journal of Adaptive Control and Signal Processing vol. 5, p 395.
- LJUNG et CAINES, 1979, "*Asymptotic normality of Prediction error estimation for approximate system Models*", Stochastics vol. 3, p29. MACCHI et EWEDA, 1983, "*Second order convergence analysis of stochastic adaptive linear filtering*", IEEE Trans. AC-28 n°1, p76.
- MACCHI et EWEDA, 1988, "*Compared speed and accuracy of the RLS and LMS algorithms with constant forgetting factor*", RAIRO APII vol22,N°3
- MACCHI et BELLANGER, 1987, "*Le point sur le filtrage adaptatif transverse*", Colloque GRETSI, Nice 1987.
- MONFORT, 1980, Cours de statistique mathématique, Economica, Paris.
- OZEKI et UMEDA, 1984, "*An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties*", Electronics and Communications in Japan, vol. J67-A , p. 19.
- RAGHOTHAMAN et al., 2000, "*A new method for low-rank transform domain adaptive filtering*", IEEE Trans. on SP, vol. 48, n°4, p1097.
- ROBBINS et MONRO, 1951, "*A stochastic approximation method*", Ann. Math.Stat. vol. 22, p400.
- ROMBOUITS et MOONEN, 2002, "*A sparse block exact affine projection algorithm*", IEEE Trans. on SP, vol. 10, février 2002, p. 100.
- SANKARAN et BEEB, 2000, "*Convergence behavior of affine projection algorithms*", IEEE Transactions on Signal Processing, vol. 48, Avril 2000, p. 1086.
- SHIN et SAYED, 2004, "*Mean-square performance of a family of affine projection algorithms*", IEEE Trans. on SP, vol. 52, Janvier 2004, p. 90.
- SLOCK, 1993, "*On the convergence behavior of the LMS and the normalized LMS algorithms*", IEEE Trans. on SP, vol. 41, n°9, September 1993, pp. 2811.
- SOLO, 1989, "*The limiting behavior of LMS*", IEEE Trans. ASSP-37, n°12, p1909.
- SOLO, 1992, "*The Error Variance of LMS with Time-Varying Weights*", IEEE Trans. SP vol40, Avril 92, p803.
- SOLO et KONG, 1995, Adaptive Signal Processing Algorithms, Stability and Performance, Prentice Hall
- SOO et PANG, 1990, "*Multidelay block frequency domain adaptive filter*", IEEE Trans. ASSP, vol. ASSP-38, février 1990, p 373.
- TANAKA et al., 1995, "*Fast projection algorithm and its step size control*", Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 2, mai 1995, p. 945.

- TANAKA et al., 1999, "*A block exact fast affine projection algorithm*", IEEE Trans. on SAP , vol. 7, n°1, Janvier 1999, p.79.
- TARRAB et FEUER, 1988, "*Convergence and Performance Analysis of The Normalized LMS Algorithm with Uncorrelated Gaussian Data*", IEEE Trans. Inform. Theory, vol. 34, No.4, juillet 1988, p 680.
- WIDROW et al., 1976, "*Stationary and non-stationary learning characteristics of the LMS adaptive filter*", Proc. IEEE vol. 64 n°8, 1151.

Chapitre 3

Algorithmes adaptatifs : méthodes générales

Ce Chapitre présente quelques techniques pour l'étude théorique d'un algorithme adaptatif générique : la question de la convergence des algorithmes a été abordée par des méthodes spécifiques pour le LMS et le RLS du cas standard (Chapitre 2). Ces méthodes étaient liées étroitement au caractère quadratique de la fonction de coût $J(\theta)$. Le but est ici d'aborder, par une méthode générale, l'étude de la convergence asymptotique d'algorithmes d'approximations stochastiques plus généraux : entreront dans ce champ les structures de filtrage étendues (treillis, filtres récursifs abordés au Chapitre 5), les modélisations étendues ARMA, ARMAX, Kalman, les estimations de retards et de sous espaces, les méthodes d'adaptation sans référence (Chapitre 6) et les méthodes d'apprentissage non-linéaire (Chapitre 11).

L'étude du comportement de ces algorithmes se décompose en trois phases de complexité croissante : étude de la convergence initiale, étude des fluctuations en régime stationnaire et étude de la capacité de poursuite de non-stationnarités lentes.

Tout au long de ce chapitre, les outils introduits sont illustrés par trois exemples :

- l'exemple très élémentaire de l'estimation de la moyenne d'une suite de v.a. indépendantes et équidistribuées (i.i.d.),
- un exemple relevant des algorithmes étendus : identification RII (1-0),
- un exemple relevant d'algorithmes d'approximation stochastique plus général : le neurone d'Oja.

Chapitre rédigé par Jean-Pierre DELMAS, Jean-Marc BROSSIER et François MICHAUT.

3.1. Nécessité de traitements adaptatifs plus généraux

3.1.1. Filtrage RII, modélisation ARMA

Prenons comme exemple l'identification d'un système linéaire (correspondant par exemple au problème de l'annulation d'écho, cf. Chapitre 8). Le schéma en a été donné au Chapitre 1, Fig-1.3. Supposons que le système inconnu soit représenté par un filtre récursif RII de transfert $H_*(z) = \frac{B_*(z)}{A_*(z)}$ avec $B_*(z) = \sum_0^q \beta_k z^{-k}$ et $A_*(z) = 1 - \sum_1^p \alpha_k z^{-k}$ paramétré par $\theta_* = (\alpha_1, \dots, \alpha_p, \beta_0, \dots, \beta_q)^T$. Le filtre est supposé causalement stable, les zéros de $A_*(z)$ sont à l'intérieur du cercle unité.

L'identification standard par un filtre de Wiener RIF ne donnera qu'une approximation du système, et nécessitera éventuellement un filtre très long : il est alors intéressant de chercher à modéliser directement le système par un filtre RII, du type $H(z) = \frac{B(z)}{A(z)}$, paramétré par le vecteur $\theta = (a_1, \dots, a_p, b_0, \dots, b_q)^T$ des coefficients du filtre. Il est clair que l'EQM de sortie $J(\theta) = E(e_n^2) = \sigma_b^2 + \text{var}[(H - H_*)u_n] \geq \sigma_b^2$ possède un minimum global atteint pour $H = H_*$ donc $\theta = \theta_*$. Ainsi l'identification se ramène encore à la minimisation de cette fonction de coût d'EQM. La différence par rapport au problème de base traité au Chapitre 2 réside dans le caractère non-quadratique en θ de cette fonction. Donnons maintenant quelques exemples qui illustrent ce point :

Exemple correctement paramétré. Pour un filtre (1-0) $H_*(z) = \frac{\beta}{1 - \alpha z^{-1}}$ de modèle $\frac{b}{1 - az^{-1}}$ on obtient

$$J(\theta) = \sigma_b^2 + \sigma_u^2 \left(\frac{\beta^2}{1 - \alpha^2} + \frac{b^2}{1 - a^2} - \frac{2b\beta}{1 - a\alpha} \right)$$

et l'on établit facilement sur ce cas que le minimum global est atteint pour $\theta = \theta_*$, qui est l'unique point stationnaire de $J(\theta)$.

Exemple sous-paramétré. L'un des problèmes posés par cette modélisation récursive est la possibilité d'apparition de minima locaux parasites dans la fonction de coût. C'est le cas notamment en situation de "sous-paramétrage", c'est-à-dire lorsque les ordres (p, q) choisis pour le modèle adaptatif θ sont inférieurs aux ordres (p_0, q_0) du vrai système. On le vérifie sur l'exemple suivant [Shynk 89] pour

$$H_*(z) = \frac{\beta_0 + \beta_1 z^{-1}}{1 - \alpha_1 z^{-1} - \alpha_2 z^{-2}} \quad \text{de modèle : } H(z) = \frac{b}{1 - az^{-1}},$$

on a, dans le cas d'une entrée u_n blanche avec $\sigma_u^2 = 1$ sans bruit de sortie, $J(\theta) = \text{Cste} + \frac{b^2}{1 - a^2} - 2bH_*(a^{-1})$, et on obtient par exemple pour $(\beta_0, \beta_1, \alpha_1, \alpha_2) = (0.05, -0.4, 1, -0.49)$, trois minima locaux :

a	b	$J(a, b)$ -Cste
-0.503302	0.11531	-0.0178071
0.125	0	0
0.779282	-0.198307	-0.100137

L'étude de ces minima locaux parasites est donc un élément important de caractérisation des points de convergence possibles pour les algorithmes adaptatifs associés. Dans cet exemple de modélisation RII, des conditions suffisantes d'unimodalité de la fonction $J(\theta)$ ont été établies dans [Söderström 75], [Åström 74] : si l'entrée u_n est un bruit blanc et l'ordre du modèle est suffisant ($p \geq p_0, q \geq q_0$), tout point stationnaire de $J(\theta)$ est un minimum global et vérifie la condition d'identification correcte du filtre $A_\theta(z)B_*(z) = A_*(z)B_\theta(z)$. L'étude détaillée du filtrage adaptatif RII sera faite au Chapitre 5.

Exemple de la modélisation ARMA. La modélisation ARMA est le problème dual de la modélisation RII, et conduit aux mêmes types de fonctions de coût fortement non quadratiques, et de problèmes de minima locaux : le signal ARMA analysé est $x_n = \frac{B_*(z)}{A_*(z)}\epsilon_n$, généré par filtrage stable et à phase minimale du bruit blanc d'innovation ϵ_n . L'identification du modèle se fait en minimisant l'EQM $J(\theta) = E(e_n^2)$ à partir de l'erreur de prédiction $e_n = \frac{A(z)}{B(z)}x_n$, et on vérifie facilement qu'en situation de "bonne modélisation" (ordre du modèle au moins égal à l'ordre du filtre générateur), tout point stationnaire est un minimum global et donne le modèle ARMA correct. Ces deux approches du filtrage adaptatif RII seront développées au Chapitre 5.

Exemple utilisé pour la suite du chapitre. Dans la suite, nous illustrerons les méthodes sur le problème type de l'identification RII (1-0) : les filtres sont paramétrés par $\theta = (a, b)^T$

$$H_*(z) = \frac{B_*(z)}{A_*(z)} = \frac{\beta}{1 - \alpha z^{-1}} \quad H_\theta(z) = \frac{B(z)}{A(z)} = \frac{b}{1 - az^{-1}} \quad (3.1)$$

le domaine de stabilité des filtres est $S = \{(a, b), |a| < 1, b \in \mathbb{R}\}$, et les signaux sont :

$$x_n = H_*(z)u_n + b_n \quad y_n(\theta) = H_\theta(z)u_n \quad e_n(\theta) = x_n - y_n(\theta) \quad (3.2)$$

avec le signal d'entrée u_n à filtrer, le bruit de sortie b_n indépendant de l'entrée, et l'erreur d'estimation $e_n(\theta)$.

3.1.2. Egalisation linéaire aveugle

On considère l'égalisation du canal $S(z)$ défini par le schéma de filtrage [Fig. 3.1] (voir [Benveniste 80]).

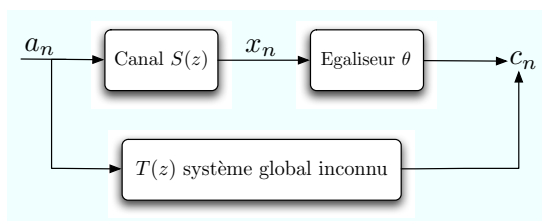


Figure 3.1. Egalisation aveugle

Le problème posé est ici de retrouver l'entrée a_n ou d'identifier le filtre inverse $S^{-1}(z)$ à partir de l'observation complète de la sortie x_n et de la connaissance de la loi de probabilité de l'entrée a_n suite i.i.d. Si le filtre $S(z)$ est à *phase minimale*, on se ramène à un problème de modélisation ARMA, qui se traite par des méthodes d'erreur de prédiction minimale, conduisant à la situation vue à la section précédente. Par contre, si $S(z)$ n'est pas à phase minimale, son inverse n'est pas causalement stable et ne peut donc être identifié de cette façon : il faudra utiliser d'autres moments que l'ordre deux (et en particulier on ne pourra pas faire l'identification lorsque la loi de l'entrée est gaussienne). On utilisera le théorème suivant :

Théorème : Soit θ un égaliseur linéaire tel que la distribution de probabilité de c_n coïncide avec celle de a_n non-gaussienne. Alors $T = \pm I$ à un retard près.

L'égaliseur linéaire est obtenu en minimisant une fonction de coût de la forme $J(\theta) = \mathbb{E}[\Psi(c)]$ où Ψ est une fonction paire autre que $(\cdot)^2$ (mettant en jeu la loi de a_n au-delà de son ordre 2), permettant de mesurer la distorsion de loi imposée par le filtre T , et possédant un minimum unique pour $T = \pm I$ à un retard près.

EXEMPLE 1 : ALGORITHME DE SATO. La distorsion de loi peut être mesurée ici par le critère $\Psi(c) = (c - \gamma \text{Sign}(c))^2$ où $\gamma = \frac{\mathbb{E}[|a_n|^2]}{\mathbb{E}[|a_n|]}$. On montre alors que $T = \pm I$ (à un retard près) est l'unique minimum local (donc global) de J lorsque la loi ν est *sous-gaussienne* ([Benveniste 80]). Bien que ce résultat ne soit strictement valide que pour un égaliseur de taille infinie (dont la structure est capable de représenter l'inverse du canal), il reste néanmoins possible de construire un système à réponse impulsionnelle finie pratiquement utilisable pour peu que la taille de l'égaliseur soit suffisante. Si l'on paramètre l'égaliseur sous forme transversale (non-causale, centrée sur 0 en R.I. tenant compte du fait que l'inverse stable du filtre $S(z)$ à phase non-minimale n'est pas causal) :

$$\begin{aligned} \text{Etat } \varphi_n &= (x_{n-N}, \dots, x_n, \dots, x_{n+N})^T \\ \text{Egaliseur } \theta &= (\theta_{-N}, \dots, \theta_N)^T \quad c_n = \theta^T \varphi_n \end{aligned}$$

l'algorithme de gradient stochastique pour minimiser $J(\theta)$ donne alors l'*algorithme de Sato* (ou algorithme du signe) :

Gradient stochastique (Sato) :

$$\theta_{n+1} = \theta_n - \mu(c_n - \gamma \text{sign}(c_n))\varphi_n \quad (3.3)$$

qui converge vers un filtre $T = \pm I$ (à un retard près), et donne en sortie $c_n = \pm a_{n-n_0}$. Le Chapitre 6 (Volume 2) développera les méthodes d'égalisation aveugle (sans référence).

3.1.3. Estimation de retards

Nous citons ici un autre exemple dans lequel un algorithme adaptatif est issu de la minimisation d'une fonction de coût non quadratique. Considérons deux signaux y_n et x_n centrés stationnaires du second ordre retardés l'un par rapport à l'autre d'un retard inconnu θ_* (supposé entier) et bruités où le signal de référence x_n peut être atténué d'un facteur α inconnu ($\alpha > 0$) :

$$y_n = s_{n-\theta_*} + b_{1,n} \quad \text{et} \quad x_n = \alpha s_n + b_{2,n}.$$

Nous supposons que les signaux s_n , $b_{1,n}$ et $b_{2,n}$ sont mutuellement non corrélés. L'estimation du retard θ_* peut être réalisée par minimisation adaptative de la fonction $J(\theta) = E(y_n - x_{n-\theta})^2$. En effet puisque $J(\theta) = (1 + \alpha^2)r_s(0) - 2\alpha r_s(\theta_* - \theta) + r_{b_1}(0) + r_{b_2}(0)$, la minimisation de $J(\theta)$ est équivalente à la maximisation de l'autocorrélation $r_s(\theta_* - \theta)$ de s_n , qui a bien lieu pour $\theta = \theta_*$ et naturellement cette autocorrélation n'est pas une fonction quadratique de θ .

3.1.4. Approximation de sous-espaces, réseau de neurones

Etant donnée une matrice de covariance ¹ $R_x = E(xx^T)$ d'un vecteur aléatoire x centré, appelons $\{\lambda_1 \geq \dots \geq \lambda_m\}$ les valeurs propres de R_x et (v_1, \dots, v_m) une base orthonormée de vecteurs propres associés. Nous nous intéressons à deux problèmes distincts :

- (1) **L'estimation adaptative** d'une base orthonormée de r vecteurs propres associés aux r plus grandes [resp. plus petites] valeurs propres distinctes $(\lambda_1, \dots, \lambda_r)$ [resp. $\lambda_{m-r+1}, \dots, \lambda_m$] de R_x .
- (2) **L'estimation adaptative d'un sous espace invariant majorant ou minorant**, c'est à dire d'un sous espace engendré par les vecteurs propres de R_x associé aux r plus grandes [resp. plus petites] valeurs propres où nous supposons seulement $\lambda_r > \lambda_{r+1}$ [resp. $\lambda_{m-r} > \lambda_{m-r+1}$]. Dans ce dernier cas nous pourrions

1. Ces méthodes s'étendent aux vecteurs aléatoires complexes circulaires du deuxième ordre avec $R_x = E(xx^H)$.

estimer de façon adaptative, soit une base orthonormée ou approximativement orthonormée d'un tel sous espace, soit une matrice de projection associée, orthogonale ou approximativement orthogonale.

A partir d'une suite x_n de réalisations de la v.a. x , il existe une abondante littérature permettant de répondre aux deux problèmes précédents. Parmi celle-ci, des *algorithmes d'approximations stochastiques* issus de *méthodes d'analyse numérique* ou d'*optimisation de critères sous ou sans contraintes*. Comme leurs noms l'indiquent, ces algorithmes sont conçus au départ comme des approximations et quelquefois assez empiriques de méthodes mathématiques standards. Pour être retenu, l'algorithme ainsi obtenu devra faire l'objet d'études de convergence comme nous le verrons à la section (3.2). Ces études détermineront en particulier si nous avons affaire à un algorithme de la famille (1) ou (2). Nous verrons ainsi que ces méthodes (développées à la section (3.5)) se distinguent de celles des exemples précédents par le fait que bien que s'écrivant sous la forme générale (3.8), elles ne sont pas toujours interprétables comme issues de la minimisation d'une fonction de coût et l'ensemble des points stationnaires associés à cette équation (3.8) peut être un *continuum* de points. A titre d'exemple, considérons le cas le plus simple ($r = 1$ pour lequel les deux problèmes (1) et (2) sont confondus) en examinant la construction empirique du célèbre neurone d'Oja [Oja 82]

$$w_n = w_{n-1} + \mu_n [I_m - w_{n-1} w_{n-1}^T] x_n x_n^T w_{n-1}, \quad (3.4)$$

qui fournit une estimée adaptative w_n d'un vecteur propre normalisé v_1 associé à la plus grande valeur propre λ_1 (supposée unique) de la matrice de covariance $R_x = E(x_n x_n^T)$ du processus stationnaire centré $x_n \in \mathbb{R}^m$:

A partir de l'algorithme d'analyse numérique de la puissance itérée

$$\begin{aligned} w'_n &= R_x w_{n-1} \quad \text{avec} \quad w_0 \text{ quelconque non orthogonal à } v_1 \text{ et } \|w_0\| = 1 \\ w_n &= \frac{w'_n}{\|w'_n\|} \end{aligned}$$

dont la vitesse de convergence peut être améliorée en remplaçant R_x par $I_m + \mu R_x$ (avec $0 < \mu \ll 1$), le neurone d'Oja est obtenu grâce aux approximations suivantes (en tenant compte que $\|w_{n-1}\| = 1$) :

$$\begin{aligned} \frac{w'_n}{\|w'_n\|} &= \frac{(I_m + \mu R_x) w_{n-1}}{\sqrt{w_{n-1}^T (I_m + \mu R_x)^2 w_{n-1}}} \approx \frac{(I_m + \mu R_x) w_{n-1}}{\sqrt{1 + 2\mu w_{n-1}^T R_x w_{n-1}}} \\ &\approx (w_{n-1} + \mu R_x w_{n-1}) (1 - \mu w_{n-1}^T R_x w_{n-1}) \\ &\approx w_{n-1} + \mu [I_m - w_{n-1} w_{n-1}^T] R_x w_{n-1}, \end{aligned}$$

et en remplaçant R_x par son estimée instantanée $x_n x_n^T$ et μ par la suite de pas μ_n . Il est à noter comme nous le verrons à la section (3.5), que cet algorithme d'approximation stochastique est issu de nombreuses approches empiriques différentes.

Si de plus, nous nous intéressons à l'estimation adaptative α_n de la valeur propre associée λ_1 , nous pouvons considérer la minimisation de la fonctionnelle $J(\alpha) \equiv (\alpha - v_1^T R_x v_1)^2$. En utilisant la version stochastique associée à l'algorithme du gradient et en y injectant l'estimée w_{n-1} de v_1 fournie par l'algorithme (3.4), l'algorithme adaptatif suivant peut compléter l'algorithme (3.4) :

$$\alpha_n = \alpha_{n-1} + \mu_n (w_{n-1}^T x_n x_n^T w_{n-1} - \alpha_{n-1}). \quad (3.5)$$

3.1.5. Conception des algorithmes

La démarche de conception des algorithmes a été présentée au Chapitre 1 dans le cas d'une problématique d'EQM minimale. Appliquons celle-ci à la modélisation RII. La fonction à minimiser est donc ici l'EQM $J(\theta)$. Il reste à en calculer le gradient, puis à appliquer le principe d'approximation stochastique, pour obtenir un algorithme de gradient stochastique généralisant l'algorithme LMS du Chapitre 2.

Calcul du gradient

L'erreur d'estimation s'écrit $e_n(\theta) = x_n - y_n(\theta)$ avec $y_n(\theta) = \theta^T \varphi_n(\theta)$ à l'aide du vecteur d'état $\varphi_n(\theta) = (y_{n-1}, \dots, y_{n-p}, u_n, \dots, u_{n-q})^T$. Notons que l'état est ici généré par un modèle markovien linéaire

$$\varphi_n(\theta) = A(\theta)\varphi_{n-1}(\theta) + B(\theta)U_{n-1} \quad (3.6)$$

avec des matrices $A(\theta), B(\theta)$ fonctions respectivement des parties a et b de θ .

On a donc pour le gradient $\nabla e_n = -\nabla y_n$ et par dérivation

$$\nabla y_n = \varphi_n(\theta) + \theta^T \nabla \varphi_n(\theta) = \varphi_n(\theta) + \sum_{k=1}^p a_k \nabla y_{n-k}$$

ce qui permet d'assimiler le gradient $\nabla e_n = -\psi_n$, processus aléatoire stationnaire, à la solution stationnaire de l'équation récurrente ci-dessus, interprétée en termes de filtrage récursif sous la forme

$$\nabla J(\theta) = -2E(e_n \psi_n) \quad \text{avec l'état filtré } \psi_n = \frac{1}{A(z)} \varphi_n(\theta)$$

On aboutit ainsi à l'algorithme de Gradient déterministe RII, pour minimiser $J(\theta)$

$$\theta_n = \theta_{n-1} + \mu_n E(e_n \psi_n) \quad \text{GD-RII}$$

Gradient stochastique LMS-RII

Le passage à un algorithme adaptatif se fait

– par approximation stochastique du gradient

– par approximation de l'état gelé (à θ constant) (3.6) par un état à mémoire finie, obtenu en mémorisant les valeurs antérieures et en ne recalculant que la nouvelle sortie y_n . On fait de même pour l'état filtré ψ_n .

On obtient l'algorithme appelé ici LMS-RII

LMS-RII

$$\begin{aligned}\theta_n &= \theta_{n-1} + \mu_n e_n \psi_n \quad \text{avec } e_n = x_n - y_n \text{ et } y_n = \varphi_n^T \theta_{n-1} \\ \varphi_n &= (y_{n-1}, \dots, y_{n-p}, u_n, \dots, u_{n-q})^T \\ \psi_n &= (s_{n-1}, \dots, s_{n-p}, t_n, \dots, t_{n-q})^T \\ s_n &= y_n + \sum_{k=1}^p a_k(n-1) s_{n-k} \text{ et } t_n = u_n + \sum_{k=1}^p a_k(n-1) t_{n-k}\end{aligned}$$

Un des problèmes posés par cet algorithme stochastique est le risque d'explosion (instabilité de l'état) lié au filtrage récursif des données par le filtre aléatoire $\frac{1}{A_\theta(z)}$ pour générer l'état et l'état filtré.

Méthodes quasi-newtoniennes stochastiques

L'algorithme Quasi-Newton sera obtenu en calculant le Hessien $\nabla^2 J(\theta_*)$ et en approchant son inverse : on montre ici (voir [Benveniste 81]) que

$$\frac{1}{2} \nabla^2 J(\theta_*) = E(\psi_n(\theta_*) \psi_n(\theta_*)^T) \quad (3.7)$$

on pose alors $\Sigma(\theta) = E(\psi_n(\theta) \psi_n(\theta)^T)$ (matrice de covariance de l'état filtré), et on l'approche par son estimateur empirique $R_n = \frac{1}{n} \sum_1^n \psi_k \psi_k^T$ mis sous forme récursive : cela donne l'algorithme RML (*Recursive Maximum Likelihood*), extension du RLS à ce filtrage récursif

RML

$$\begin{aligned}\theta_n &= \theta_{n-1} + \mu_n R_n^{-1} e_n \psi_n \quad \mu_n = \frac{1}{n} \\ R_n &= R_{n-1} + \mu_n (\psi_n \psi_n^T - R_{n-1})\end{aligned}$$

Afin de limiter les risques d'instabilité liés au calcul de l'état filtré $\psi_n = \frac{1}{A_\theta(z)} \varphi_n$, des variantes sont souvent utilisées, dont la plus simple consiste à ne pas filtrer : on remplace ψ_n par φ_n , ce qui revient à appliquer directement l'algorithme RLS en ignorant la dépendance en θ de l'état. On parle alors de méthode de pseudo-gradient, et on a l'algorithme ELS (*Extended Least Squares*).

3.2. Forme générale d'algorithmes, méthodes pour la convergence (ODE)

CONTENU.— Nous abordons ici la forme générale des algorithmes adaptatifs et les méthodes générales d'étude de convergence dans un contexte stationnaire. Celles-ci sont toutes basées sur le moyennage du champ de vecteur de l'algorithme, conduisant à un algorithme déterministe moyen associé à l'algorithme adaptatif (stochastique). Il s'agit ensuite de comparer les trajectoires de ces deux algorithmes. On distingue les méthodes discrètes (raisonnant directement sur l'algorithme moyen : système dynamique discret déterministe ou équation récurrente non-linéaire), et les méthodes continues considérant l'algorithme déterministe comme une discrétisation (à pas $\mu_n \rightarrow 0$) d'une *Equation Différentielle Moyenne* dite ODE selon l'abréviation anglo-saxonne habituelle d'*Ordinary Differential Equation*. Celle-ci constitue un système dynamique non-linéaire déterministe, à temps continu. Ce chapitre présente la méthode de l'ODE, qui est la plus répandue. Elle a été introduite pour les algorithmes adaptatifs par [Ljung 83], développée ensuite dans le contexte signal par [Benveniste 87] et dans des contextes beaucoup plus généraux d'approximation stochastique par [Kushner 97]. On la retrouve également dans la littérature statistique [Duflo 90]. L'approche directe en discret a été développée par [Solo 95].

3.2.1. Forme générale, vecteur d'état

On cherche à estimer un vecteur θ_* de manière récursive. Les informations disponibles au temps n sont groupées dans un vecteur X_n appelé *état du système*. La forme d'algorithme étudiée doit être suffisamment générale pour permettre d'englober la plupart des systèmes adaptatifs et assez particulière pour se prêter à l'étude théorique. Pour cela, on utilise des algorithmes itératifs de la forme :

$$\theta_n = \theta_{n-1} + \mu_n H(\theta_{n-1}, X_n) \quad (3.8)$$

faisant apparaître les éléments suivants :

- μ_n pas de calculs, scalaires positifs tels que $\sum_1^\infty \mu_n = +\infty$. On distinguera les algorithmes à pas décroissant (ou à mémoire longue) pour lesquels $\mu_n \rightarrow 0$, et les algorithmes à pas constant (ou à mémoire courte) pour lesquels $\mu_n = \mu > 0$.
- X_n vecteur d'état de l'algorithme, mémoire finie de taille fixe assurant le caractère récursif en ligne

– $H(\theta, X)$ le *champ de vecteur* de l'algorithme, il détermine la manière dont l'algorithme corrige l'estimée θ_n en fonction des nouvelles informations X_n qui lui sont fournies à la $n^{\text{ième}}$ itération. C'est une fonction déterministe du paramètre et de l'état, calculable directement sur les données (pas d'opérateur d'espérance mathématique)

Le traitement sous cette forme générale des algorithmes newtoniens, incluant une mise à jour en deux étapes avec relaxation, nécessite l'utilisation d'un paramètre étendu (θ, Γ) incluant le gain matriciel, et induit dans (3.8) un terme de perturbation en $O(\mu_n^2)$ [Benveniste 87]. Ce terme étant "négligeable" dans les études asymptotiques à $\mu_n \rightarrow 0$, nous ne l'écrivons pas ici.

Hypothèses sur le vecteur d'état

On supposera vérifiées les deux hypothèses générales (HG) suivantes :

(HG1) L'état X_n admet une représentation markovienne contrôlée par $\theta : X_n = f(\xi_n)$ où ξ_n est appelé état étendu et

$$P(\xi_n \in G \mid \xi_{n-1}, \dots, \theta_{n-1}, \dots) = \int_G \Pi_{\theta_{n-1}}(\xi_{n-1}, dx).$$

Π_θ représente la probabilité de transition de la chaîne de Markov ξ_n . Pour tout θ appartenant au domaine utile au fonctionnement de l'algorithme (domaine de stabilité S), la chaîne de Markov contrôlée ξ_n sera supposée asymptotiquement stationnaire de distribution limite $P_\theta(d\xi)$. Ceci signifie que la loi de X_n tend vers une loi limite lorsque $n \rightarrow \infty$ (loi invariante par la probabilité de transition Π_θ). En pratique, les deux cas vérifiant cette hypothèse markovienne seront les suivants :

(HG1') L'état X_n est stationnaire indépendant de θ (cas standard)

(HG1'') L'état obéit à une dynamique linéaire markovienne stable, définie par la récurrence

$$X_n = A(\theta_{n-1})X_{n-1} + B(\theta_{n-1})W_n \quad (3.9)$$

où W_n est un bruit blanc vectoriel (suite i.i.d. centrée). Le domaine de stabilité S sera alors défini à l'aide du rayon spectral ρ de A :

$$S = \{\theta ; \rho(A(\theta)) < 1\}.$$

Comme en (3.6), on associera à la récurrence (3.9) l'équation de "l'état gelé" à θ constant

$$X_n(\theta) = A(\theta)X_{n-1}(\theta) + B(\theta)W_n \quad (3.10)$$

et on raisonnera par rapport à la solution stationnaire, dont la loi est la probabilité invariante P_θ de la chaîne de Markov.

Le champ moyen

Le champ moyen est défini par :

$$h(\theta) = E(H(\theta, X_n)),$$

l'espérance étant prise par rapport à la loi stationnaire invariante de l'état X_n . L'hypothèse de régularité portera sur ce champ moyen.

(HG2) La fonction $h(\theta)$ existe et est suffisamment régulière (Lipschitzienne).

Le champ moyen $h(\theta)$ indique la manière dont l'erreur d'estimation est prise en compte (en moyenne).

3.2.2. Algorithme déterministe et méthode de l'ODE

On associe à l'algorithme adaptatif (3.8), l'algorithme moyenné ou déterministe

$$z_n = z_{n-1} + \mu_n h(z_{n-1}) \quad (3.11)$$

Cet algorithme nous donnera les attracteurs possibles de (3.8) et une mesure par sa vitesse de convergence de la durée du transitoire en adaptatif en fonction du choix de la suite de pas μ_n de calcul. Ainsi, dans le cas d'un algorithme du gradient stochastique, la version moyennée n'est autre que l'algorithme du gradient déterministe initial : l'opération de moyennage pour l'étude de convergence peut être vue comme l'inverse de l'opération d'approximation stochastique servant à construire ces algorithmes adaptatifs.

L'heuristique permettant de comprendre la convergence des trajectoires de (3.8) vers celles de (3.11) lorsque le pas de calcul fixe $\mu \rightarrow 0$, est motivée par :

1) le fait que θ_n est lentement variable par rapport à X_n ($\mu \ll 1$) et du fait que H est régulière.

2) la loi des grands nombres qui permet d'écrire :

$$\begin{aligned} \theta_{n+N} &= \theta_n + \mu \sum_{i=0}^{N-1} H(\theta_{n+i}, X_{n+i+1}) \\ &\simeq \theta_n + (N\mu) \frac{1}{N} \sum_{i=0}^{N-1} H(\theta_n, X_{n+i+1}) \\ &\simeq \theta_n + (N\mu) h(\theta_n) \end{aligned}$$

On se reportera au Chapitre 1, section (1.4.1), pour l'étude de la vitesse de convergence du GD permettant de mesurer la durée transitoire de l'algorithme LMS du cas

standard (Chapitre 2). De même, l'algorithme déterministe DRLS associé au RLS est donné en (2.47) et la durée du transitoire de l'algorithme est présentée dans la figure [Fig.2.5]. Bien sûr, cette interprétation de l'algorithme moyenné en termes d'attracteurs potentiels et de vitesse transitoire de l'algorithme adaptatif suppose qu'on établisse la comparaison des trajectoires de ces deux algorithmes et leur proximité lorsque le pas de calcul est suffisamment petit. Cette étude directe sur les algorithmes itératifs est faite dans [Solo 95]. Commençons par l'exemple basique, avant les résultats généraux.

EXEMPLE 2 : ESTIMATEUR DE MOYENNE. Le cas de l'estimation d'une moyenne a été présenté au Chapitre 1, (p. 34). sa version adaptative en (2.2). Ce cas d'école montre comment l'estimateur empirique de la moyenne peut être mis sous la forme d'un algorithme récursif. Ce parallèle incite à transposer au domaine des algorithmes adaptatifs deux résultats classiques de la théorie des probabilités que sont la loi des grands nombres et le théorème de la limite centrale. Ces deux résultats s'interprètent alors en termes de théorème de convergence presque sûre, pour ce qui est de la loi des grands nombres, et de théorème de caractérisation de la vitesse de convergence asymptotique, en ce qui concerne le théorème de la limite centrale. La généralisation de ces deux concepts à une forme très générale d'algorithme en contexte non stationnaire, conduit à la méthode de l'équation différentielle moyenne (ODE) ainsi qu'à des résultats de normalité asymptotique.

Ici, la convergence de l'algorithme stochastique repose sur l'ergodisme de la suite X_n , c'est-à-dire sur la loi des grands nombres. Le fonctionnement de cet algorithme d'estimation de moyenne est illustré par la figure 3.2. Sur cette figure, on peut voir plusieurs trajectoires de l'estimateur θ_n (courbes en trait plein). Les deux courbes en pointillés représentent l'enveloppe à plus et moins deux fois l'écart type théorique de l'estimateur. On vérifie ainsi expérimentalement la convergence en moyenne quadratique. Les conditions de simulation sont : calcul sur 1000 points, les observations sont une suite de v.a. indépendantes, centrées et de variance égale à un.

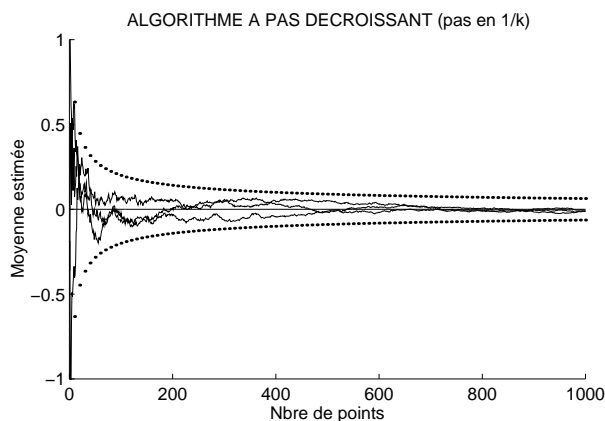


Figure 3.2. Estimateur empirique de la moyenne.

Méthode de l'ODE, horizon fini

Le but de cette méthode est d'analyser la convergence en moyenne de l'algorithme (3.8). Il s'agit de l'étape la plus simple du point de vue mathématique mais aussi la plus essentielle du point de vue de l'utilisateur. Nos hypothèses permettent d'associer à un algorithme de la forme (3.8) son équation différentielle moyenne (ODE) ; elle est définie à partir du champ moyen par :

$$\frac{d\theta}{dt} = \theta' = h[\theta(t)]. \quad (3.12)$$

L'algorithme moyen, déterministe (3.11) est interprété comme une discrétisation de cette équation différentielle déterministe avec l'équivalence temps continu-temps discret

$$t_n = \sum_{k=1}^n \mu_k \quad \text{avec} \quad \lim_{n \rightarrow \infty} t_n = +\infty.$$

Si on construit par interpolation linéaire à partir de la solution z_n de (3.11), la fonction continue $\theta(t)$, on a

$$\frac{\theta(t_n) - \theta(t_{n-1})}{\mu_n} = h(\theta_{n-1}) \rightarrow \theta'(t_n) \quad \text{si} \quad \mu_n \rightarrow 0$$

donc une convergence vers une trajectoire de l'ODE (3.12).

L'ODE caractérise le comportement statistique moyen d'un algorithme et le lien entre ses trajectoires et celles de l'algorithme adaptatif est fixé, sur une durée T , par le théorème suivant qui s'applique à tout type d'algorithme, avec $\mu = \max(\mu_n, t_n < T)$. On note $\theta(t)$ la trajectoire de l'ODE, de même condition initiale θ_0 que l'algorithme.

THÉORÈME. – Benveniste 87

S'il existe un tube de diamètre non nul entourant la trajectoire de l'ODE dans lequel les hypothèses (HG) précédentes sont vérifiées, alors pour $\varepsilon > 0$, μ petit et $t < T$ fini on a :

$$P(\max \|\theta_n - \theta(t_n)\| > \varepsilon) < C(\mu, T)$$

où $C(\mu, T) \rightarrow 0$ si $\mu \rightarrow 0$.

Ce résultat garantit que l'algorithme reste proche de l'ODE pour μ petit. Le contrôle uniforme sur toute la trajectoire (finie) est une propriété forte.

Horizon infini, convergence p.s. et stabilité

Ayant contrôlé la proximité des trajectoires sur une durée finie T , la question suivante est celle du devenir des trajectoires lorsque $T \rightarrow \infty$. Notamment quels sont les liens entre les attracteurs de l'ODE et ceux de l'algorithme (voir la section 3.2.3 pour la définition des attracteurs, et les outils d'étude de leur stabilité).

Dans de nombreux cas, l'ODE possédera un attracteur unique θ_* (le paramètre optimal recherché), de domaine d'attraction D_* : plaçons-nous dans cette hypothèse simple.

On aimerait obtenir un résultat de convergence p.s. de l'algorithme vers ce θ_* . Nous avons déjà vu au Chapitre 2, pour le cas standard, que ceci n'est possible que pour des algorithmes à pas décroissant, et nous avons alors obtenu ce résultat de convergence p.s.. Ces résultats particuliers au cas standard ne s'étendent malheureusement pas directement aux algorithmes étendus, en raison des problèmes posés

- par la non-linéarité en θ du champ de vecteur $H(\theta, X_n)$
- par les risques d'instabilité et d'explosion liés au bouclage entre l'état X_n et le paramètre θ .

La question de la stabilité exponentielle des transitoires est traitée, pour les modèles linéaires étendus, par [Anderson 82] et surtout par [Solo 95]. Elle passe par la construction d'un "système d'erreur" : algorithme sur l'écart $\theta - \theta_*$ dans le cas des modèles non-bruités (à l'optimum θ_* , l'erreur d'estimation est nulle), et établit effectivement par des techniques de fonctions de Liapounoff des propriétés de stabilité exponentielle pour l'algorithme stochastique (3.8), à partir des mêmes propriétés pour l'algorithme moyen (3.11).

Pour la convergence p.s. (dans le cas des algorithmes à pas décroissant), elle n'est obtenue par la méthode de l'ODE ([Ljung 83], [Benveniste 87]) que sous des hypothèses qu'on ne peut établir a priori sur la suite θ_n de l'algorithme : bornitude, et récurrence-stabilité (retour infiniment souvent dans le domaine d'attraction D_* de l'attracteur θ_*).

Les seuls résultats effectifs sont établis par [Kushner 97] sur les algorithmes contraints. On s'y reportera pour des approches détaillées et multiples de cette question sur des algorithmes stochastiques très généraux. Nous citerons le résultat principal, lié au modèle d'état markovien que nous avons supposé ici. L'algorithme contraint est défini par un domaine borné B inclus dans le domaine de stabilité de l'état ($B \subset S$), et un procédé Π_B de reprojektion sur B lorsqu'on en sort. Il s'écrit :

$$\theta_n = \Pi_B[\theta_{n-1} + \mu_n H(\theta_{n-1}, X_n)] = \theta_{n-1} + \mu_n H(\theta_{n-1}, X_n) + \mu_n Z_n \quad (3.13)$$

où Z_n désigne la correction de projection éventuelle sur B . On lui associe l'ODE contrainte

$$\theta' = h(\theta) + z(t) \quad z \in -C(\theta) \quad (3.14)$$

où $C(\theta)$ est le cône de projection (réduit à 0 si $\theta \in B$). On note L_B l'ensemble limite de l'ODE dans B (s'il y a un paramètre optimal unique, $L_B = \{\theta_*\}$). On introduit la perturbation totale $p_n(\theta, X_n) = \sum_n^{\infty} \mu_k [H(\theta, X_k(\theta)) - h(\theta)]$ initialisée à $X_n(\theta) = X_n$ dans l'équation de "l'état gelé" (3.10), les conditions de convergence sont du type

- (C1) $\sup E(|H(\theta_n, X_n)|) < \infty$
- (C2) $H(\theta, X_n)$ est continu en θ
- (C3) la perturbation totale est "asymptotiquement négligeable" p.s.

On définit par interpolation de l'algorithme (3.13) la fonction $\theta^0(t)$ continue sur \mathbb{R}^+ , ainsi que ses translatées $\theta^n(t) = \theta^0(t_n + t)$ avec $t_n = \sum_1^n \mu_k$ (de même pour Z). On obtient le théorème de convergence de Kushner suivant :

THÉORÈME.— *Convergence p.s.* [Kushner 97]
Sous les conditions (C), pour l'algorithme contraint (3.13) à pas décroissant $\mu_n \rightarrow 0$, en dehors d'un ensemble de probabilité nulle N , toute limite $(\theta(t), Z(t))$ d'une sous-suite convergente de la suite des translatées $(\theta^n(t), Z^n(t))_{n \in \mathbb{N}}$ est solution de l'ODE (3.14). La suite de l'algorithme θ_n converge vers un ensemble invariant de l'ODE dans B , soit L_B . Si θ_ est l'unique attracteur de l'ODE, alors*

$$\theta_n \rightarrow \theta_* \quad p.s.$$

Signalons un résultat utile de [Fort et Pagès, Th. 6] qui précise que sous certaines conditions techniques d'existence d'une fonction judicieuse de Liapounoff, si l'ODE admet un nombre fini de points stationnaires asymptotiquement stables et que chaque solution de l'ODE (3.12) converge vers un de ces points (sauf sur un ensemble de conditions initiales de mesure nulle), alors θ_n donné par l'algorithme d'approximation stochastique (3.8) converge presque sûrement vers un de ces points. C'est en particulier le cas où le champ moyen $h(\theta)$ dérive d'un potentiel : $h(\theta) = -\frac{1}{2} \nabla J(\theta)$ où $J(\theta)$ est une fonction positive admettant un nombre fini de minima locaux.

Enfin le résultat suivant [Benveniste 87, Th. 7 p. 50] justifie l'étude des points stationnaires stables de l'ODE pour la convergence des algorithmes d'approximations stochastiques à pas décroissants : s'il existe un point θ_* tel que $P(\theta_n \rightarrow \theta_*) > 0$ alors nécessairement θ_* est un point stationnaire au moins localement stable de l'ODE.

3.2.3. Stabilité de l'ODE et fonction de Liapounoff

Le comportement asymptotique de l'algorithme stochastique ($n \rightarrow \infty$) va être étroitement relié au comportement asymptotique des trajectoires de l'ODE associée : c'est donc la question de la stabilité de cette ODE et de ses attracteurs qui doit être étudiée.

Les concepts et outils fondamentaux ont été introduits par [Liapounoff 1907], et se retrouvent dans [La Salle 61]. Considérons une ODE définie sur un ouvert $\Omega \subset \mathbb{R}^d$:

$$\theta' = h(\theta) \quad \text{avec } h \text{ fonction continue}$$

Un ensemble $A \subset \Omega$ est dit *localement stable* (au sens de Liapounoff) si pour tout $\delta > 0$, il existe $\delta_1 > 0$ tel que toute trajectoire partant d'un voisinage $V(A, \delta_1)$ reste dans le voisinage $V(A, \delta)$. Si toutes les trajectoires partant de $V(A, \delta_1)$ convergent dans A , on parle d'un ensemble *asymptotiquement stable* (au sens de Liapounoff), ou *attracteur local*. On parle aussi d'*ensemble invariant* par le flot associé à l'ODE. Enfin, si cette convergence vers A a lieu pour toute condition initiale $\theta_0 \in \Omega$, on parle d'un *attracteur global* de l'ODE. La plupart du temps, on s'intéressera à un ensemble A réduit à un point stationnaire : θ_* vérifiant $h(\theta_*) = 0$ ou à l'ensemble fini ou dénombrable de ces points stationnaires.

Stabilité par linéarisation

Dans le cas d'une ODE linéaire $\theta'(t) = A\theta(t)$ pour une matrice fixée A régulière, le seul point stationnaire est 0. Comme la solution de l'ODE s'écrit ici $\theta(t) = e^{tA}\theta(0)$, le point d'équilibre est attracteur (global) ssi $\text{Re}[\lambda(A)] < 0$ (valeurs propres de A à partie réelle strictement négative).

Pour un point stationnaire θ_* d'une ODE non linéaire $\theta' = h(\theta)$, on étudie l'équation linéarisée

$$\theta' = D_h(\theta_*)(\theta - \theta_*) \quad D_h \text{ Matrice différentielle de } h.$$

La condition suffisante de stabilité asymptotique (locale) de θ_* pour l'ODE est alors

$$\text{Condition de stabilité} \quad \text{Re}[\lambda(D_h(\theta_*))] < 0$$

Evidemment, ceci donne une condition d'attraction locale, et ne permet pas de conclure sur l'attraction globale. S'il existe au moins une valeur propre de $D_h(\theta_*)$ à partie réelle strictement positive, θ_* est un équilibre instable de l'ODE. Dans les autres cas, on ne peut rien dire sur θ_* à partir du seul système linéarisé.

Stabilité par fonction de Liapounoff

Cette méthode a l'avantage de s'appliquer directement à des ODE non linéaires, et de permettre une preuve de stabilité globale et non plus seulement locale. On se ramène par translation au cas d'une ODE

$$\theta'(t) = h(\theta(t)) \tag{3.15}$$

possédant le point d'équilibre 0 (càd $h(0) = 0$). Notons $\theta_a(t)$ la trajectoire partant du point a à $t = 0$.

Une *fonction de Liapounoff* sera une fonction $V : \Omega \rightarrow \mathbb{R}^+$, de classe C^1 , telle que

$$(L1) \quad V(0) = 0 \quad \text{et} \quad \forall \theta \neq 0, \quad V(\theta) > 0$$

$$(L2) \quad \frac{d}{dt}V(\theta(t)) = \nabla V(\theta(t))^T h(\theta(t)) \equiv -k(\theta(t)) \leq 0,$$

donc $V(\theta(t))$ est décroissante le long des trajectoires de l'ODE. Si une telle fonction existe, alors le point 0 est un *attracteur local* de l'ODE. Le cas le plus intéressant est celui où cette fonction vérifie en plus la condition de *coercivité*

$$(L3) \quad V(\theta) \rightarrow +\infty \quad \text{lorsque } \theta \rightarrow \partial\Omega$$

On a alors le théorème suivant :

THÉORÈME.— *Stabilité globale de l'ODE [Liapounoff]*

Sous les conditions (L1), (L2), (L3), l'ensemble $K_\alpha = \{\theta : V(\theta) \leq \alpha\}$ est un compact de Ω . Toute trajectoire $\theta(t)$ de l'ODE (3.15) converge pour $t \rightarrow \infty$ vers l'attracteur global

$$D_* = \{\theta \in K_\alpha : k(\theta) = 0\} \quad \text{avec } \alpha = V(\theta(0)).$$

La preuve en est simple : par décroissance, on a $\forall t > 0, \theta(t) \in K_\alpha$. De plus sur l'ensemble compact K_α la fonction $f(t) = V(\theta(t))$ est positive, décroissante, de dérivée $f'(t) \leq 0$. Il existe donc des limites $a = \inf(f(t), t > 0)$ et $b = \inf(-f'(t), t > 0)$. Si on avait $b > 0$, cela entraînerait $f(t) < C - bt \rightarrow -\infty$, impossible. Donc $b = 0$ et $\lim_{t \rightarrow \infty} k(\theta(t)) = 0$, ce qui entraîne par continuité de k la convergence de $\theta(t)$ vers D_* .

Par exemple, si le champ de vecteur de l'ODE dérive d'un potentiel positif :

$$h(\theta) = -\frac{1}{2}\nabla J(\theta),$$

on prend comme fonction de Liapounoff $V(\theta) = J(\theta) - J_{\min}$, on a $\frac{d}{dt}V(\theta(t)) = -\|h(\theta(t))\|^2$ qui tend vers 0. Il en résulte ici que, si la fonction $J(\theta)$ vérifie la condition de coercivité (L3), l'ODE possède comme attracteur global l'ensemble $D_* = \{\theta_* : \nabla J(\theta_*) = 0\}$ des *points stationnaires* de la fonction J .

3.2.4. Exemples : ODE, convergence, condition SPR

EXEMPLE 3 : ODE DE L'ESTIMATEUR DE MOYENNE. Ces résultats permettent en particulier de retrouver la convergence de l'estimateur de la moyenne empirique. Pour cet exemple, nous avons : $h(\theta) = \theta_* - \theta$ (on observe une suite de v.a. i.i.d. de variance σ^2). L'algorithme moyen associé à (2.2) est donc la récurrence linéaire

$$\theta_n = (1 - \mu_n)\theta_{n-1} + \mu_n\theta_*$$

dans le cas du pas fixe $\mu_n = \mu$, il est stable ssi $0 < \mu < 2$. Il donne l'ODE :

$$\frac{d\theta}{dt} = \theta_* - \theta$$

et

$$\theta_* - \theta_{t_n} = (\theta_0 - \theta_*) \exp(-n\mu).$$

Ainsi, l'erreur d'estimation décroît exponentiellement vers zéro, et ce d'autant plus rapidement que μ est grand. Il faut néanmoins prendre garde au fait que la trajectoire de l'ODE n'est réellement proche de celle l'algorithme que sous la condition $\mu \rightarrow 0$.

EXEMPLE 4 : ODE DU LMS-RII. Puisqu'on a ici un algorithme de gradient stochastique, l'ODE dérive ici du potentiel $J(\theta)$:

$$\theta' = E(e_n(\theta)\psi_n(\theta)) = -\frac{1}{2}\nabla J(\theta)$$

et les trajectoires de l'ODE sont les lignes de plus grande pente de la surface de coût $J(\theta)$. Il y a "bonne modélisation", donc θ_* est l'unique point stationnaire et minimum global de $J(\theta)$, c'est donc aussi l'attracteur global unique de l'ODE sur le domaine de stabilité S . Par application des théorèmes précédents, on conclut que, si l'algorithme n'explose pas, il convergera p.s. vers θ_* . On ne peut garantir ceci qu'en appliquant l'algorithme contraint (3.13) avec projection sur un domaine borné $B \subset S$.

EXEMPLE 5 : ODE DE ELS ET CONDITION SPR. Dans l'exemple de l'identification RII ci-dessus, l'algorithme ELS a été obtenu en supprimant le filtrage du vecteur d'état. Il s'agit donc d'un algorithme de "pseudo-gradient", et le champ moyen ne dérive plus d'un potentiel, l'ODE doit donc être étudiée spécifiquement. L'algorithme et l'ODE sont les suivants :

ELS

$$\begin{aligned} \theta_n &= \theta_{n-1} + \mu_n R_n^{-1} e_n \varphi_n & \text{avec} & \quad \mu_n = \frac{1}{n} \\ R_n &= R_{n-1} + \mu_n (\varphi_n \varphi_n^T - R_{n-1}) \end{aligned}$$

ODE de ELS

$$\theta' = R^{-1}h(\theta) \quad R' + R = \Sigma(\theta)$$

avec ici

$$h(\theta) = E(e_n(\theta)\varphi_n(\theta)) = \left[\frac{ab^2}{a^2-1} + \frac{ab\beta}{1-a\alpha}, \beta-b \right]^T \quad \Sigma(\theta) = E(\varphi_n \varphi_n^T) = \begin{pmatrix} \frac{b^2}{1-a^2} & 0 \\ 0 & 1 \end{pmatrix}$$

On vérifie à nouveau que $\theta_* = (\alpha, \beta)^T$ est l'unique équilibre de l'ODE. L'étude du caractère attracteur de cet équilibre est fait dans [Benveniste 87] (dans un contexte dual de modélisation ARMA), et utilise pour construire une fonction de Liapounoff (assurant l'attraction globale sur le domaine de stabilité), une condition de *Stricte Positivité Réelle* communément appelée SPR, suffisante mais non nécessaire pour la stabilité de l'équilibre :

$$\text{Condition SPR} \quad \text{Re} \left(\frac{1}{A^*(e^{2i\pi\nu})} \right) - \frac{1}{2} > 0 \quad \forall \nu$$

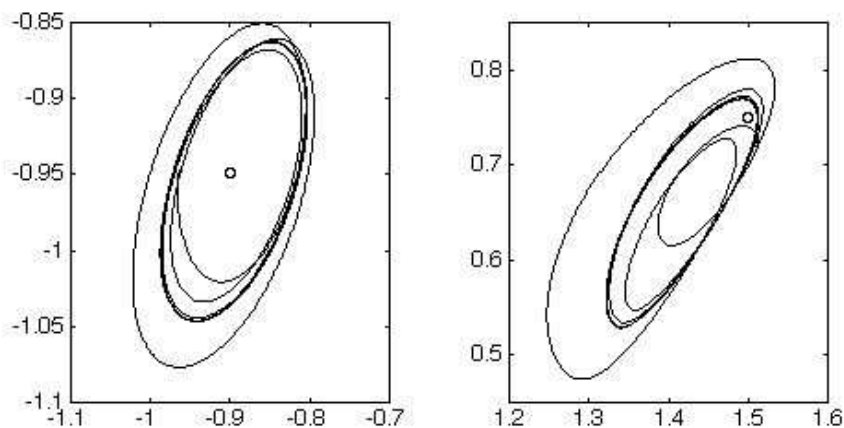


Figure 3.3. ELS : contreexemple, cycle limite (plans α et β)

Cette condition est donc importante pour garantir la convergence vers l'optimum de ELS. Un contreexemple est donné par [Ljung 75] dans le cas de la modélisation ARMA : il s'agit du modèle ARMA(2-2) donné par $\alpha = (-0.9, 0.95)$ et $\beta = (1.5, 0.75)$. Ce modèle met en défaut la condition SPR, et il montre que θ_* n'est pas attracteur, donc qu'aucun point ne peut l'être. On a dans ce cas un cycle limite-attracteur de l'ODE (voir fig3.3) et l'algorithme ELS oscille autour de l'optimum.

Convergence p.s. de ELS2 : un résultat de [Solo 79]

Cet article démontre pour la variante de ELS "à erreur a posteriori", à pas décroissant, pour l'identification RII ou la modélisation ARMA, que la condition SPR entraîne la convergence p.s. $\theta_n \rightarrow \theta_*$.

EXEMPLE 6 : NEURONE D'OJA. L'ODE associée au neurone d'Oja (3.4) est

$$\frac{dw}{dt} = R_x w - w w^T R_x w.$$

Elle admet comme points stationnaires, les solutions de $R_x w = (w^T R_x w)w$, soit tout vecteur propre normalisé $\pm v_i$ de R_x . Pour étudier la stabilité de ces points stationnaires, considérons les valeurs propres de la dérivée du champ moyen $D_h(w) = R_x - (w^T R_x w)I_m - 2w w^T R_x$ aux points $\pm v_i$, soit $D_h(w)|_{w=\pm v_i} = R_x - \lambda_i I_m - 2\lambda_i v_i v_i^T$. Puisque ses valeurs propres sont $-2\lambda_i$ et $(\lambda_j - \lambda_i)_{j \neq i}$, ces valeurs propres sont ici réelles et sont toutes strictement négatives que si $i = 1$. Par suite $\pm v_1$ sont les seuls points localement asymptotiquement stables de l'ODE associée.

Si nous considérons l'estimation adaptative conjointe de v_1 et λ_1 , nous remarquons que les algorithmes adaptatifs imbriqués (3.4) et (3.5) peuvent se mettre sous la forme générale (3.8) à

l'aide du paramètre étendu $\theta_n \equiv (w_n^T, \alpha_n)^T$ et du champ moyen étendu

$$\tilde{h}(\theta) = \begin{pmatrix} R_x w - w w^T R_x w \\ w^T R_x w - \alpha \end{pmatrix}$$

dont les points stationnaires sont $\theta_* = (\pm v_i^T, \lambda_i)^T$. Puisque

$$D_{\tilde{h}}(\theta_*) = \begin{pmatrix} D_h(w) & 0 \\ 2w^T R_x & -1 \end{pmatrix} \Big|_{w=\pm v_i, \alpha=\lambda_i}$$

a pour valeurs propres $-2\lambda_i$, $(\lambda_j - \lambda_i)_{j \neq i}$ et -1 , $(\pm v_1, \lambda_1)$ sont les seuls points localement asymptotiquement stables de l'ODE associée.

3.3. Algorithmes à pas constant : fluctuation stationnaire

Pour les algorithmes à pas constant $\mu_n = \mu$, les seuls utilisés pour être vraiment adaptatifs et actifs en continu sur un flux de données, il ne peut y avoir de convergence vers l'optimum θ_* . Après la phase transitoire où l'algorithme se rapproche du voisinage de θ_* en suivant de plus ou moins près les trajectoires de l'ODE, on entre dans une phase de fluctuation asymptotique. L'analyse du second ordre va permettre de préciser la démarche de l'ODE, en estimant la variance de fluctuation. On dispose dans ce but de deux types d'approches : les méthodes asymptotiques (à $\mu \rightarrow 0$), prolongeant la méthode de l'ODE, et des méthodes non asymptotiques, basées sur l'équation récurrente de la fluctuation et de sa variance.

EXEMPLE 7 : ANALYSE DIRECTE DE L'ESTIMATEUR DE MOYENNE. La version à pas fixe μ de l'algorithme de gradient stochastique (2.2) est

$$\theta_n = \theta_{n-1} + \mu(X_n - \theta_{n-1}) \quad (3.16)$$

avec $0 < \mu < 2$ et $\theta_0 = 0$. Dans cet algorithme, μ est un petit paramètre scalaire qui fixe la longueur de la fenêtre de pondération. La faculté qu'a ce nouvel algorithme d'oublier le passé lui permet de poursuivre les variations de $E(X_n)$. En contrepartie, dans le cas où la moyenne à estimer est une constante ($E(X_n) = E(X)$), la variance de l'écart $(\theta_n - E(X))$ ne tend plus vers zéro. En effet, on peut écrire :

$$E(\theta_n) = (1 - \mu)E(\theta_{n-1}) + \mu E(X_n)$$

Ceci montre que la suite $E(\theta_n)$ converge bien vers $\theta_* = E(X)$. Par contre, on a aussi :

$$\text{var}(\theta_n) = (1 - \mu)^2 \text{var}(\theta_{n-1}) + \mu^2 \sigma^2$$

La variance asymptotique de cet estimateur est donc non nulle. Plus précisément, cette récurrence donne : $\text{var}(\theta_n) = \frac{\mu \sigma^2}{2 - \mu} (1 - (1 - \mu)^{2n})$, soit :

$$\lim_{n \rightarrow \infty} \text{var}(\theta_n) = \frac{\mu \sigma^2}{2 - \mu} \approx \mu \frac{\sigma^2}{2} > 0 \quad \text{pour } \mu \ll 1 \text{ fixé.}$$

Pour ce type d'algorithme (gain μ constant), la convergence de θ_n vers θ_* n'est plus qu'une convergence en moyenne. Les algorithmes à gain constant ne sont pas destinés à identifier des paramètres invariants.

Le fonctionnement de l'algorithme (3.16) d'estimation de moyenne est illustré par la figure [Fig.3.4]. Sur cette figure, on peut observer les trajectoires de l'estimateur θ_n pour différentes valeurs du pas d'adaptation μ . On vérifie ainsi expérimentalement la convergence en moyenne vers la valeur zéro. Lorsque μ augmente, on vérifie également l'augmentation des fluctuations ainsi que la diminution de la durée de la convergence. Les conditions de la simulation sont : calcul sur 1 000 points, les observations sont une suite de v.a. i.i.d. centrées de variance égale à un, valeur initiale de l'estimateur égale à quatre, les différentes valeurs de μ utilisées sont notées sur la figure.

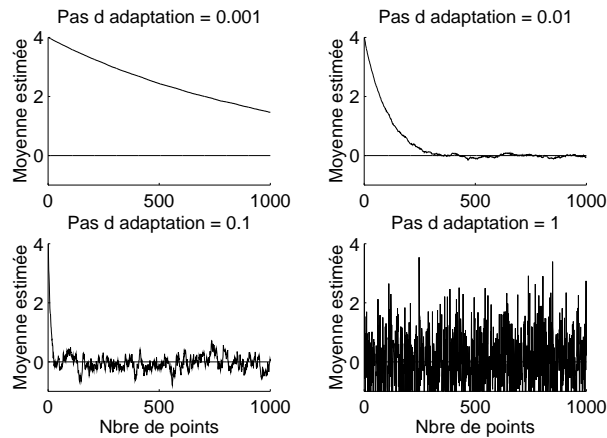


Figure 3.4. Estimateur adaptatif de la moyenne en stationnaire.

Convergence en loi des processus aléatoires

Dans le cas de l'algorithme estimateur de moyenne à pas décroissant $\mu_n = \frac{1}{n}$:

$$\theta_n = \theta_{n-1} + \frac{1}{n}(X_n - \theta_{n-1}) \quad \theta_0 = 0,$$

qui n'est autre que l'estimation classique $\theta_n = \frac{1}{N} \sum_{k=1}^N X_k$, un résultat classique caractérise alors son comportement asymptotique. En effet le *théorème de limite centrale* (TLC) affirme que $\sqrt{n}(\theta_n - E(X))$ converge en loi vers une v.a. gaussienne centrée de même variance que X caractérisant la vitesse de convergence asymptotique en indiquant que la variance de l'écart $(\theta_n - E(X))$ décroît en $1/n$.

Mais dans le cas de l'algorithme à pas fixe, la solution de la récurrence (3.16) s'écrit

$$\theta_n = \mu \sum_{i=0}^{n-1} (1 - \mu)^i X_{n-i}$$

et un simple calcul de la fonction caractéristique de $\theta_n - E(X)$ montre que même dans cet exemple simple, on ne peut conclure à des convergences en loi que dans des cas très particuliers. Ainsi pour μ fixé et X_n non gaussiens, $\theta_n - E(X)$ ne converge pas en loi vers une v.a. gaussienne de variance $\frac{\mu\sigma^2}{2}$ pour $n \rightarrow \infty$. De même nous ne pouvons pas conclure que l'écart $\frac{1}{\sqrt{\mu}}(\theta_n - E(X))$ converge asymptotiquement en loi quand $n \rightarrow \infty$ et $\mu \rightarrow 0$ indépendamment. Par contre, cet écart converge en loi vers une v.a. gaussienne de variance $\frac{\sigma^2}{2}$ si de plus $n\mu$ tend vers $+\infty$. D'un point de vue pratique nous dirons que pour $n \gg 1$, $\mu \ll 1$ et $n\mu > 3$, l'écart $\frac{1}{\sqrt{\mu}}(\theta_n - E(X))$ suit approximativement une loi gaussienne centrée de variance $\frac{\sigma^2}{2}$. Ces ordres de grandeur seront étendus aux algorithmes d'approximations stochastiques plus généraux (3.8) grâce à l'exploitation pratique de l'équation de Liapounoff (3.19). Cet exemple sera poursuivi d'une façon plus formelle dans l'exemple (8, page 122).

La généralisation de l'étude théorique du comportement en loi de la suite (3.16), qu'on notera θ_n^μ , pour la double asymptotique ($n \rightarrow \infty$, $\mu \rightarrow 0$) est donc plus complexe et relève des méthodes de convergence en loi des processus aléatoires et des *théorèmes de limite centrale fonctionnels*. Ils consistent à plonger le processus discret dans un espace E de processus à temps continu à trajectoires continues ([Billingsley 68], ou [Kushner 97, p. 228]). On étudie alors la convergence faible de la famille des lois de probabilité sur E des processus $\theta^\mu(t)$ lorsque $\mu \rightarrow 0$.

Le paragraphe suivant présente un outil général d'étude des fluctuations d'un algorithme adaptatif du type (3.8) autour de sa trajectoire moyenne (solution de l'ODE), basée sur cette approche fonctionnelle de la convergence en loi des processus.

3.3.1. Fluctuation : Approximation-Diffusion gaussienne et EDS

La convergence du transitoire étant assurée, il est alors légitime de s'intéresser au comportement d'un point de vue asymptotique. Le point de convergence est toujours supposé fixe. La question est de quantifier la variance de l'estimation fournie par l'algorithme. Les résultats mathématiques utilisés sont semblables au théorème de la limite centrale. On montre que les fluctuations de l'algorithme stochastique autour de sa trajectoire moyenne (celle que donne l'ODE) peuvent s'approcher à l'aide d'une diffusion gaussienne. Nous nous intéressons seulement aux algorithmes à gain constant.

3.3.1.0.1. Notations

- $D_h(\theta) = \partial_\theta h$ la dérivée du champ moyen.
- $G(\theta) = \sum_{-\infty}^{+\infty} \text{cov}_\theta [H(\theta, X_n), H(\theta, X_0)]$ la covariance (par rapport à μ_θ) cumulée du champ aléatoire H .
- la fluctuation normalisée de l'algorithme est

$$\tilde{\theta}_{t_n}^\mu = \frac{1}{\sqrt{\mu}} [\theta_n - \theta(t_n)] \quad (3.17)$$

et représente l'écart entre l'algorithme stochastique et son ODE vu au travers d'une loupe de grossissement $1/\sqrt{\mu}$. Ce terme de normalisation est équivalent au facteur \sqrt{n} du théorème central limite (pour un algorithme à pas décroissant $\mu_n = 1/n$, $1/\sqrt{\mu_n} = \sqrt{n}$).

- dW_t désigne l'accroissement d'un processus de Wiener-Lévy vectoriel de covariance identité (Mouvement brownien).

Lorsque μ tend vers zéro, l'interpolation linéaire par morceaux $\tilde{\theta}_t^\mu$ du processus discret $\tilde{\theta}_{t_n}^\mu$ converge en loi vers la solution de l'équation différentielle stochastique (EDS) :

$$d\tilde{\theta}_t = D_h[\theta(t)]\tilde{\theta}_t dt + G^{1/2}[\theta(t)] dW_t.$$

Nous ne faisons que mentionner ce résultat dont la démonstration difficile est donnée dans [Benveniste 87]. En horizon infini, $\theta(t)$ est proche de θ_* et $\tilde{\theta}_t^\mu$ converge en loi vers la solution de l'EDS linéaire homogène :

$$d\tilde{\theta}_t = D_h(\theta_*)\tilde{\theta}_t dt + G^{1/2}(\theta_*) dW_t \quad (3.18)$$

Nous disposons ainsi d'une approximation gaussienne de $\tilde{\theta}_t$. L'équation (3.18) étant linéaire et son entrée gaussienne, sa sortie est également gaussienne. $\tilde{\theta}_t$ est la sortie d'un filtre linéaire de fonction de transfert $[sI - D_h(\theta_*)]^{-1} G^{1/2}(\theta_*)$ attaqué par un bruit blanc gaussien de puissance unité. Si toutes les valeurs propres de h_θ sont de partie réelle négative, ce filtre est stable et sa sortie suit une loi normale centrée de matrice de covariance C_θ obtenue par résolution d'une équation de Liapounoff. Résumons cela dans le théorème suivant.

THÉORÈME.— Pas constant, horizon infini [Benveniste 87]

Sous les hypothèses (HG) d'un modèle d'état markovien exponentiellement stable, si l'ODE possède l'unique attracteur stable θ_ , la fluctuation normalisée (3.17) de (3.8) vérifie le théorème de limite-centrale :*

Si $\mu \rightarrow 0$ et $t \rightarrow \infty$, $\tilde{\theta}_t^\mu$ converge en loi vers $\mathcal{N}(0, C_\theta)$ et la covariance C_θ est l'unique solution positive de l'équation de Liapounoff

$$D_h(\theta_*)C_\theta + C_\theta D_h(\theta_*)^T + G(\theta_*) = 0. \quad (3.19)$$

En pratique ce théorème sera appliqué sous la forme suivante : si $\mu \ll 1$, $n \gg 1$ et $n\mu > 3$, la matrice de covariance de $(\theta_n - \theta_*)$ est approximativement égale à μC_θ

3.3.2. Exemples

EXEMPLE 8 : Diffusion gaussienne pour l'estimateur de moyenne. Ce résultat permet en particulier de retrouver la variance résiduelle de $\mu\sigma^2/2$ pour l'estimateur de moyenne. Pour cet exemple, nous avons : $D_h(\theta_*) = -1$ et $G(\theta_*) = \sigma^2$ (on observe une suite de v.a. i.i.d. de variance σ^2). Ainsi, l'équation de Liapounoff s'écrit : $-2C_\theta + \sigma^2 = 0$ d'où $C_\theta = \sigma^2/2$. Or $\tilde{\theta}_{t_n}^\mu = \frac{1}{\sqrt{\mu}} [\theta_n - \theta(t_n)]$, d'où la valeur $\mu\sigma^2/2$ pour la variance de l'estimateur. Il faut toujours prendre garde au fait que le résultat obtenu en résolvant l'équation de Liapounoff n'est rigoureusement valide que pour $\mu \rightarrow 0$.

EXEMPLE 9 : FLUCTUATION DU LMS STANDARD, CAS D'UNE RÉGRESSION LINÉAIRE. Traitons le cas du LMS (2.3), avec la fonction de coût d'EQM (2.1). On se place sur un modèle de régression linéaire, où le signal de référence est donné par $x_n = U_n^T w_* + b_n$ et b_n est un bruit blanc de variance σ_b^2 , indépendant de l'observation u_n . On a donc $e_n(w_*) = b_n$. Le champ moyen est ici $h(w) = p - R w = -R(w - w_*)$, donc la dérivée en est $D_h(w) = -R$, et la covariance totale est

$$G(w_*) = \sum_n \text{cov}[b_n b_0 U_n U_0] = \sigma_b^2 R,$$

l'équation de Liapounoff s'écrit donc

$$-RC_w - C_w R + \sigma_b^2 R = 0$$

sa solution positive est évidemment $C_w = \frac{1}{2}\sigma_b^2 I$. On obtient donc le résultat de fluctuation asymptotique

$$F(\mu) = \lim_{n \rightarrow \infty} E(\|w_n - w_*\|^2) = \mu \text{Tr}(C_w) = \mu \frac{J_{min}}{2} M$$

résultat qui précise (2.14) et redonne l'estimation de l'EQMR à "petit pas" obtenue par l'étude directe du LMS en (2.17).

EXEMPLE 10 : FLUCTUATION POUR LE LMS-RII D'ORDRE (1-0). Reprenons l'exemple (3.1). On a déjà vu que $h(\theta) = -\nabla J(\theta)$, donc $D_h(\theta) = -\frac{1}{2}\nabla^2 J(\theta)$. Par ailleurs, comme dans l'exemple précédent, on a pour la covariance totale

$$G(\theta) = \sum_n \text{cov}[b_n b_0 \psi_n \psi_0] = \sigma_b^2 E(\psi_0 \psi_0^T)$$

et à l'optimum, d'après (3.7), on obtient $G(\theta_*) = \sigma_b^2 R$ et $D_h(\theta_*) = -R$, l'équation de Liapounoff s'écrit donc

$$-RC_\theta - C_\theta R + \sigma_b^2 R = 0$$

sa solution positive est à nouveau $C_\theta = \frac{1}{2}\sigma_b^2 I$. Ceci montre qu'asymptotiquement (pour $\mu \rightarrow 0$), les estimateurs des différents paramètres sont indépendants de même variance $\mu \frac{I_{min}}{2}$. Dans cet exemple, on peut aussi calculer explicitement le champ moyen et la covariance de l'état filtré : on obtient

$$h(\theta) = \left[\frac{\alpha\beta b}{(1-\alpha\alpha)^2} - \frac{ab^2}{(1-a^2)^2}, \frac{\beta}{1-\alpha\alpha} - \frac{b}{1-a^2} \right]^T$$

qui s'annule ssi $a = \alpha$ et $b = \beta$.

EXEMPLE 11 : NEURONE D'OJA. Nous avons vu dans l'exemple 6, page 117 que $D_{\tilde{h}}(v_1) = R_x - \lambda_1 I_m - 2\lambda_1 v_1 v_1^T$. Dans le cas d'observations x_n gaussiennes centrées indépendantes, il a été démontré dans [Delmas 98-a] que $G(v_1) = \sum_{k=2}^m \lambda_1 \lambda_k v_k v_k^T$ et que par suite la solution de l'équation de Liapounoff (3.19) est :

$$C_{v_1} = \sum_{k=2}^m \frac{\lambda_1 \lambda_k}{2(\lambda_1 - \lambda_k)} v_k v_k^T.$$

Si l'on considère l'algorithme adaptatif (3.5) qui estime conjointement la valeur propre λ_1 , alors $\theta_* = (v_1^T, \lambda_1)^T$ et la covariance cumulée du champ étendu devient :

$$G(\theta_*) = \begin{pmatrix} G(v_1) & 0 \\ 0 & 2\lambda_1^2 \end{pmatrix}$$

qui associée à la valeur de $D_{\tilde{h}}(\theta_*)$ donnée à l'exemple 6, page 117 donne comme solution de l'équation de Liapounoff (3.19)

$$C_\theta = \begin{pmatrix} C_{v_1} & 0 \\ 0 & \lambda_1^2 \end{pmatrix}.$$

Nous retrouvons les propriétés classiques de l'estimation batch des structures propres : les estimées w_n et α_n de v_1 et λ_1 sont asymptotiquement non corrélées bien que les algorithmes adaptatifs soient imbriqués et de plus le conditionnement de la recherche de la valeur propre ne pose pas de problèmes contrairement à celui du vecteur propre associé (cas où λ_2 est proche de λ_1).

Des expressions analytiques des matrices de covariance C_θ ont été données dans [Delmas 98-a] pour plusieurs algorithmes d'estimations de vecteurs propres principaux ou minorants. Il s'est avéré par simulation de Monte-Carlo que le domaine de validité de l'approximation $\text{cov}(\theta_n) \approx \mu C_\theta$ s'étend à d'assez grandes valeurs de pas μ ($\mu < 0.05$ pour $\text{Tr}(R_x) = m$ pour $n > 1000$). La déviation à l'orthonormalité : $d^2(\mu) \equiv E\|W_n^T W_n - I_r\|_{\text{Fro}}^2$ a été également estimée par simulation de Monte-Carlo où il a été montré que $d^2(\mu) \propto \mu$ [resp. $\propto \mu^2$] pour les algorithmes (introduits en section (3.5)) GHA et OFA [resp. WSA et SGA]. Les vitesses de convergence des différents algorithmes ont été comparées avec des valeurs de pas μ ajustées de façon à avoir des EQM asymptotiques numériquement identiques (ces EQM sont choisies égales à $\text{Tr}(\mu C_\theta)$ car un raisonnement heuristique nous montre que le biais asymptotique de θ_n varie en μ^{-2}). Cela permet des choix optimaux des paramètres (α_i et β_i) de ces algorithmes en fonction du compromis précision/vitesse de convergence choisi.

3.4. Poursuite de non-stationnarités

CONTENU.— Nous abordons ici la dernière phase générale d'étude des performances des algorithmes : la capacité de s'adapter efficacement à un environnement non-stationnaire à *variations lentes* (par rapport au temps d'adaptation des algorithmes). Elle ne concerne que les algorithmes à pas constant, et l'approche habituelle consiste à représenter la non-stationnarité sous forme d'une valeur optimale du paramètre variant dans le temps $\theta_*(n)$ qu'il s'agit de poursuivre le mieux possible. Le problème est très différent de ceux étudiés précédemment (vitesse de convergence initiale en transitoire, fluctuation résiduelle), et les critères et résultats de comparaison des algorithmes seront alors différents et liés à la nature de la non-stationnarité supposée. Pour une définition de base de la problématique de poursuite dans le contexte des algorithmes standards (LMS-RLS), on se reportera à [Macchi-Bellanger 78]. On présente ici la méthode générale, selon [Benveniste 87], application de la méthode de l'EDS déjà vue pour la fluctuation résiduelle. On présente enfin quelques résultats appliqués au cas des algorithmes standards du Chapitre 2.

3.4.1. Problématique de la poursuite

L'exemple de base, pris en général comme modèle pour la poursuite, est celui d'une *régression linéaire variable*

$$x_n = \varphi_n^T z_{n-1} + b_n \quad (3.20)$$

où b_n est un bruit blanc de puissance σ^2 , indépendant des vecteurs de régression φ_n , et z_n le vecteur de filtre optimal variable. Le signal d'observation u_n mis dans φ_n est supposé stationnaire, indépendant de b_n , de matrice de covariance $\Sigma = E(\varphi_n \varphi_n^T)$.

Les modèles de variation temporelles de z_n que l'on se donnera prendront l'une des formes suivantes :

- (HM1) Variation déterministe bornée : $z_n = z_{n-1} + \delta_n$ avec $\|\delta_n\| < d$,
- (HM2) Dérive constante $z_n = z_{n-1} + \delta z$ avec $\|\delta z\| = d$,
- (HM3) Marche aléatoire : $z_n = z_{n-1} + w_n$, avec $E(z_n) = z$ et $w_n =$ bruit blanc stationnaire, $\text{cov}(w_n) = Q$ et $\text{Tr}(Q) = d^2$.

Vitesse et degré de non-stationnarité

Pour la poursuite, on utilise obligatoirement un algorithme à *pas constant* μ , la seule chose qu'on puisse espérer est donc une *stabilisation* de l'écart quadratique moyen

$$\delta_n = E(\|\theta_n - z_n\|^2)$$

lorsque $n \rightarrow \infty$. On pourra dire que l'algorithme arrive à poursuivre le modèle si la limite $\delta = \lim \delta_n$ existe, et que la poursuite est impossible si au contraire l'EQM

δ_n explose quand $n \rightarrow \infty$. Si la poursuite a lieu, cette limite δ de l'EQM servira de mesure de performance de l'algorithme adaptatif.

D'un point de vue pratique, il est donc intéressant de pouvoir quantifier la "vitesse de non-stationnarité" permettant d'espérer une poursuite et la qualité de cette poursuite mesurée asymptotiquement par δ réalisée par l'algorithme. Nous partons pour cette problématique du cas particulier simple du gradient stochastique LMS suivant l'approche de [Macchi 87, 89], avant de passer à une étude plus globale des algorithmes. On reviendra ensuite à cet exemple simple pour lui appliquer ces résultats généraux.

Approche directe : cas du LMS

La *capacité d'adaptation* du LMS à pas μ ($\theta_n = \theta_{n-1} + \mu e_n \varphi_n$) peut être grossièrement mesurée par la puissance moyenne de l'incrément sur un pas de l'algorithme : supposant acquis le "vrai filtre" z_{n-1} , on aura $e_n = b_n$ et par indépendance avec φ_n on obtient la puissance d'incrément

$$\Delta(\mu) = \mu^2 \sigma^2 \text{Tr}(\Sigma)$$

qui sera majorée par la borne supérieure (optimiste) de μ compatible avec la stabilité du LMS. Nous avons vu au Chapitre 2 que l'ordre de grandeur de ce pas maximal était $\mu_{\max} \approx \frac{1}{\text{Tr}(\Sigma)}$ ce qui permet d'estimer la capacité d'adaptation maximale sur un pas du LMS à

$$\Delta_{\max} = \frac{\sigma^2}{\text{Tr}(\Sigma)}$$

Intuitivement, il est naturel de penser que cet algorithme arrivera à poursuivre une variation du filtre $T_n = z_n - z_{n-1}$ dont le carré moyen $d^2 = E(|T_n|^2)$ sera inférieure à la capacité d'adaptation Δ_{\max} : ceci nous conduit à définir le *degré de non-stationnarité* du modèle (3.20) par

$$v = \left(\frac{d^2 M \sigma_u^2}{\sigma^2} \right)^{1/2} \quad (3.21)$$

et à considérer que des *variations lentes* du modèle par rapport à l'algorithme seront définies par

$$\text{Variations lentes} \quad v \ll 1$$

Cette condition permettra d'espérer une poursuite correcte du modèle, et il nous reste alors à mesurer la performance selon le pas de calcul μ .

REMARQUE.— Il est important de noter que la vitesse de variation du modèle ne se mesure pas dans l'absolu, mais en relation avec le *rapport signal à bruit de sortie* optimal $\rho = \frac{\sigma_s^2}{\sigma^2}$ du filtre. Cela signifie que la poursuite est d'autant plus difficile que le bruit de sortie est faible.

3.4.2. Exemple de l'estimateur de moyenne

EXEMPLE 12 : ANALYSE DIRECTE DE L'ESTIMATEUR DE MOYENNE On suppose que la moyenne évolue, le rôle de l'algorithme adaptatif est alors de poursuivre efficacement ses variations. La mesure de cette efficacité passe nécessairement par une modélisation de l'évolution du paramètre à estimer. Cette évolution est couramment modélisée à l'aide d'une approximation stochastique formellement analogue à l'algorithme lui-même. Cette équation récurrente s'appelle l'*hypermodèle*. Une fois modélisées les variations du vrai système, l'efficacité de la procédure itérative peut être mesurée par la distance quadratique $E_n = E(\theta_n - z_n)^2$ (avec $z_n = E(X_n)$) entre les trajectoires du vrai système z_n et celles de l'algorithme θ_n . En considérant, par exemple, que la moyenne évolue selon l'hypermodèle déterministe suivant :

$$z_n = z_{n-1} + \xi \text{ avec } \xi \text{ constante et } \xi \ll \sigma$$

l'ensemble {système, algorithme} se modélise par le couple d'équations :

$$\begin{aligned} z_n &= z_{n-1} + \xi \\ \theta_n &= \theta_{n-1} + \mu(X_n - \theta_{n-1}). \end{aligned}$$

Les données X_n indépendantes ayant une variance supposée constante $E(X_n - z_n)^2 = \sigma^2$, on désire suivre l'évolution de :

- l'erreur quadratique moyenne (EQM) du paramètre estimé $E_n = E(\theta_n - z_n)^2$,
- le biais du paramètre estimé $B_n = E(\theta_n) - z_n$,

il vient :

$$\begin{aligned} E_n &= (1 - \mu)^2 E_{n-1} - 2\xi(1 - \mu)^2 B_{n-1} + \xi^2(1 - \mu)^2 + \mu^2\sigma^2 \\ B_n &= (1 - \mu) B_{n-1} + \xi(\mu - 1). \end{aligned}$$

Soit, sous forme matricielle :

$$\begin{pmatrix} E_n \\ B_n \end{pmatrix} = \begin{pmatrix} \xi^2(1 - \mu)^2 + \mu^2\sigma^2 \\ \xi(\mu - 1) \end{pmatrix} + \begin{pmatrix} (1 - \mu)^2 & -2\xi(1 - \mu)^2 \\ 0 & 1 - \mu \end{pmatrix} \begin{pmatrix} E_{n-1} \\ B_{n-1} \end{pmatrix}$$

La matrice $\begin{pmatrix} (1 - \mu)^2 & -2\xi(1 - \mu)^2 \\ 0 & 1 - \mu \end{pmatrix}$ admet $1 - \mu$ et $(1 - \mu)^2$ pour valeurs propres ; son rayon spectral $\rho = 1 - \mu$ est inférieur à 1. Dans ces conditions, on peut montrer que la suite vectorielle précédente converge et que sa limite vaut :

$$\lim_{n \rightarrow \infty} \begin{pmatrix} E_n \\ B_n \end{pmatrix} = \begin{pmatrix} E \\ B \end{pmatrix} = \begin{pmatrix} \mu \frac{\sigma^2}{2 - \mu} + \frac{\xi^2}{\mu^2} (1 - \mu)^2 \\ (\mu - 1) \xi / \mu \end{pmatrix}.$$

Soit :

$$E = \mu \frac{\sigma^2}{2 - \mu} + \frac{\xi^2}{\mu^2} (1 - \mu)^2. \quad (3.22)$$

Puisque l'EQM E , le biais B et la variance asymptotique V sont reliées par $E = V + B^2$, la variance asymptotique est donnée par $V = \mu \frac{\sigma^2}{2-\mu}$. Remarquons que la décomposition asymptotique de l'EQM pour $\mu \rightarrow 0$ et $n \rightarrow +\infty$ s'écrit :

$$\lim_{n \rightarrow \infty} E(\theta_n - z_n)^2 \sim \mu \frac{\sigma^2}{2} + \frac{\xi^2}{\mu^2} \quad \text{pour } \mu \rightarrow 0. \quad (3.23)$$

Les résultats théoriques seront toujours énoncés en régime asymptotique ($n \rightarrow \infty$) sous l'hypothèse $\mu \rightarrow 0$. Ceci signifie simplement que les résultats ne sont rigoureusement valides que pour μ petit et donc pour des non-stationnarités lentes.

L'analyse de l'estimateur de la moyenne dans le cas où celle-ci évolue, montre la nécessité d'un compromis entre les deux exigences contradictoires suivantes :

- bonne capacité de poursuite des non-stationnarités qui impose que μ soit grand,
- faibles fluctuations, ce qui nécessite un μ petit.

Un compromis classique entre ces deux exigences consiste à minimiser l'EQM asymptotique (3.23). Le pas μ qui optimise ce compromis (obtenu par dérivation) s'écrit :

$$\mu_* = \left(\frac{4\xi^2}{\sigma^2} \right)^{1/3}.$$

Le fonctionnement de l'algorithme (3.16) en contexte non stationnaire est illustré par la figure

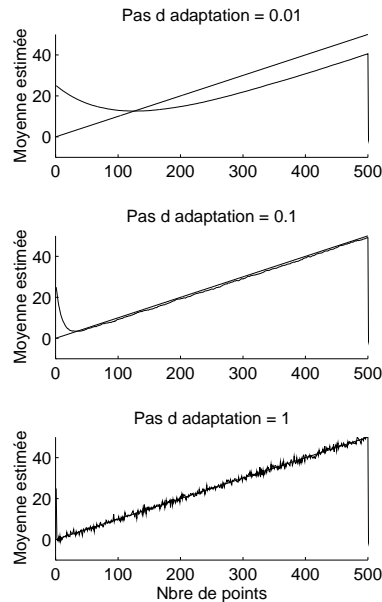


Figure 3.5. *Estimateur adaptatif de la moyenne en non stationnaire.*

[Fig.3.5]. Sur cette figure, on peut observer les trajectoires de l'estimateur θ_n pour différentes valeurs du pas d'adaptation μ . Les conditions de la simulation sont : calcul sur 500 points, les observations sont une suite de v.a. indépendantes de variance égale à un dont la moyenne dérive de $\xi = 0.1$ d'un point au suivant, la valeur initiale de l'estimateur est 25, les différentes valeurs de μ utilisées sont notées sur la figure. Lorsque μ est trop petit l'estimateur est clairement en retard sur la rampe ; lorsque μ est trop grand les fluctuations sont nettement visibles. Une valeur intermédiaire de μ révèle l'existence d'une valeur optimale pour μ . On vérifie de cette manière l'augmentation des fluctuations ainsi que la diminution du biais lorsque μ augmente.

Ce compromis biais-variance en contexte non stationnaire peut être mis en évidence expérimentalement en calculant la valeur empirique de l'erreur quadratique moyenne d'estimation. Cette EQM est calculée en ne retenant que les valeurs prises par l'estimateur une fois la période de convergence initiale terminée. La figure 3.6 représente l'EQM asymptotique théorique donnée par la formule (3.23) (tirets), l'EQM asymptotique théorique donnée par la formule (3.22) (pointillés) et l'EQM calculée expérimentalement (trait plein). Pour la courbe expérimentale, les conditions de la simulation sont celles de la figure 3.5 où ici l'EQM est calculée sur les 500 derniers points d'un calcul mené sur 1000 points. On constate l'excellent accord entre les résultats théoriques et expérimentaux. Notons dès maintenant que la formule (3.23) est celle que donnent les outils théoriques présentés maintenant. Ceci est cohérent puisque ces outils sont fondés sur l'hypothèse que le pas d'adaptation tend vers zéro.

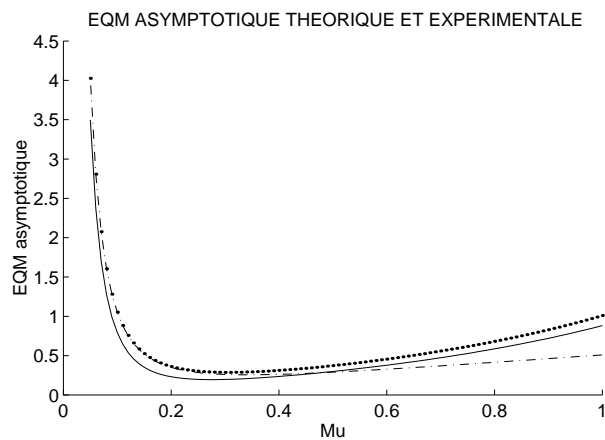


Figure 3.6. EQM asymptotique.

3.4.3. Méthode générale : diffusions gaussiennes et ODE

Cette phase : synthèse des deux précédentes (ODE et diffusion gaussienne) permet l'étude des algorithmes adaptatifs dans le cadre où ils sont précisément utilisés. Les

connaissances *a priori* de l'utilisateur quant à l'évolution du système à poursuivre z_n sont prises en compte grâce à l'équation d'évolution :

$$z_n = z_{n-1} + K(z_{n-1}, \zeta_n)$$

dans laquelle l'état ζ_n est markovien. Cet hypermodèle est naturellement indépendant de l'algorithme. Finalement, l'ensemble {système, algorithme} est associé au couple d'équations suivant :

$$z_n = z_{n-1} + K(z_{n-1}, \zeta_n) \quad (3.24)$$

$$\theta_n = \theta_{n-1} + \Gamma H(\theta_{n-1}, z_{n-1}; X_n) \quad (3.25)$$

où selon [Benveniste 87], l'algorithme est défini par un *gain matriciel* constant Γ . On pourra ainsi chercher à optimiser soit le pas scalaire μ , soit la direction de déplacement Γ elle-même, et comparer ainsi les performances en poursuite des méthodes de gradient et des méthodes newtoniennes.

REMARQUE.— Pour associer des ODE, il faut définir l'échelle de temps de ces algorithmes, et on sera amené à préciser deux échelles avec $K = \gamma K'$ et $\Gamma = \mu \Gamma'$, avec γ échelle de temps de variation du paramètre, et μ échelle de temps de variation de l'algorithme de poursuite. La notion de variation lente, permettant la poursuite, est donc liée à un rapport entre ces deux échelles, et celui-ci dépend de la nature du modèle de non-stationnarité. Dans les résultats énoncés ci-dessous, il y a donc lieu de renormaliser éventuellement selon l'échelle μ de l'algorithme, incluse implicitement dans le gain matriciel Γ .

Notations

– $P_{\theta, z}$ loi asymptotique pour laquelle X_n (chaîne contrôlée par θ, z) est stationnaire,

– P_z loi asymptotique pour laquelle ζ_n (chaîne contrôlée par z) est stationnaire,

– $k(z) = \lim_{n \rightarrow \infty} E_z K(z, \zeta_n)$ où E_z désigne l'espérance relative à la distribution P_z ,

– $h(\theta, z) = \lim_{n \rightarrow \infty} E_{\theta, z} H(\theta, z, X_n)$ où $E_{\theta, z}$ désigne l'espérance relative à $P_{\theta, z}$,

– $G(\theta, z) = \sum_{-\infty}^{+\infty} \text{cov}_{\theta, z} [H(\theta, z, X_n), H(\theta, z, X_0)]$ où $\text{cov}_{\theta, z}$ est la covariance relative à $P_{\theta, z}$,

– $Q(z) = \sum_{-\infty}^{+\infty} \text{cov}_z [K(\theta, \zeta_n), K(\theta, \zeta_0)]$ où cov_z est la covariance relative à P_z .

Les hypothèses générales sont les suivantes

- L'état ζ_n est à représentation markovienne et X_n est maintenant à représentation markovienne contrôlée par θ et z ,
- Les champs moyens h et k sont suffisamment réguliers (Lipschitziens),
- Le vrai système est identifiable par l'algorithme i.e. $h(\theta, z) = 0 \iff \theta = z$,
- Pour tout z utile au fonctionnement de l'algorithme, l'ensemble des gains admissibles, i.e. tels que toutes les valeurs propres de Γh_θ soient de partie réelle négative, est non vide.

Etant donné un modèle d'évolution du paramètre à estimer θ , les résultats qui suivent permettent d'en déduire les paramètres optimaux correspondants pour l'algorithme. Le critère de distance système-algorithme utilisé est celui de l'EQM : minimum de $E \|\theta_n - z_n\|^2$. L'obtention de ces résultats (cf. [Benveniste 87]) résulte de l'application directe à l'hypermodèle des méthodes ODE et approximation-diffusion gaussienne étudiées précédemment. On ne fait ici que donner les résultats utilisables.

Hypermodèle sans dérive. On qualifie de sans dérive un hypermodèle dont la moyenne du paramètre à estimer reste constante au cours du temps. Ce type d'évolution purement aléatoire comprend en particulier les modèles de marche aléatoire. Pour ces modèles, l'EQM est donnée par :

$$E \|\theta_n - z_n\|^2 = \text{Tr}(C_\theta)$$

où C_θ est solution de l'équation :

$$(\Gamma D_h(\theta))C_\theta + C_\theta (\Gamma D_h(\theta))^T + \Gamma G \Gamma^T + Q = 0. \quad (3.26)$$

Les valeurs optimales Γ_* du gain Γ et $C_{\theta*}$ de la covariance du paramètre estimé satisfont :

$$C_{\theta*} \Delta^{-1} C_{\theta*} = Q$$

et

$$\Gamma_* = -C_{\theta*} D_h(\theta)^T G^{-1} = Q^{1/2} G^{-1/2}$$

où $\Delta \equiv D_h(\theta)^{-1} G D_h(\theta)^{-T}$.

Hypermodèle avec dérive. Ils correspondent aux situations dans lesquelles la moyenne de z_n change au cours du temps. L'exemple type est celui de la dérive purement déterministe. L'EQM admet la décomposition :

$$E \|\theta_n - z_n\|^2 = \left\| (\Gamma D_h(\theta))^{-1} k \right\|^2 + \text{Tr}(C_\theta)$$

où C_θ est solution de :

$$(\Gamma D_h(\theta)) C_\theta + C_\theta (\Gamma D_h(\theta))^T + \Gamma G \Gamma^T = 0$$

Critère de qualité Δ .

La quantité Δ définie précédemment a été introduite par A. Benveniste, M. Métivier et P. Priouret dans [Benveniste 87]. Elle chiffre la capacité de poursuite d'un filtre adaptatif grâce au résultat suivant.

Sous la contrainte $\Gamma D_h(\theta) = -\Gamma_0$, où Γ_0 est un gain fixé et asymptotiquement stable, l'EQM admet la décomposition asymptotique suivante :

$$\mathbb{E} \|\theta_n - z_n\|^2 = \gamma^\alpha \Psi(\mu_0, \Delta)$$

avec $\alpha = 1$ pour les hypermodèles sans dérive et $\alpha = 2/3$ pour les hypermodèles avec dérive. γ mesure la vitesse d'évolution du vrai système. Dans cette expression Ψ est une fonction *croissante de* Δ . Plus Δ est petit, meilleur est la capacité de poursuite de l'algorithme. Ce résultat est d'une grande importance pratique puisqu'il permet de classer les algorithmes adaptatifs indépendamment de tout hypermodèle particulier.

EXEMPLE 13 : ODE ET DIFFUSION POUR L'ESTIMATEUR DE MOYENNE. Ces résultats permettent en particulier de retrouver la décomposition biais-variance pour l'estimateur de moyenne. Pour cet exemple, nous avons : $D_h(\theta_*) = -1$, $G(\theta_*) = \sigma^2$ (on observe une suite de v.a. blanche de variance σ^2) et $k = \xi$ (dérive purement déterministe). C_θ est solution de $(\Gamma D_h(\theta)) C_\theta + C_\theta (\Gamma D_h(\theta))^T + \Gamma G \Gamma^T = 0$ c'est-à-dire : $-2\mu C_\theta + \mu^2 \sigma^2 = 0$ d'où $C_\theta = \mu \sigma^2 / 2$. La formule :

$$\mathbb{E} \|\theta_n - z_n\|^2 = \|(\Gamma D_h(\theta))^{-1} k\|^2 + \text{Tr}(C_\theta)$$

devient ainsi :

$$\mathbb{E} \|\theta_n - z_n\|^2 = \xi^2 / \mu^2 + \mu \sigma^2 / 2$$

Il faut toujours prendre garde au fait que le résultat obtenu n'est rigoureusement valide que pour $\mu \rightarrow 0$. Ce point est clair si l'on se rappelle les résultats du calcul direct (exemple 12). La normalité asymptotique de l'estimateur se vérifie expérimentalement en traçant l'histogramme des valeurs prises par l'erreur d'estimation. La figure 3.7 représente la densité de probabilité théorique et la densité estimée. Ici encore, l'accord entre les deux est excellent. Les conditions de la simulation sont : calcul sur 10 000 points (histogramme calculé sur les 9 000 derniers), les observations sont une suite de v.a. indépendantes, centrées, de variance égale à un, la valeur initiale de l'estimateur est 4 et $\mu = 0.1$.

3.4.4. *Le cas standard : LMS et RLS, exemples*

On applique ici la méthode générale sur le cas d'une régression linéaire variable, présenté en (3.20). On se place dans l'hypothèse d'une marche aléatoire. On compare dans ce contexte les performances des algorithmes LMS et RLS. On présente ensuite une méthode à pas adaptatif, permettant de converger automatiquement vers le pas optimal pour la poursuite. Dans ce contexte de régression linéaire, on a avec $\Sigma = \mathbb{E}(\varphi_n \varphi_n^T)$:

$$h(\theta, z) = \Sigma(z - \theta) \quad D_h(\theta) = -\Sigma G(z, z) = \sigma^2 \Sigma$$

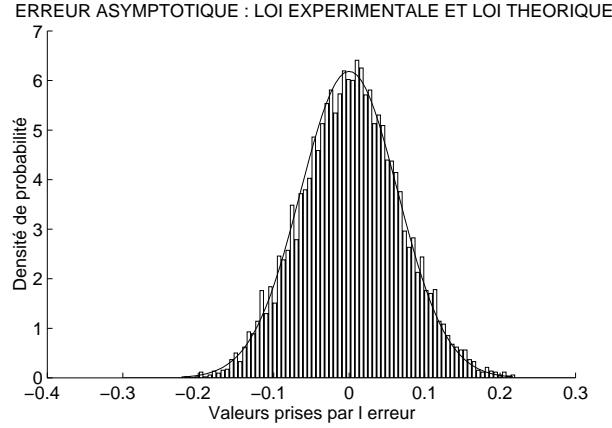


Figure 3.7. Densité de probabilité de l'erreur.

Comparaison LMS et RLS

Pour le LMS, on a $\Gamma = \mu I$ et l'équation de Liapounoff (3.26) devient

$$\Sigma C_\theta + C_\theta \Sigma = \sigma^2 \mu \Sigma + \frac{Q}{\mu}. \quad (3.27)$$

On en déduit l'EQM $J_{\text{lms}} = \text{cov}(\varphi_n^T(\theta_n - z_n))$. En admettant en régime stationnaire une indépendance entre $\theta_n - z_n$ et φ_n , on a

$$J_{\text{lms}} = \text{E}(\varphi_n^T C_\theta \varphi_n) = \text{Tr} \text{E}(C_\theta^{1/2} \varphi_n^T \varphi_n C_\theta^{1/2}) = \text{Tr}(C_\theta^{1/2} \Sigma C_\theta^{1/2}) = \text{Tr}(C_\theta \Sigma)$$

et par utilisation de (3.27)

$$J_{\text{lms}} = \mu \frac{\sigma^2}{2} \text{Tr}(\Sigma) + \frac{1}{2\mu} \text{Tr}(Q)$$

L'optimisation du pas donne, en fonction du degré de non-stationnarité (3.21)

$$\mu_{\text{opt}} = \frac{v}{\text{Tr}(\Sigma)} \quad J_{\text{lms}}^* = d\sigma_u \sigma \sqrt{M}.$$

Pour le RLS, il correspond à un gain matriciel variable K_n fluctuant autour de Σ^{-1} de l'ordre de $\sqrt{\mu}$. Multipliée par μ , la fluctuation devient négligeable en asymptotique à $\mu \rightarrow 0$, on va donc approcher RLS par l'algorithme à gain matriciel fixe $\Gamma = \Sigma^{-1}$ et pas fixe $\mu = 1 - \lambda$. l'équation de Liapounoff (3.26) donne directement la solution

$$C_\theta = \frac{Q}{2\mu} + \mu \frac{\sigma^2}{2} \Sigma^{-1}.$$

On en déduit l'EQM

$$J_{\text{rls}} = \mu \frac{\sigma^2}{2} M + \frac{1}{2\mu} \text{Tr}(Q\Sigma).$$

L'optimisation du pas donne, en fonction de $s = \text{Tr}(Q\Sigma)$

$$\mu_{\text{opt}} = \left(\frac{s^2}{M\sigma^2} \right)^{1/2} \quad J_{\text{rls}}^* = s\sigma\sqrt{M}.$$

Exemple : On prend $M = 2$, $\Sigma = \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix}$, $Q = \frac{d^2}{2} \begin{pmatrix} 1 & q \\ q & 1 \end{pmatrix}$. On a alors le rapport des EQM optimales

$$\frac{J_{\text{rls}}^*}{J_{\text{lms}}^*} = \sqrt{1 + aq}.$$

On en déduit que

- si $aq > 0$, LMS est meilleur que RLS
- si $aq < 0$, RLS est meilleur que LMS
- si $aq = 0$ (cas où u_n est blanc, ou fluctuations des coefficients du filtre indépendantes), LMS et RLS ont des performances comparables.

Les résultats simulés du tableau ci-contre confirment la validité de ces estimations théoriques. Les paramètres, pour le critère d'erreur

$$D(\mu) = E(|\theta_n - z_n|^2),$$

sont : $\sigma^2 = 0.2$; $a = 0.8$; $d^2 = 0.05$; $q = -0.96$; $\frac{D_{\text{rls}}^*}{D_{\text{lms}}^*} = \frac{1}{\sqrt{1-aq}} = 0.75$ et le rapport estimé sur les données est de 0.745.

D(μ) LMS		D(μ) RLS	
Théorique	Estimé	Théorique	Estimé
0.4449	0.3166	0.1932	0.1379
0.2308	0.1736	0.1078	0.0902
0.1632	0.1269	0.0843	0.0765
0.1322	0.1083	0.0763	0.0731
0.1158	0.1007	0.0745	0.0738
0.1068	0.0985	0.0758	0.0766
0.1019	0.0994	0.0789	0.0806
0.0996	0.1024	0.0830	0.0856

CONCLUSION.- Pour la poursuite d'un filtre variable selon une marche aléatoire, les performances optimales à petit pas (sous la condition de degré de stationnarité petit) du RLS peuvent être meilleures ou moins bonnes que celles du LMS, selon la statistique des signaux, l'algorithme optimal en poursuite n'est pas forcément donné par le gain de Newton Σ^{-1} du RLS. Ce résultat est qualitativement intéressant : il montre que dans ce cas de poursuite, le choix d'un algorithme RLS de type newtonien n'apporte pas forcément de gain de performances en termes de poursuite. La supériorité des gains de Newton en transitoire n'est plus vraie en poursuite. On trouvera d'autres exemples détaillés de comparaison d'algorithmes dans diverses situations non stationnaires dans la synthèse [Macchi 89], et un traitement du cas de la poursuite d'une

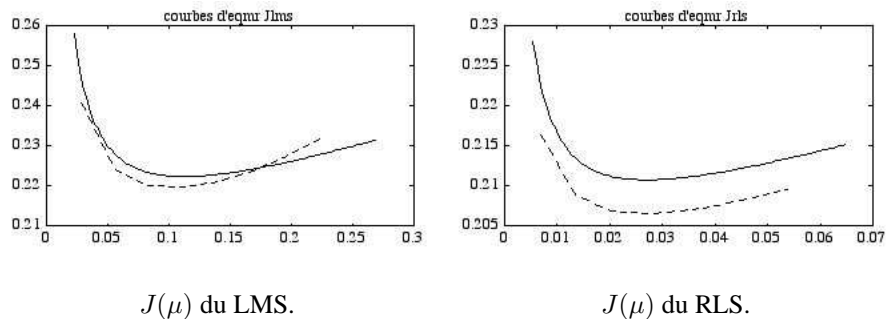


Figure 3.8. Courbes théorique et pratique EQM $J(\mu)$ du LMS et du RLS.

dérive de fréquence (signal "Chirp") dans [Bershad 80, 89], [Macchi 91], [Michaut 92].

3.5. Application : Méthodes d'approximation de sous-espaces

Dans cette section, on reprend et développe les méthodes adaptatives d'approximation de sous-espaces, présentées rapidement en section (3.1.4). On traite d'abord la construction des algorithmes adaptatifs en se limitant aux algorithmes d'approximations stochastiques de la forme (3.8). Puis on s'attachera au problème spécifique de l'étude de la convergence dans le cas où la convergence vers une base particulière de l'espace invariant recherché n'est pas assurée.

3.5.1. Construction des algorithmes

Il existe une abondante littérature permettant de répondre aux deux problèmes décrits en section (3.1.4) et ces algorithmes adaptatifs intègrent souvent une étape de normalisation ou d'orthonormalisation qui leur donne une structure plus générale que la forme (3.8). Ces algorithmes sont issus soit de méthodes d'analyse numérique et principalement de la méthode de la *puissance itérée* (dite aussi "simultaneous iteration method" [Rutishauser 69]) soit d'*optimisation de critères* $J(W)$ sous ou sans contraintes. Nous allons introduire quelques uns d'entre eux en se limitant aux algorithmes de la forme (3.8).

Ainsi de nombreuses méthodes de la littérature traitement de signal ou neuronale sont issues de différentes approximations de la méthode d'analyse numérique "simultaneous iteration method" [Rutishauser 69] suivante :

$$W'_n = W_{n-1} + R_n W_{n-1} \Gamma_n \quad (3.28)$$

$$W_n = W'_n S_n^{-1} \quad (3.29)$$

où $W_n = (w_{n,1}, \dots, w_{n,r}) \in \mathbb{R}^{m \times r}$ est une matrice dont les colonnes sont orthonormées et approximent les r vecteurs propres dominants de R_x . Dans (3.28), la matrice Γ_n est la matrice $r \times r$ diagonale usuelle de gains. Nous supposons que $\Gamma_n = \mu_n I_r$ ou plus généralement $\Gamma_n = \mu_n \text{diag}(1, \alpha_2, \dots, \alpha_r)$ avec $\alpha_i > 0$. La matrice R_n de (3.28) est une estimée de la matrice de covariance R_x , elle est souvent réduite à l'estimée instantanée $x_n x_n^T$. Quelquefois, on utilise l'estimée lissée $R_n = R_{n-1} + \alpha \mu_n (x_n x_n^T - R_{n-1})$. Dans (3.29), S_n est une matrice dépendant de W'_n qui orthonormalise les colonnes de W'_n . D'où W_n a ses colonnes orthonormées pour tout n . Dépendant du choix de la matrice S_n , différentes versions d'algorithmes d'approximations stochastiques peuvent être obtenues.

Ainsi par exemple puisque W_{n-1} a des colonnes orthonormées, alors pour $\mu_n \ll 1$, les colonnes de W'_n dans (3.28) seront linéairement indépendantes quoique non orthonormées. Alors $W_n'^T W'_n$ sera définie positive et W_n aura des colonnes orthonormées si $S_n = (W_n'^T W'_n)^{1/2}$. L'algorithme d'approximation stochastique appelé *Subspace Network Learning* (SNL) est obtenu en utilisant un développement du premier ordre de S_n^{-1} et en omettant le terme $O(\mu_n^2)$ grâce à l'hypothèse $\mu_n \ll 1$. L'algorithme s'écrit alors sous la forme suivante :

$$W_n = W_{n-1} + \mu_n [I_m - W_{n-1} W_{n-1}^T] x_n x_n^T W_{n-1}, \quad (3.30)$$

L'étude de cet algorithme a été faite dans [Williams 85] et dans [Yan 94], où il a été démontré que les solutions de l'ODE associée (3.12) ne convergeaient que vers une base orthonormée du sous espace propre dominant de R_x (ces vecteurs pouvant se déplacer dans cet espace). C'est à dire que $\lim_{t \rightarrow \infty} W_t^T W_t = I_r$ et $\lim_{t \rightarrow \infty} W_t W_t^T = \sum_{i=1}^r v_i v_i^T$. Cet algorithme appartient ainsi à la famille (2).

Un autre point de départ pour dériver un algorithme d'approximation stochastique à partir de (3.28) et de (3.29) est de considérer que la matrice S_n réalise une orthogonalisation de Gram-Schmidt des colonnes de W'_n . L'algorithme appelé *Stochastic Gradient Ascent* (SGA) est obtenu en faisant un développement au premier ordre de W_n en supposant $\mu_n \ll 1$, en omettant le terme $O(\mu_n^2)$ dans ce développement, on obtient pour $k = 1, \dots, r$:

$$w_{n,k} = w_{n-1,k} + \mu_n [I_m - w_{n-1,k} w_{n-1,k}^T - 2 \sum_{i=1}^{k-1} w_{n-1,i} w_{n-1,i}^T] x_n x_n^T w_{n-1,k} \quad (3.31)$$

Une extension [Oja 85] de cet algorithme est obtenue si $\Gamma_n = \mu_n \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_r)$ avec $\alpha_1 = 1$ et $\alpha_i > 0$. Soit pour $k = 1, \dots, r$:

$$w_{n,k} = w_{n-1,k} + \alpha_k \mu_n [I_m - w_{n-1,k} w_{n-1,k}^T - \sum_{i=1}^{k-1} (1 + \frac{\alpha_i}{\alpha_k}) w_{n-1,i} w_{n-1,i}^T] x_n x_n^T w_{n-1,k}.$$

Cet algorithme par le choix des valeurs α_i permet d'améliorer le compromis vitesse de convergence/précision. L'algorithme *Generalized Hebbian Algorithm* (GHA) est

dérivé à partir de l'algorithme (3.30) en remplaçant la matrice $W_{n-1}^T x_n x_n^T W_{n-1}$ de l'algorithme SNL par sa diagonale et la partie située au dessus de celle-ci :

$$W_n = W_{n-1} + \mu_n [x_n x_n^T W_{n-1} - W_{n-1} \text{upper}(W_{n-1}^T x_n x_n^T W_{n-1})]$$

dans lequel l'opérateur matriciel "upper" assigne la valeur 0 à tous les termes situés sous la diagonale principale. Ecrit sous forme colonnes, cet algorithme est similaire à l'algorithme SGA (3.31), la différence provenant seulement du terme 2 dans la somme :

$$w_{n,k} = w_{n-1,k} + \mu_n [I_m - \sum_{i=1}^k w_{n-1,i} w_{n-1,i}^T] x_n x_n^T w_{n-1,k} \quad \text{pour } k = 1, \dots, r.$$

Oja *et al* [Oja 92-a] proposa un algorithme dénoté *Weighted Subspace Algorithm* (WSA), qui est similaire à l'algorithme SNL, à part les paramètres scalaires β_1, \dots, β_r :

$$w_{n,k} = w_{n-1,k} + \mu_n [I_m - \sum_{i=1}^r \frac{\beta_k}{\beta_i} w_{n-1,i} w_{n-1,i}^T] x_n x_n^T w_{n-1,k} \quad \text{pour } k = 1, \dots, r,$$

avec $0 < \beta_1 < \dots < \beta_r$. Si $\beta_i = 1$ pour tout i , cet algorithme se ramène à l'algorithme SNL.

Il a été démontré par Oja que les seuls points asymptotiquement localement stables de l'ODE associée aux algorithmes SGA, GHA et WSA étaient les vecteurs propres $\pm v_1, \dots, \pm v_r$. D'autre part, nous notons que le premier vecteur ($k = 1$) estimé par les algorithmes SGA et GHA et le vecteur ($r = k = 1$) estimé par les algorithmes SNL et WSA coïncident avec l'algorithme *Constrained Hebbian learning rule* ou célèbre neurone de base introduit par [Oja 82]

$$w_{n,1} = w_{n-1,1} + \mu_n [I_m - w_{n-1,1} w_{n-1,1}^T] x_n x_n^T w_{n-1,1}.$$

Cet algorithme coïncide avec de très nombreux algorithmes issues selon des approches empiriques différentes. Par exemple il coïncide avec l'algorithme *Direct Adaptive Subspace Estimator*, proposé par Riou *et al* [Riou 96] pour $k = 1$ qui s'écrit pour $k = 1, \dots, r$:

$$w_{n,k} = w_{n-1,k} + \mu_n [I_m - (I_m - \sum_{i=1}^{k-1} w_{n-1,i} w_{n-1,i}^T) w_{n-1,k} w_{n-1,k}^T] x_n x_n^T w_{n-1,k},$$

dont les points stationnaires de l'ODE associée sont les ensembles de r vecteurs propres normalisés de R_x , mais ces points ne sont asymptotiquement localement stable pour cet ODE que pour $r = 1$. Pour remédier à cette difficulté, les auteurs proposent de normaliser à chaque étape les vecteurs $w_{n,k}$.

Curieusement des algorithmes de type minorants peuvent également être générés pas cette même approche. Par exemple [Xu 92] introduisirent l'algorithme *Optimal Fitting Analyzer* (OFA) en modifiant l'algorithme SGA. Cet algorithme s'écrit :

$$\begin{aligned}
 w_{n,k} &= w_{n-1,k} + \mu_n [I_m - x_n x_n^T + w_{n-1,k} w_{n-1,k}^T x_n x_n^T - w_{n-1,k} w_{n-1,k}^T] \\
 &- \beta \sum_{i=k+1}^n w_{n-1,i} w_{n-1,i}^T x_n x_n^T w_{n-1,k} \quad \text{pour } k = n - r + 1, \dots, n.
 \end{aligned}$$

[Oja 92-b] démontra que sous la condition que les valeurs propres soient distinctes et que $\lambda_{n-r+1} < 1$ $\beta > \frac{\lambda_{n-r+1}}{\lambda_n} - 1$, l'ODE associée admet comme seuls points localement asymptotiquement stable les vecteurs propres $\pm v_{m-r+1}, \dots, \pm v_m$. Il est à noter que l'idée de base derrière ce passage d'un algorithme de composantes principales à un algorithme de composantes mineures est que si l'on connaît un majorant α des valeurs propres de R_x , les r vecteurs propres associées aux r plus petites valeurs propres de R_x sont les r vecteurs propres associées aux r plus grandes valeurs propres de $\alpha I_m - R_x$. Cette idée est à la base de la conception de nombreux algorithmes d'approximations stochastiques de composantes mineures déduits d'approximations stochastiques de composantes principales.

A titre d'exemples d'algorithmes d'approximations stochastiques issus d'optimisations de critères $J(W)$ sous ou sans contraintes, citons ceux basés sur le *quotient de Rayleigh* et ceux basés sur le projecteur orthogonal.

A partir de la notion de quotient de Rayleigh, les optimisations $\max_{W^T W = I_r} \sum_{i=1}^r w_i^T R_x w_i$ [resp. $\min_{W^T W = I_r} \sum_{i=1}^r w_i^T R_x w_i$] ont pour solution toute base orthonormée de l'espace vectoriel engendré par les vecteurs propres associés aux r plus grandes [resp. petites] valeurs propres de R_x . Cela entraîne l'algorithme du gradient stochastique suivant [Yang 88] :

$$\begin{aligned}
 W'_n &= W_{n-1} \pm \mu_n x_n x_n^T W_{n-1} \quad + \text{ pour estimer un espace majorant} \\
 &\quad \text{[resp. - pour estimer un espace minorant]} \\
 W_n &= \text{Orthonormalisation des colonnes de } W'_n,
 \end{aligned}$$

dont les points stationnaires globalement asymptotiquement stables sont constitués en général de l'ensemble de ces bases orthonormées. Notons que différentes approches peuvent être utilisées pour réaliser l'étape d'orthonormalisation : méthodes exactes (QR, Gram-Schmidt,...) ou approchées et que dans le cas de l'estimation d'un espace

majorant, cet algorithme peut être interprété comme un cas particulier de l'algorithme (3.28)(3.29) issu de la puissance itérée.

A partir de la définition du quotient de Rayleigh

$$v_1 = \arg \max_{\|w\|^2=1} w^T R_x w$$

et

$$v_i = \arg \max_{\|w\|^2=1, w \perp \{w_1, \dots, w_{i-1}\}} w^T R_x w \quad \text{pour } i = 2, \dots, r,$$

[resp. $v_m = \min_{\|w\|^2=1} w^T R_x w$ et $v_i = \min_{\|w\|^2=1, w \perp \{w_m, \dots, w_{i+1}\}} w^T R_x w$ pour $i = m-1, \dots, m-r+1$], et en remplaçant les contraintes par une paramétrisation de Givens.

$$w_1 = Q_1 \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, w_2 = Q_1 \begin{bmatrix} Q_2 \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix} \end{bmatrix}, \dots, w_r = Q_1 \begin{bmatrix} Q_2 \begin{bmatrix} Q_r \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \\ 0 \\ 0 \end{bmatrix} \end{bmatrix}$$

où Q_i sont les matrices orthogonales suivantes d'ordre $m-i+1$:

$$Q_i = U_{i,1} \dots U_{i,j} \dots U_{i,m-i} \quad \text{avec} \quad U_{i,j} \equiv \begin{bmatrix} I_{j-1} & 0 & 0 & 0 \\ 0 & -\sin \theta_{i,j} & \cos \theta_{i,j} & 0 \\ 0 & \cos \theta_{i,j} & \sin \theta_{i,j} & 0 \\ 0 & 0 & 0 & I_{n-i-j} \end{bmatrix}$$

et où $\theta_{i,j} \in]-\frac{\pi}{2}, +\frac{\pi}{2}]$, l'étape d'orthonormalisation de l'algorithme du gradient stochastique peut être supprimée. Dans le but de simplifier l'algorithme ainsi obtenu, une procédure de déflation a été proposée par [Regalia 90]. La première maximisation [resp. minimisation] est réalisée par un algorithme classique de gradient stochastique pour le paramètre $\theta_{1,1}, \dots, \theta_{1,m-1}$, tandis que les maximisations [resp. minimisations] suivantes sont réalisées grâce à des algorithmes de gradient stochastique par rapport aux paramètres $\theta_{i,1}, \dots, \theta_{i,m-i}$ dans lesquels les paramètres précédents $\theta_{l,1}(n-1), \dots, \theta_{l,m-l}(n-1)$ pour $l = 1, \dots, i-1$ sont injectés à partir des $i-1$ algorithmes précédents. Cette procédure de déflation entraîne un couplage d'algorithmes de gradients stochastiques. On aboutit à un algorithme d'approximation stochastique dont les points globalement asymptotiquement stables sont les vecteurs propres dominants v_1, \dots, v_r , (resp. minorants v_{m-r+1}, \dots, v_m) [Delmas 98-b].

Considérons maintenant le critère $J(W) \equiv E(\|x - WW^T x\|^2)$ où W est une matrice $m \times r$. Sachant que W est un point stationnaire de $J(W)$ si et seulement si $W = UQ$ où les colonnes de la matrice $m \times r$ U contiennent r vecteurs propres

arbitraires de R_x et où Q est une matrice $m \times m$ orthogonale arbitraire et que tous ces points stationnaires sont des points selles de $J(W)$ sauf les points $W = UQ$ où $U = [v_1, \dots, v_r]$, pour lesquels $J(W)$ atteint son minimum global, différents algorithmes d'approximations stochastiques ont été proposés par [Yang 95] à partir de la minimisation de ce critère. De manière très classique a été proposé un algorithme du gradient stochastique

$$W_n = W_{n-1} + \mu_n [2x_n x_n^T - x_n x_n^T W_{n-1} W_{n-1}^T - W_{n-1} W_{n-1}^T x_n x_n^T] W_{n-1}, \quad (3.32)$$

dans lequel le terme entre parenthèse est la symétrisation du terme associé de l'algorithme SNL (3.30). L'algorithme (3.30) peut d'ailleurs être obtenu à partir de (3.32) en y faisant l'approximation $W_{n-1}^T W_{n-1} = I_r$. D'après ce qui précède, l'ODE associée à cet algorithme admet cette fois-ci comme points globalement asymptotiquement stables, les points $W = UQ$ où $U = [v_1, \dots, v_r]$, contrairement à l'algorithme SNL pour lequel ces points n'étaient que localement asymptotiquement stables.

A partir de ce critère $J(W)$, [Yang 95] introduisit une approche de type RLS dans laquelle W_n est solution du minimum du critère classique moyenne pondérée avec facteur d'oubli :

$$\sum_{i=1}^n \beta^{n-i} \|x_i - W_n W_n^T x_i\|^2$$

dans laquelle le terme $W_n^T x_i$ est approximé par $y_i \equiv W_{i-1}^T x_i$, donc connu pour $i = 1, \dots, n$. Il en résulte une fonction de coût modifiée qui est maintenant quadratique en W_n à chaque instant n :

$$J(W_n) = \sum_{i=1}^n \beta^{n-i} \|x_i - W_n y_i\|^2,$$

d'où le nom *projection approximative subspace tracking* (PAST) attribué à cet algorithme. La minimisation de ce critère se ramène alors à un problème classique de moindre carré récursif avec facteur d'oubli vu à la section 2.2.2. Notons qu'avec cette approximation les colonnes de W_n ne sont plus qu'approximativement orthonormées. Par conception, cet algorithme ne donne accès qu'à une estimation du sous espace dominant. Si l'on désire une estimation des vecteurs propres de ce sous espace, on peut utiliser la variante PASTd de cet algorithme. Elle consiste à utiliser une méthode de déflation, c'est à dire à appliquer l'algorithme PAST pour $r = 1$ pour générer $w_{n,1}$ puis à réutiliser l'algorithme PAST pour $r = 1$ en remplaçant x_n par $x_n - w_{1,n} w_{1,n}^T x_n$ pour la génération de $w_{n,2}$ et ainsi de suite (car $E((x - v_1 v_1^T x)(x - v_1 v_1^T x)^T) = R_x - \lambda_1 v_1 v_1^T$). Pour $\beta = 1$, Il a été démontré dans [Yang 96] que les algorithmes PAST and PASTd prennent la structure générale (3.8) avec $\mu_n = \frac{1}{n}$ et que la solution de l'ODE associée convergeait respectivement vers le sous espace dominant ou vers les vecteurs propres de cet espace dominant (sauf sur un ensemble de conditions initiales de mesure nulle).

Signalons qu'une version modifiée de l'algorithme PASTd a été proposée par [Sakai 98] pour estimer les vecteurs propres d'un espace propre minorant. Il s'agit simplement de changer un signe devant le gain d'adaptation et d'effectuer une orthonormalisation à chaque étape. Et pour $\beta = 1$, il a été démontré dans [Sakai 98] que cet algorithme PASTd modifié prend la structure générale (3.8) avec $\mu_n = \frac{1}{n}$ et que la solution de l'ODE associée convergeait vers les vecteurs propres de l'espace minorant (sauf sur un ensemble de conditions initiales de mesure nulle).

3.5.2. Etude de convergence

On s'intéresse ici au problème spécifique de l'étude de la convergence dans le cas d'algorithmes de sous espaces où l'ensemble des points localement ou globalement asymptotiquement stable des ODE associés à ces algorithmes est constitué de l'ensemble des bases orthonormées de l'espace dominant [resp. minorant] de dimension r de R_x qui constitue ainsi un *continuum* d'attracteurs. Les colonnes de W_n ne convergent pas en général vers les vecteurs propres v_1, \dots, v_r [resp. v_{m-r+1}, \dots, v_m] et nous n'avons aucune preuve de convergence de W_n vers une base orthonormée particulière de l'espace dominant [resp. minorant] associé pour une séquence de gain μ_n vérifiant les conditions : $\sum_{n=1}^{\infty} \mu_n = +\infty$ et $\lim_{n \rightarrow +\infty} \mu_n = 0$. Par suite, considérer la distribution asymptotique de W_n est sans fondement et les résultats résumés à la section 3.2 ne s'appliquent pas. Pour résoudre cette difficulté, une solution ([Delmas 98-c]) est d'étudier le comportement de la trajectoire de

$$\Pi_n \equiv W_n W_n^T \quad (3.33)$$

dont la dynamique pour les algorithmes (3.30) et (3.32) est commandée par une équation stochastique de la forme :

$$\Pi_n = \Pi_{n-1} + \mu_n H(\Pi_{n-1}, x_n x_n^T) + \mu_n^2 K(\Pi_{n-1}, x_n x_n^T) \quad (3.34)$$

obtenue en combinant (3.30) ou (3.32) avec (3.33). Nous constatons que le champ H et le terme complémentaire K de cet algorithme ne dépendent que de Π_{n-1} et *non* de W_{n-1} . Cette propriété rend possible l'étude de l'évolution de Π_n sans tenir compte de l'évolution de la matrice W_n sous jacente. Les caractéristiques de Π_n sont en fait très pertinentes, car elles caractérisent le sous espace invariant dominant estimé. Puisque l'ODE associée à l'algorithme d'approximation stochastique (3.34) admet un unique point globalement asymptotiquement stable $\Pi_* \equiv \sum_{i=1}^r v_i v_i^T$, (résultat en particulier déduit de [Baldi 89]) concernant les ODE associées aux algorithmes (3.30) et (3.32)), nos algorithmes stochastiques (3.34) convergent presque sûrement vers Π_* si Π_n reste infiniment souvent dans un compact (ce que nous admettons). Les résultats généraux d'approximations gaussiennes de [Benveniste 87, th.2, p.104] peuvent alors s'appliquer, avec cette fois l'algorithme d'approximation stochastique plus général :

$$\theta_n = \theta_{n-1} + \mu_n H(\theta_{n-1}, x_n) + \mu_n^2 K(\theta_{n-1}, x_n)$$

assortis de la condition supplémentaire H3 (ii), ([Benveniste 87, p.220]) : $K(\theta, x)$ doit être une fonction uniformément bornée pour (θ, x) appartenant à un compact fixé. Il a été démontré [Delmas 98-c] que cette dernière condition est bien vérifiée par ces algorithmes. Cependant, nous remarquons que certaines valeurs propres de la dérivée $D_h(\theta_*)$ du champ moyen h de nos algorithmes sont réelles positives, ce qui ne permet pas l'application directe des résultats de [Benveniste 87, th.2, p.104] pour ces algorithmes. De plus, le paramètre multidimensionnel θ_n qui dans notre problème est constitué par les termes de la matrice Π_n est contraint à être de rang r alors que l'équation de Liapounoff (3.19) ne tient pas compte de cette contrainte.

Pour résoudre ces difficultés, une paramétrisation de la matrice symétrique réelle Π_n qui s'appuie sur la propriété suivante [Delmas 98-c] peut être considérée :

Si Π est une matrice $m \times m$ symétrique réelle de rang r , alors

$$\Pi = \Pi_* + \sum_{(i,j) \in \mathcal{A}_\pi} \theta_{i,j}(\Pi) A_{i,j} + O(\|\Pi - \Pi_*\|^2),$$

où $\mathcal{A}_\pi \equiv \{(i, j) \mid 1 \leq i \leq j \leq m \text{ et } i \leq r\}$ et où $(A_{i,j})_{i=1, \dots, m, j=1, \dots, m}$ est la base orthonormée suivante (avec le produit scalaire $(A, B) \equiv \text{Tr}(A^T B)$) de l'espace vectoriel sur le corps \mathbb{R} des matrices symétriques réelle $m \times m$:

$$A_{i,j} = \begin{cases} v_i v_j^T & i = j \\ \frac{v_i v_j^T + v_j v_i^T}{\sqrt{2}} & i < j. \end{cases}$$

Comme il y a $\frac{r}{2}(2m - r + 1)$ paires (i, j) dans \mathcal{A}_π , ce qui est le nombre de paramètres libres de la variété constituée par les matrices $m \times m$ symétriques réelles de rang r (conf. la paramétrisation de leurs SVD), cette propriété montre que l'ensemble de matrices $\{A_{i,j} \mid (i, j) \in \mathcal{A}_\pi\}$ est en fait une base orthonormée du plan tangent à cette variété au point Π_* . Par suite les composantes dans la base orthonormée $A_{i,j}$ de la projection orthogonale de la matrice Π_n sur le plan tangent à cette variété au point Π_* sont données par $\Theta(\Pi_n) \equiv \mathcal{A}^T \text{Vec}(\Pi_n - \Pi_*)$ avec $\mathcal{A} \equiv [\dots, \text{Vec}(A_{i,j}), \dots]$, $(i, j) \in \mathcal{A}_\pi$. D'où :

$$\text{Vec}(\Pi_n) = \text{Vec}(\Pi_*) + \mathcal{A}\theta(\Pi_n) + O(\|\theta(\Pi_n)\|^2).$$

Et si $\mathcal{P}(\theta)$ désigne la matrice $m \times m$ symétrique réelle de rang r unique sur un voisinage de $\theta = 0$, telle que $\mathcal{A}^T \text{Vec}(\mathcal{P}(\theta) - \Pi_*) = \theta$, alors $\text{Vec}(\mathcal{P}(\theta)) = \text{Vec}(\Pi_*) + \mathcal{A}\theta + O(\|\theta\|^2)$. Par suite à l'aide de la bijection $\mathcal{P}(\theta) \leftrightarrow \theta$, nous pouvons considérer l'équation stochastique décrivant l'évolution de θ_n issue de (3.34) décrivant l'évolution de Π_n :

$$\theta_n = \theta_{n-1} + \mu_n \phi(\theta_{n-1}, x_n) + \mu_n^2 \psi(\theta_{n-1}, x_n) \quad (3.35)$$

avec $\phi(\theta, x) \equiv \mathcal{A}^T \text{Vec}(H(\mathcal{P}(\theta), xx^T))$ et $\psi(\theta, x) \equiv \mathcal{A}^T \text{Vec}(G(\mathcal{P}(\theta), xx^T))$. Après avoir vérifié la condition ([Benveniste 87, A3(ii), p.220]) sur ψ et après donné des expressions analytiques de la dérivée $D_h(\theta_*)$ du champ moyen (dont les valeurs propres

sont maintenant à parties réelles strictement négatives) et de la covariance du champ ϕ de l'algorithme (3.35) (pour des observations x_n gaussiennes indépendantes centrées), l'équation de Liapounoff associée (??) peut être résolue analytiquement. On obtient pour les algorithmes (3.30) ou (3.32), l'expression commune :

$$C_{\Pi} = \sum_{1 \leq i \leq r < j \leq m} \frac{\lambda_i \lambda_j}{2(\lambda_i - \lambda_j)} (v_i \otimes v_j + v_j \otimes v_i)(v_i \otimes v_j + v_j \otimes v_i)^T. \quad (3.36)$$

Cette méthodologie a été étendue dans [Delmas 98c] :

– aux versions “lissées” des algorithmes étudiés à l'aide du nouveau paramètre $\theta_n \equiv \begin{bmatrix} \theta(R_n) \\ \theta(\Pi_n) \end{bmatrix}$ avec $\theta(R_n) \equiv \mathcal{A}'^T \text{Vec}(R_n)$ où $\mathcal{A}' \equiv [\dots, \text{Vec}(A_{i,j}), \dots]$, $(i, j) \in \{1, \dots, m\}^2$.

– aux signaux complexes en considérant la base orthonormée suivante de l'espace vectoriel des matrices $m \times m$ hermitiennes :

$$A_{i,j} = \begin{cases} v_i v_i^H & i = j \\ \frac{v_i v_j^H + v_j v_i^H}{\sqrt{2}} & i < j \\ \frac{v_i v_j^H - v_j v_i^H}{i\sqrt{2}} & i > j. \end{cases}$$

où l'ensemble \mathcal{A}_π et remplacé par l'ensemble $\mathcal{A}_\pi \equiv \{(i, j) \mid 1 \leq i \leq r \text{ ou } 1 \leq j \leq r\}$ qui contient maintenant $r(2m - r)$ paires (i, j) .

– à des observations x_n corrélées dérivées du modèle de corrélation spécifique $x_n \equiv [x(n), x(n-1), \dots, x(n-m+1)]^T$ avec $x(n)$ processus ARMA stationnaire.

Notons que l'expression de la matrice de covariance (3.36) obtenue pour les deux algorithmes adaptatifs considérés est très voisine de celle obtenue dans les mêmes conditions par maximum de vraisemblance block issu de (x_1, \dots, x_n) . Ce résultat provient de la propriété d'invariance de l'estimateur du maximum de vraisemblance. Il est démontré dans [Delmas 98d] que l'estimateur du maximum de vraisemblance de Π_* coïncide avec la matrice $\Pi_n \equiv W_n W_n^T$ où W_n est solution de la minimisation du critère de [Yang 95] :

$$\frac{1}{n} \sum_{k=1}^n \|x_k - W W^T x_k\|^2$$

et que la solution W_n de cette minimisation est solution de $\frac{1}{n} \sum_{k=1}^n H'(W, x_k) = 0$ (où H' est le champ de l'algorithme de Yang (3.32)) et que par suite Π_n est solution de $\frac{1}{n} \sum_{k=1}^n H(\Pi, x_n x_n^T) = 0$. Enfin, en appliquant des résultats classiques sur les M -estimateurs [Bickel 93, th.1, p.312], il s'en déduit la matrice de covariance C_{Π} de la loi asymptotique de Π_n .

$$C_{\Pi} = \sum_{1 \leq i \leq r < j \leq n} \frac{\lambda_i \lambda_j}{(\lambda_i - \lambda_j)^2} (v_i \otimes v_j + v_j \otimes v_i)(v_i \otimes v_j + v_j \otimes v_i)^T.$$

De plus remarquons que par continuité, le comportement de n'importe quelle fonction différentiable de Π_n peut être obtenu. On peut ainsi déduire les lois asymptotiques des DOA θ_k estimées par l'algorithme MUSIC standard et comparer les matrices de covariance de ces lois asymptotiques à celles obtenues en estimation batch. Là aussi ces expressions sont très voisines et peuvent être interprétées. Ainsi par exemple dans le cas d'une seule source, on obtient [Delmas 98d] respectivement à partir des estimées de Π_n issues des algorithmes adaptatifs non lissés et du maximum de vraisemblance bloc :

$$\text{var } \theta_1 \sim \mu \frac{m\sigma_1^2}{2\alpha_1} \left(1 + \frac{1}{m\rho_1}\right) \frac{1}{\rho_1} \qquad \text{var } \theta_1 \sim \frac{1}{n} \frac{1}{\alpha_1} \left(1 + \frac{1}{m\rho_1}\right) \frac{1}{\rho_1}$$

où m , ρ_1 et α_1 désignent respectivement le nombre de capteurs, le rapport signal à bruit et un facteur géométrique qui ne dépend que de la structure du réseau de capteurs. Ainsi les termes respectifs $\frac{\mu m \sigma_1^2}{2}$ et $\frac{1}{n}$ des algorithmes adaptatifs et batch jouent le même rôle.

3.6. Conclusion

En résumé, les différents points rencontrés dans l'étude d'un algorithme adaptatif sont :

- l'étude de la convergence,
- l'évaluation de la vitesse asymptotique de convergence,
- la poursuite des non-stationnarités,
- la distinction entre algorithmes à gain décroissant, utiles à l'identification de paramètres fixes, et algorithmes à gain constant, indispensables à la poursuite de non-stationnarités.

Ces différents points se retrouvent dans tous les problèmes d'estimation adaptative. L'étude des estimateurs de moyenne a permis de découvrir les principaux comportements caractéristiques des algorithmes adaptatifs et d'introduire la notion d'hypermodèle. Les résultats généraux présentés permettent le calcul des paramètres optimaux d'un algorithme adaptatif et de l'EQM asymptotique correspondante.

3.7. Bibliographie

- ANDERSON et JOHNSON, 1982, "*Exponential convergence of adaptive identification and control algorithms*", Automatica vol.18.
 ASTRÖM et SÖDERSTRÖM, 1974, "*Uniqueness of the maximum likelihood estimates of the parameters of an ARMA model*", IEEE Trans. AC-19 n°6.
 BALDI et HORNIK, 1989, "*Neural networks and principal component analysis* :

- Learning from examples without local minima* , Neural Networks, 1989.
- BENVENISTE et al., 1980, "*Robust identification of a non-minimum phase system : blind adjustment of a linear equalizer in data communications*", IEEE Trans. AC-25, n°3.
- BENVENISTE, METIVIER, PRIOURET, 1987, Algorithmes adaptatifs et Approximations stochastiques, Masson, Paris.
- BERSHAD et al., 1980, "*Tracking characteristics of the LMS ALE to a linear chirp signal in noise*", IEEE Trans ASSP-28, 1980.
- BERSHAD et MACCHI, 1989, "*Comparison of RLS and LMS for tracking of a chirped signal*", Proc. ICASSP Glasgow, 1989.
- BICKEL et al., 1993, Efficient and adaptive estimation for semiparametric models, Johns Hopkins series in Math. Sciences.
- BILLINGSLEY, 1968, Convergence of Probability Measures, Wiley, New York.
- DELMAS et ALBERGE, 1998-a, "*Asymptotic performance analysis of subspace adaptive algorithms introduced in the neural network literature*", IEEE Trans. on Signal Processing, vol. 46, 1998.
- DELMAS, 1998-b, "*Performance analysis of a Givens parameterized adaptive eigenspace algorithm*" , Signal Processing (Elsevier), vol. 68, 1998.
- DELMAS et CARDOSO, 1998-c, "*Asymptotic distributions associated to Oja's learning equation for Neural Networks*", IEEE Trans. on Neural Networks, vol. 9, 1998.
- DELMAS et CARDOSO, 1998-d, "*Performance analysis of an adaptive algorithm for tracking dominant subspace.*" IEEE Trans. on Signal Processing, vol.46, 1998.
- DUFLO, 1990, Méthodes récursives aléatoires, Masson, Paris.
- FORT et PAGES, 1996, "*Convergence of stochastic algorithms : from the Kushner and Clark theorem to the Lyapunov functional method*" , Adv. in Applied Proba., no. 28, 1996.
- KUSHNER et YIN, 1997, Stochastic Approximation Algorithms and Applications, Springer.
- LA SALLE et LEFSCHETZ, 1961, Stability by Liapunov's direct method, Academic Press, New York.
- LIAPOUNOFF, 1907, Problème général de la stabilité du mouvement, réédition Editions Jacques Gabay, Paris, 1989.
- LJUNG, 1975, "*Counterexamples to general convergence of a commonly used recursive identification method*" IEEE Trans. AC-20.
- LJUNG et SODERSTRÖM, 1983, Theory and Practice of Recursive Identification, MIT Press, Cambridge.
- MACCHI et BELLANGER, 1987, "*Le point sur le filtrage adaptatif transverse*", Colloque GRETSI, 1987.
- MACCHI et al. , 1989, "*Comparaison des algorithmes adaptatifs pour la poursuite de non-stationnarités*", Numéro spécial de Traitement du Signal, vol.6, n°5, 1989.
- MACCHI et BERSHAD, 1991, "*Adaptive recovery of a Chirped sinusoid in Noise*", IEEE Trans. on Signal Processing, vol.39, n°3.
- MICHAUT, 1992, Méthodes Adaptatives pour le Signal, Hermès, Paris.

- OJA, 1982, "A simplified neuron model as a principal components analyzer", J. Math. Biol., vol. 15, 1982.
- OJA 1985, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix", Journal of mathematical analysis and applications, vol. 106, 1985.
- OJA, OGAWA WANGVIWATTANA, 1992-a, *Principal component analysis by homogeneous neural networks, Part I : The weighted subspace criterion*, IEICE Trans. Inform. and Syst., vol.E75-D, 1992.
- OJA, 1992-b, "Principal components, minor components and linear neural networks", Neural Networks, vol. 5, 1992.
- REGALIA, 1990, *An adaptive unit norm filter with applications to signal analysis and Karhunen Loève transformations*, IEEE Trans. on Circuits and Systems, vol. 37, 1990.
- RIOU, CHONAVEL, COCHET, 1996, "Adaptive subspace estimation - Application to moving sources localization and blind channel identification", Proc. ICASSP Atlanta, GA, 1996.
- RUTISHAUSER, 1969, "Computational aspects of F.L.Bauer's simultaneous iteration method", Numer. Math. vol. 13.
- SAKAI et SHIMIZU, 1998, *A new adaptive algorithm for minor component analysis*, Signal Processing (Elsevier), vol. 71, 1998.
- SHYNK, 1989, "Adaptive IIR Filtering", IEEE ASSP Magazine, 1989.
- SÖDERSTRÖM, 1975, "On the uniqueness of maximum likelihood identification", Automatica vol. 11.
- SOLO, 1979, "The convergence of AML", IEEE Trans.-AC24 p958.
- SOLO et KONG, 1995, *Adaptive Signal Processing Algorithms, Stability and performance*, Prentice Hall.
- WILLIAMS, 1985, "Feature discovery through error-correcting learning", Inst. Cognitive Sci., Univ. California, San Diego, Tech. Rep. 8501, 1985.
- XU, OJA, SUEN, 1992, "Modified Hebbian learning for curve and surface fitting", Neural Networks, vol. 5, no.3, 1992.
- YAN, HELMKE, MOORE 1994, "Global analysis of Oja's flow for neural networks", IEEE Trans. on Neural Networks, vol. 5, no. 5, September 1994.
- YANG, KAVEH, 1988, "Adaptive eigenspace algorithms for direction or frequency estimation and tracking", IEEE Trans. on ASSP, vol. 36, no. 2, February 1988.
- YANG, 1995, "Projection approximation subspace tracking", IEEE, Trans. on Signal Processing, vol. 43, no. 1, Jan. 1995.
- YANG, 1996, "Asymptotic convergence analysis of the projection approximation subspace tracking algorithms", Signal Processing (Elsevier), vol. 50, 1996.

Chapitre 4

Algorithmes rapides, vitesse et stabilité numérique

4.1. Introduction et notations

Nous avons vu au chapitre 2 les nombreux avantages de l'algorithme des Moindres carrés récursifs RLS pour la réalisation du filtrage optimal de Wiener. Une des difficultés auxquelles on se heurte pour sa mise en œuvre sur des applications temps-réel est la *complexité de calcul* en $O(M^2)$, qui devient insurmontable dès que la taille M des filtres à estimer est élevée. Le but des *Algorithmes rapides* est de résoudre cette difficulté. Notre but est de réaliser le même travail d'optimisation que l'algorithme des Moindres carrés récursifs RLS, mais en réduisant la complexité de calcul par itération à un $O(M)$ (donc comparable à celle du LMS). La structure à optimiser est toujours ici le filtrage de Wiener optimal (voir Figure 1.4 au Chapitre 1).

L'idée de base est de se débarrasser des opérations de mise à jour matricielles du RLS, responsables de la complexité $O(M^2)$, en exploitant la structure de Toeplitz de la matrice de covariance R : elle permet, par l'algorithme de Levinson, de résoudre les équations normales en $O(M^2)$ opérations au lieu de $O(M^3)$ pour une matrice quelconque. La difficulté à résoudre vient de ce qu'on travaille ici directement sur les données, et donc sur des matrices de covariance estimées R_M qui n'ont pas strictement la structure Toeplitz de R : elles ont cependant des propriétés d'invariance par décalage qui en font des matrices "proches de Toeplitz". Trois approches seront traitées pour cette inversion rapide : elles diffèrent par le mode de représentation du filtre adaptatif et les jeux de paramètres utilisés pour représenter la matrice inverse. Équivalentes

Chapitre rédigé par Maurice BELLANGER .

mathématiquement et en complexité de calcul, ces méthodes ont des comportements différents par rapport à la stabilité numérique des algorithmes.

4.2. Filtres transverses rapides

Les techniques de moindres carrés nécessitent l'inversion de la matrice d'autocorrélation (AC) du signal d'entrée. Cette matrice AC inverse est complètement déterminée par les coefficients de prédiction et la puissance de l'erreur de prédiction. Il en est de même pour l'estimation en temps réel de la matrice, qui est déterminée par les estimations des coefficients de prédiction avant et arrière et de l'énergie de l'erreur. Dans ces conditions, tous les éléments nécessaires aux techniques de moindres carrés récursives sont contenus dans ces paramètres, qui peuvent être calculés et mis à jour. Les algorithmes transversaux rapides réalisent cette fonction efficacement pour les filtres RIF (à réponse impulsionnelle finie) en structure directe.

4.2.1. Equations de récurrence du filtre d'ordre M

Le filtre adaptatif RIF d'ordre M est défini par :

$$e(n+1) = y(n+1) - \theta^t(n)X(n+1) \quad (4.1)$$

où les vecteurs $\theta(n)$ des coefficients du filtre et $X(n)$ des données d'entrée les plus récentes comportent M éléments. L'entrée $x(n)$, la référence $y(n)$ et l'erreur de sortie $e(n)$ sont des scalaires.

La fonction coût, définie comme l'énergie de l'erreur et qui s'écrit :

$$E_M(n) = \sum_{p=1}^n \lambda^{n-p} [y(p) - \theta^t(n)X(p)]^2 \quad (4.2)$$

conduit, par annulation des dérivées, à la solution suivante :

$$\theta(n) = R_M^{-1}(n)r_{yx}(n) \quad (4.3)$$

avec :

$$R_M(n) = \sum_{p=1}^n \lambda^{n-p} X(p)X^t(p) \quad ; \quad r_{yx}(n) = \sum_{p=1}^n \lambda^{n-p} y(p)X(p) \quad (4.4)$$

En séparant le dernier terme dans les sommes, on peut faire apparaître une relation de récurrence sur la matrice d'autocorrélation et le vecteur d'intercorrélations qui, combinées à la relation (4.3) de définition des coefficients du filtre, donnent la relation de récurrence suivante sur les coefficients :

$$\theta(n+1) = \theta(n) + R_M^{-1}(n+1)X(n+1) [y(n+1) - X^t(n+1)\theta(n)] \quad (4.5)$$

Pour être complet, il faut noter que l'on peut mettre en évidence une autre relation de récurrence sur les coefficients. En effet, au temps $n + 1$, la définition des coefficients donne :

$$[\lambda R_M(n) + X(n+1)X^t(n+1)]\theta(n+1) = \lambda r_{yx}(n) + y(n+1)X(n+1) \quad (4.6)$$

ce qui, après quelques manipulations, conduit à :

$$\theta(n+1) = \theta(n) + \lambda^{-1}R_M^{-1}(n)X(n+1)[y(n+1) - \theta^t(n+1)\theta(n+1)] \quad (4.7)$$

Dans ces expressions, la matrice $R_M^{-1}(n+1)$ peut être mise à jour par récurrence grâce au lemme d'inversion, extension matricielle d'une identité algébrique classique [Giordano 89].

Etant donné quatre matrices vérifiant l'équation :

$$A = B + CDC^t \quad (4.8)$$

L'inverse de la matrice A s'écrit :

$$A^{-1} = B^{-1} - B^{-1}C[C^tB^{-1}C + D^{-1}]^{-1}C^tB^{-1} \quad (4.9)$$

En fait, la matrice A^{-1} peut prendre différentes formes, qui se déduisent de l'identité :

$$[B - UDV]^{-1} = [I_M - B^{-1}UDV]B^{-1} \quad (4.10)$$

où B est une matrice supposée non singulière, par le développement en série suivant :

$$(I_M - B^{-1}UDV)^{-1}B^{-1} = [I_M + B^{-1}UDV + (B^{-1}UDV)^2 + \dots]B^{-1} \quad (4.11)$$

La convergence de cette série est assurée si les valeurs propres de $(B^{-1}UDV)$ sont, en valeur absolue, inférieures à l'unité. L'équation (4.11) est un lemme d'inversion généralisée [Tylarski 86]. Par exemple, en regroupant et en sommant tous les termes de la série, sauf le premier, il vient :

$$(B - UDV)^{-1} = I_M + B^{-1}U[I_M - DVB^{-1}U]^{-1}DVB^{-1} \quad (4.12)$$

ce qui est une autre forme de l'inverse.

Ce lemme s'applique au calcul de R_M^{-1} de telle sorte qu'il n'est plus besoin de faire d'inversion de matrice, mais juste une division par un scalaire. Considérant la récurrence :

$$R_M(n+1) = \lambda R_M(n) + X(n+1)X^t(n+1) \quad (4.13)$$

on choisit :

$$B = \lambda R_M(n) ; C = X(n+1) ; D = 1$$

et l'équation (4.9) donne :

$$R_M^{-1}(n+1) = \frac{1}{\lambda} \left[R_M^{-1}(n) - \frac{R_M^{-1}(n)X(n+1)X^t(n+1)R_M^{-1}(n)}{\lambda + X^t(n+1)R_M^{-1}(n)X(n+1)} \right] \quad (4.14)$$

Il est commode de définir le gain d'adaptation $G(n)$ par :

$$G(n) = R_M^{-1}(n)X(n) \quad (4.15)$$

ce qui, en reportant dans (4.14), et après simplification, donne :

$$G(n+1) = \frac{1}{\lambda + X^t(n+1)R_M^{-1}(n)X(n+1)} R_M^{-1}(n)X(n+1) \quad (4.16)$$

Alors, l'expression (4.14) et la récurrence (4.5) se réécrivent :

$$R_M^{-1}(n+1) = \frac{1}{\lambda} [R_M^{-1}(n) - G(n+1)X^t(n+1)R_M^{-1}(n)] \quad (4.17)$$

et :

$$\theta(n+1) = \theta(n) + G(n+1) [y(n+1) - X^t(n+1)\theta(n)] \quad (4.18)$$

Les trois équations (4.16), (4.17) et (4.18) constituent une technique récursive pour mettre à jour les coefficients du filtre sans inversion de matrice : c'est l'algorithme RLS à oubli λ , déjà vu et étudié au Chapitre 2 (voir section 2.2.2). Bien entendu, une valeur initiale non nulle $R_M^{-1}(0)$ est nécessaire au départ ; ce point est repris ultérieurement.

Avec cette technique, le nombre d'opérations arithmétiques est proportionnel à M^2 , à cause des multiplications de matrice. En fait, on peut se débarrasser complètement des matrices et rendre la complexité proportionnelle à M seulement, en considérant $R_M(n)$ comme l'estimation en temps réel de la matrice AC du signal d'entrée et en exploitant le fait que son inverse peut être représentée par les paramètres de prédiction. Avant d'introduire les algorithmes rapides, il est opportun de mettre en évidence plusieurs relations fondamentales entre les variables des moindres carrés.

4.2.2. Relations entre les variables des Moindres carrés

Dans la technique des moindres carrés récursifs, l'inversion de matrice est évitée par l'introduction d'un scalaire. Soit $\psi(n+1)$ la quantité :

$$\psi(n+1) = \frac{\lambda}{\lambda + X^t(n+1)R_M^{-1}(n)X(n+1)} \quad (4.19)$$

On vérifie directement avec (4.14) que cette quantité s'écrit aussi :

$$\psi(n+1) = 1 - X^t(n+1)R_M^{-1}(n+1)X(n+1) \quad (4.20)$$

Le scalaire qui apparaît dans cette expression, $\gamma(n)$ défini par :

$$\gamma(n) = X^t(n)R_M^{-1}(n)X(n) \quad (4.21)$$

a une interprétation particulière en traitement du signal. Il s'écrit

$$\gamma(n) = b^T[A + bb^T]^{-1}b \quad \text{avec } b = X(n) \text{ et } A = \lambda R_M(n-1)$$

et en réinjectant le Lemme d'inversion matricielle (2.31) on obtient

$$\gamma(n) = \frac{b^T A^{-1} b}{1 + b^T A^{-1} b}$$

Il en résulte par positivité de la matrice A , les bornes de la variable $\gamma(n)$:

$$0 \leq \gamma(n) \leq 1 \quad (4.22)$$

Quand on considère le terme qui figure en exposant dans la densité jointe de l'ensemble de M variables gaussiennes de moyenne nulle, on remarque que sa forme est semblable à $\gamma(n)$, qui, dans ces conditions, peut être interprété comme son estimation au temps n . D'où l'appellation de variable de vraisemblance donnée à $\gamma(n)$ en théorie de l'estimation [Kalouptsidis 93]. En fait, $\gamma(n)$ est une mesure de la vraisemblance pour que les M données d'entrées les plus récentes proviennent d'un processus Gaussien dont la matrice AC, $R_M(n)$, a été déterminée en utilisant toutes les observations passées disponibles. Une valeur faible de $\gamma(n)$ indique que les données récentes sont vraisemblablement des échantillons d'un signal Gaussien alors qu'une valeur proche de l'unité traduit des observations inattendues ; dans ce dernier cas, le vecteur $X(n+1)$ se trouve en dehors du sous-espace signal estimé, ce qui peut être dû à des variations dans le temps de la statistique du signal, par exemple. Par conséquent, $\gamma(n)$ peut être utilisé pour détecter les changements dans la statistique du signal. A noter que, si le gain d'adaptation $G(n)$ est disponible, on obtient immédiatement $\gamma(n)$ par :

$$\gamma(n) = X^t(n)G(n) \quad (4.23)$$

D'après leur définition $\psi(n)$ et $\gamma(n)$ ont les mêmes propriétés. Celles qui sont utiles pour les moindres carrés sont présentées ci-après.

En multipliant à droite les deux membres de (4.13) par $R_M^{-1}(n)$, il vient :

$$R_M(n+1)R_M^{-1}(n) = \lambda I_M + X(n+1)X^t(n+1)R_M^{-1}(n) \quad (4.24)$$

Avec l'identité :

$$\det [I_M + V_1 V_2^t] = 1 + V_1^t V_2 \quad (4.25)$$

où V_1 et V_2 sont des vecteurs à M éléments, et la définition de $\psi(n)$, on obtient :

$$\psi(n+1) = \lambda^M \frac{\det R_M(n)}{\det R_M(n+1)} \quad (4.26)$$

En raison de la définition de $R_M(n)$ et de sa positivité, et en faisant appel à la récurrence (4.13), on vérifie que $\psi(n)$ est borné par :

$$0 \leq \psi(n) \leq 1 \quad (4.27)$$

ce qui, par une autre voie, confirme l'inégalité (4.22). C'est un résultat fondamental, qui peut être exploité pour vérifier que, dans la mise en œuvre des algorithmes, les conditions des moindres carrés sont bien respectées.

Ensuite, on peut montrer que la variable $\psi(n)$ possède, en réalité, une signification physique simple.

La technique des moindres carrés récurrents, appliquée à la prédiction linéaire avant, est basée sur la fonction coût :

$$E_a(n) = \sum_{p=1}^n \lambda^{n-p} [x(p) - A^t(n)X(p-1)]^2 \quad (4.28)$$

qui est l'énergie de l'erreur de prédiction avant. Le vecteur des coefficients est donné par :

$$A(n) = R_M^{-1}(n-1)r_M^a(n) \quad (4.29)$$

avec :

$$r_M^a(n) = \sum_{p=1}^n \lambda^{n-p} x(p)X(p-1) \quad (4.30)$$

Il faut bien noter que l'indice $p-1$ ci-dessus est caractéristique de la prédiction linéaire avant. La mise à jour des coefficients se fait par :

$$A(n+1) = A(n) + G(n)e_a(n+1) \quad (4.31)$$

où :

$$e_a(n+1) = x(n+1) - A^t(n)X(n) \quad (4.32)$$

est l'erreur de prédiction avant *a priori*.

Ce sont les coefficients mis à jour qui sont utilisés pour calculer l'erreur de prédiction *a posteriori* :

$$\varepsilon_a(n+1) = x(n+1) - A^t(n+1)X(n) \quad (4.33)$$

Avec la relation (4.31), il vient :

$$\varepsilon_a(n+1) = e_a(n+1) [1 - G^t(n)X(n)] \quad (4.34)$$

et de la définition (4.20) il résulte que :

$$\psi(n) = \frac{\varepsilon_a(n+1)}{e_a(n+1)} \quad (4.35)$$

c'est-à-dire que $\psi(n)$ est simplement le rapport des erreurs de prédiction avant au temps suivant.

Un résultat similaire est obtenu avec la prédiction arrière. La fonction coût dans ce cas se définit par :

$$E_b(n) = \sum_{p=1}^n \lambda^{n-p} [x(p-M) - B^t(n)X(p)]^2 \quad (4.36)$$

Le vecteur des coefficients de prédiction arrière est donné par :

$$B(n) = R_M^{-1}(n)r_M^b(n) \quad (4.37)$$

avec :

$$r_M^b(n) = \sum_{p=1}^n \lambda^{n-p} x(p-M)X(p) \quad (4.38)$$

La mise à jour se fait par :

$$B(n+1) = B(n) + G(n+1)e_b(n+1) \quad (4.39)$$

avec :

$$e_b(n+1) = x(n+1-M) - B^t(n)X(n+1) \quad (4.40)$$

L'erreur de prédiction *a posteriori* arrière est donnée par :

$$\varepsilon_b(n+1) = x(n+1-M) - B^t(n+1)X(n+1) \quad (4.41)$$

En reportant (4.39) dans (4.41), il vient :

$$\psi(n+1) = \frac{\varepsilon_b(n+1)}{e_b(n+1)} \quad (4.42)$$

ce qui montre que $\psi(n)$ est le rapport des erreurs de prédiction arrière de même indice.

Plus généralement que la prédiction, le même résultat peut être vérifié directement pour tout filtre adaptatif, basé sur le critère des moindres carrés, c'est-à-dire :

$$\psi(n+1) = \frac{\varepsilon(n+1)}{e(n+1)} \quad (4.43)$$

A noter encore que ce résultat peut fournir une autre démonstration de l'inégalité (4.27).

En effet, l'énergie de l'erreur, donnée par (4.2), s'écrit au temps $n+1$:

$$E_M(n+1) = \lambda \sum_{p=1}^n \lambda^{n-p} [y(p) - \theta^t(n+1)X(p)]^2 + \varepsilon^2(n+1) \quad (4.44)$$

Soit la fonction $E'_M(n+1)$ telle que :

$$E'_M(n+1) = \lambda \sum_{p=1}^n \lambda^{n-p} [y(p) - \theta^t(n)X(p)]^2 + e^2(n+1) \quad (4.45)$$

Par définition, on a :

$$E'_M(n+1) \geq E_M(n+1) ; E_M(n+1) - \varepsilon^2(n+1) \geq E'_M(n+1) - e^2(n+1) \quad (4.46)$$

et par conséquent :

$$e^2(n+1) \geq \varepsilon^2(n+1) \quad (4.47)$$

L'erreur *a posteriori* est inférieure ou égale à l'erreur *a priori*.

A titre d'illustration, on peut calculer $\psi(n)$ pour des signaux simples. Par exemple, si $M = 2$ et si $x(n)$ est un signal sinusoïdal, par application directe de la définition, on trouve à l'état stationnaire :

$$\psi(n) \approx 1 - 2(1 - \lambda) = 2\lambda - 1 \quad (4.48)$$

ce résultat se généralise à M quelconque quand la fréquence de la sinusoïde satisfait les relations $\pi/M \leq \omega \leq \pi - \pi/M$.

Et pour un bruit blanc :

$$E[\psi(n)] \approx 1 - M(1 - \lambda) \quad (4.49)$$

L'énergie d'erreur de prédiction avant se calcule par récurrence. En reportant (4.29) dans l'expression de $E_a(n+1)$, il vient :

$$E_a(n+1) = \sum_{p=1}^{n+1} \lambda^{n+1-p} x^2(p) - A^t(n+1)r_M^a(n+1) \quad (4.50)$$

Les relations de récurrence pour $A(n+1)$ et $r_M^a(n+1)$, en liaison avec les définitions du gain d'adaptation et des coefficients de prédiction, donnent, après simplification :

$$E_a(n+1) = \lambda E_a(n) + e_a(n+1)\varepsilon_a(n+1) \quad (4.51)$$

et, de même pour la prédiction arrière :

$$E_b(n+1) = \lambda E_b(n) + e_b(n+1)\varepsilon_b(n+1) \quad (4.52)$$

Ces calculs apparaissent directement dans les algorithmes de moindres carrés rapides (MCR).

4.2.3. Algorithme MCR basé sur les erreurs a priori

Dans la technique des moindres carrés récurrents le gain d'adaptation est mis à jour avec l'inverse de la matrice AC. On va maintenant utiliser les paramètres de prédiction [Falconer 78].

Soit la matrice AC d'ordre $M + 1$ au temps $n+1$, $R_{M+1}(n + 1)$. On peut vérifier, en reprenant la définition, qu'elle peut être partitionnée de deux manières, avec les équations de prédiction avant et arrière, à savoir :

$$R_{M+1}(n + 1) = \begin{bmatrix} \sum_{p=1}^{n+1} \lambda^{n+1-p} x^2(p) & [r_M^a(n + 1)]^t \\ r_M^a(n + 1) & R_M(n) \end{bmatrix} \quad (4.53)$$

et :

$$R_{M+1}(n + 1) = \begin{bmatrix} R_M(n + 1) & r_M^b(n + 1) \\ [r_M^b(n + 1)]^t & \sum_{p=1}^{n+1} \lambda^{n+1-p} x^2(P - M) \end{bmatrix} \quad (4.54)$$

L'objectif est de trouver le gain au temps $n+1$, $G(n + 1)$, connaissant le gain au temps n . La définition générale du gain donne :

$$R_M(n + 1)G(n + 1) = X(n + 1) \quad (4.55)$$

Comme la matrice AC, $R_M(n)$, apparaît dans la décomposition (4.53), on calcule :

$$R_{M+1}(n + 1) \begin{bmatrix} 0 \\ G(n) \end{bmatrix} = \begin{bmatrix} [r_M^a(n + 1)]^t G(n) \\ X(n) \end{bmatrix} \quad (4.56)$$

Avec les définitions du gain et des coefficients de prédiction avant, on a :

$$[r_M^a(n + 1)]^t G(n) = A^t(n + 1)X(n) \quad (4.57)$$

Introduisant l'erreur de prédiction *a posteriori*, il vient :

$$R_{M+1}(n + 1) \begin{bmatrix} 0 \\ G(n) \end{bmatrix} = X_1(n + 1) - \begin{bmatrix} \varepsilon_a(n + 1) \\ 0 \end{bmatrix} \quad (4.58)$$

où $X_1(n + 1)$ est le vecteur des $M + 1$ données d'entrée les plus récentes. De même, le partitionnement (4.54) conduit à :

$$R_{M+1}(n + 1) \begin{bmatrix} G(n + 1) \\ 0 \end{bmatrix} = \begin{bmatrix} X(n + 1) \\ [r_M^b(n + 1)]^t G(n + 1) \end{bmatrix} \quad (4.59)$$

Avec des définitions du gain et des coefficients de prédiction arrière, il vient :

$$[r_M^b(n+1)]^t G(n+1) = B^t(n+1)X(n+1) \quad (4.60)$$

et :

$$R_M(n+1) \begin{bmatrix} G(n+1) \\ 0 \end{bmatrix} = X_1(n+1) - \begin{bmatrix} 0 \\ \varepsilon_b(n+1) \end{bmatrix} \quad (4.61)$$

Or le gain d'adaptation à l'ordre $M+1$, noté $G_1(n+1)$, est défini par :

$$R_{M+1}(n+1)G_1(n+1) = X_1(n+1) \quad (4.62)$$

ce qui fait que l'équation (4.58) se réécrit :

$$R_{M+1}(n+1) \left[G_1(n+1) - \begin{bmatrix} 0 \\ G(n) \end{bmatrix} \right] = \begin{bmatrix} \varepsilon_a(n+1) \\ 0 \end{bmatrix} \quad (4.63)$$

Quant à l'équation (4.61), elle devient :

$$R_{M+1}(n+1) \left[G_1(n+1) - \begin{bmatrix} G(n+1) \\ 0 \end{bmatrix} \right] = \begin{bmatrix} 0 \\ \varepsilon_b(n+1) \end{bmatrix} \quad (4.64)$$

Ensuite, les équations matricielles de la prédiction sont utilisées pour calculer $G_1(n+1)$ à partir de $G(n)$, puis $G(n+1)$ à partir de $G_1(n+1)$.

L'équation matricielle de la prédiction avant, qui résulte du regroupement de (4.29) et (4.50), s'écrit :

$$R_{M+1}(n+1) \begin{bmatrix} 1 \\ -A(n+1) \end{bmatrix} = \begin{bmatrix} E_a(n+1) \\ 0 \end{bmatrix} \quad (4.69)$$

En égalant les facteurs dans les relations (4.63) et (4.69), on obtient :

$$G_1(n+1) = \begin{bmatrix} 0 \\ G(n) \end{bmatrix} + \frac{\varepsilon_a(n+1)}{E_a(n+1)} \begin{bmatrix} 1 \\ -A(n+1) \end{bmatrix} \quad (4.65)$$

De même, avec l'équation de prédiction arrière :

$$R_{M+1}(n+1) \begin{bmatrix} -B(n+1) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ E_b(n+1) \end{bmatrix} \quad (4.66)$$

et, en égalant les facteurs dans (4.66) et (4.64), on obtient :

$$G_1(n+1) = \begin{bmatrix} G(n+1) \\ 0 \end{bmatrix} + \frac{\varepsilon_b(n+1)}{E_b(n+1)} \begin{bmatrix} -B(n+1) \\ 1 \end{bmatrix} \quad (4.67)$$

En réalité, le facteur scalaire du membre de droite de cette équation n'a pas à être calculé, car il est déjà disponible. Soit la partition du gain à l'ordre $M+1$:

$$G_1(n+1) = \begin{bmatrix} M(n+1) \\ m(n+1) \end{bmatrix} \quad (4.68)$$

où le vecteur $M(n+1)$ possède M éléments. Le scalaire $m(n+1)$ intervient dans la dernière ligne de l'équation (4.67), qui s'écrit :

$$m(n+1) = \frac{\varepsilon_b(n+1)}{E_b(n+1)} \quad (4.69)$$

Le gain d'adaptation à M éléments est mis à jour par :

$$G(n+1) = M(n+1) + m(n+1)B(n+1) \quad (4.70)$$

Mais il faut connaître le gain mis à jour pour calculer $B(n+1)$. En fait, on dispose de deux équations à deux inconnues et, en reportant l'équation de mise à jour des coefficients de prédiction arrière (4.39) dans (4.70), on obtient le gain en fonction de quantités disponibles :

$$G(n+1) = \frac{1}{1 - m(n+1)e_b(n+1)} [M(n+1) + m(n+1)B(n)] \quad (4.71)$$

Si, par contre, on reporte l'équation (4.70) dans l'équation de mise à jour des coefficients et que l'on calcule $B(n+1)$ en premier, alors on obtient :

$$B(n+1) = \frac{1}{1 - m(n+1)e_b(n+1)} [B(n) + M(n+1)e_b(n+1)] \quad (4.72)$$

Un nouveau scalaire apparaît dans ces équations. Comme il faut toujours se méfier des diviseurs dans les algorithmes, il est important de voir sa signification physique et d'évaluer sa gamme d'amplitude. En combinant (4.69) et la mise à jour de l'énergie (4.52), il vient :

$$1 - m(n+1)e_b(n+1) = 1 - \frac{\varepsilon_b(n+1)e_b(n+1)}{E_b(n+1)} = \frac{\lambda E_b(n)}{E_b(n+1)} \quad (4.73)$$

c'est-à-dire que ce diviseur est le rapport de deux valeurs consécutives de l'énergie d'erreur de prédiction arrière et il a pour domaine :

$$0 < 1 - m(n+1)e_b(n+1) \leq 1 \quad (4.74)$$

Bien entendu, au fur et à mesure que le temps passe, il s'approche de l'unité, d'autant plus que l'erreur de prédiction est faible. Au passage, on peut remarquer que l'équation (4.73) peut fournir une autre manière de mettre à jour l'énergie de prédiction arrière.

Egalement, il peut être intéressant, par exemple pour élaborer des algorithmes approchés mais simplifiés, de faire apparaître la relation entre deux valeurs successives des éléments du vecteur gain :

$$\begin{bmatrix} g_1(n+1) \\ \vdots \\ g_M(n+1) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ g_1(n) \\ \vdots \\ g_M(n) \end{bmatrix} + m(n+1) \begin{bmatrix} B(n+1) \\ -1 \end{bmatrix} + \frac{\varepsilon_a(n+1)}{E_a(n+1)} \begin{bmatrix} -1 \\ A(n+1) \end{bmatrix} \quad (4.75)$$

Globalement, un algorithme rapide a été obtenu et l'enchaînement des opérations est donné dans le [Tableau 4.1] (**MCR1**).

Algorithme MCR1 à oublier λ
Mémoire du filtre à l'instant n
Etat du filtre : $\theta(n)$
Prédicteurs Avant et Arrière : $A(n), B(n)$
Vecteur d'observation : $X(n)$
Gain d'adaptation : $G(n)$
Résidu de prédiction avant : $E_a(n)$
Nouvelles données à l'instant $n+1$
Signal d'entrée : $x(n+1)$; Référence : $y(n+1)$
Calcul du Gain d'adaptation
$e_a(n+1) = x(n+1) - A^t(n)X(n)$
$A(n+1) = A(n) + G(n)e_a(n+1)$
$\varepsilon_a(n+1) = x(n+1) - A^t(n+1)X(n)$
$E_a(n+1) = \lambda E_a(n) + e_a(n+1)\varepsilon_a(n+1)$
$G_1(n+1) = \begin{bmatrix} 0 \\ G(n) \end{bmatrix} + \frac{\varepsilon_a(n+1)}{E_a(n+1)} \begin{bmatrix} 1 \\ -A(n+1) \end{bmatrix} = \begin{bmatrix} M(n+1) \\ m(n+1) \end{bmatrix}$
$e_b(n+1) = x(n+1 - M) - B^t(n)X(n+1)$
$G(n+1) = \frac{1}{1 - m(n+1)e_b(n+1)} (M(n+1) + m(n+1)B(n))$
$B(n+1) = B(n) + G(n+1)e_b(n+1)$
Filtre adaptatif
$e(n+1) = y(n+1) - \theta^t(n)X(n+1)$
$\theta(n+1) = \theta(n) + G(n+1)e(n+1)$

Tableau 4.1. Algorithme rapide basé sur les erreurs a priori

L'initialisation conforme aux moindres carrés est obtenue en prenant $A(n) = B(n) = G(n) = 0$ et $E_a(0) = E_0$, constante de faible valeur, étudiée par la suite.

Pour le bilan des opérations, la mise à jour du gain d'adaptation nécessite $8M + 4$ multiplications et deux divisions, sous la forme de calculs d'inverses ; dans la partie filtrage, il faut $2M$ multiplications. Pour stocker les coefficients et les variables internes, il faut environ $6M$ mémoires. A l'évidence, la réduction du volume de calculs par rapport aux algorithmes de moindres carrés récursifs est considérable dès que l'ordre dépasse quelques unités. Cependant, il est encore possible de gagner un peu.

L'algorithme ci-dessus repose principalement les erreurs *a priori*. Par exemple, l'erreur arrière *a posteriori* n'est pas calculée ; en exploitant toutes les erreurs de prédiction, un algorithme plus efficace et mieux équilibré est obtenu [Carayannis 83, Cioffi 84].

4.2.4. Algorithme MCR basé sur l'ensemble des erreurs de prédiction

L'algorithme précédent repose sur la relation de récurrence (4.5). On peut élaborer un algorithme similaire avec la relation de récurrence duale (4.7) qui conduit à une autre définition du vecteur gain d'adaptation à M éléments :

$$R_M(n)G'(n+1) = X(n+1) \quad (4.76)$$

A cause du terme $R_M(n)$ dans $G'(n+1)$, on peut l'appeler gain *a priori*, par opposition au gain *a posteriori* $G(n+1)$. De même à l'ordre $M+1$:

$$R_{M+1}(n)G'_1(n+1) = X_1(n+1) \quad (4.77)$$

En exploitant les deux partitionnements (4.53) et (4.54) de la matrice AC estimée $R_{M+1}(n)$ comme précédemment, il vient :

$$R_{M+1}(n) \begin{bmatrix} G'(n+1) \\ 0 \end{bmatrix} = X_1(n+1) - \begin{bmatrix} 0 \\ e_b(n+1) \end{bmatrix} \quad (4.78)$$

et :

$$R_{M+1}(n) \begin{bmatrix} 0 \\ G'(n) \end{bmatrix} = X_1(n+1) - \begin{bmatrix} e_a(n+1) \\ 0 \end{bmatrix} \quad (4.79)$$

En reportant dans (4.78) la définition du gain (4.82), il vient :

$$R_{M+1}(n) \left[G'_1(n+1) - \begin{bmatrix} G'(n+1) \\ 0 \end{bmatrix} \right] = \begin{bmatrix} 0 \\ e_b(n+1) \end{bmatrix} \quad (4.80)$$

L'identification avec l'équation de la prédiction arrière (4.66) donne une première expression du gain à l'ordre $M+1$:

$$G'_1(n+1) = \begin{bmatrix} G'(n+1) \\ 0 \end{bmatrix} + \frac{e_b(n+1)}{E_b(n)} \begin{bmatrix} -B(n) \\ 1 \end{bmatrix} \quad (4.81)$$

De même, avec (4.79) et (4.77) on obtient :

$$R_{M+1}(n) \left[G'_1(n+1) - \begin{bmatrix} 0 \\ G'(n) \end{bmatrix} \right] = \begin{bmatrix} e_a(n+1) \\ 0 \end{bmatrix} \quad (4.82)$$

L'identification avec l'équation de la prédiction avant donne une autre expression du gain :

$$G'_1(n+1) = \begin{bmatrix} 0 \\ G'(n) \end{bmatrix} + \frac{e_a(n+1)}{E_a(n)} \begin{bmatrix} 1 \\ -A(n) \end{bmatrix} \quad (4.83)$$

La technique pour calculer $G'(n+1)$ consiste alors à calculer $G'_1(n+1)$ avec les paramètres de la prédiction avant par (4.83) et, ensuite, à utiliser (4.81).

Une fois le gain $G'(n)$ mis à jour, il peut servir pour calculer les coefficients du filtre, à l'aide de la relation de mise à jour des coefficients (4.7), qui devient :

$$\theta(n+1) = \theta(n) + \lambda^{-1} G'(n+1) [y(n+1) - X^t(n+1)\theta(n+1)] \quad (4.84)$$

Il faut calculer l'erreur *a posteriori* :

$$\varepsilon(n+1) = y(n+1) - X^t(n+1)\theta(n+1) \quad (4.85)$$

avec les éléments disponibles. C'est alors qu'on peut faire appel à la variable $\psi(n)$ définie par (4.19) et qui est le rapport des erreurs *a priori* et *a posteriori*. Il vient :

$$\lambda + X^t(n+1)G'(n+1) = \frac{\lambda}{\psi(n+1)} = \alpha(n+1) \quad (4.86)$$

et c'est la variable $\alpha(n+1)$ qui est calculée dans l'algorithme.

En reprenant l'équation (4.43), c'est à dire : $\varepsilon(n+1) = \psi(n+1)e(n+1)$, les coefficients sont mis à jour par :

$$\theta(n+1) = \theta(n) + \frac{e(n+1)}{\alpha(n+1)} G'(n+1) \quad (4.87)$$

A noter que les deux gains d'adaptation sont reliés par le scalaire $\alpha(n+1)$ et que le gain *a priori* peut être défini par :

$$G'(n+1) = [\lambda + X^t(n+1)R_M^{-1}(n)X(n+1)] G(n+1) = \alpha(n+1)G(n+1) \quad (4.88)$$

La variable $\alpha(n+1)$ se calcule par sa définition. Cependant une technique récursive peut être élaborée, comme pour le gain. La variable correspondant à l'ordre $M+1$, $\alpha_1(n+1)$, est définie par :

$$\alpha_1(n+1) = \lambda + X_1^t(n+1)G'_1(n+1) \quad (4.89)$$

Les deux expressions du gain (4.81) et (4.83) donnent :

$$\alpha_1(n+1) = \alpha(n) + \frac{e_a^2(n+1)}{E_a(n)} = \alpha(n+1) + \frac{e_b^2(n+1)}{E_b(n)} \quad (4.90)$$

ce qui constitue une relation de récurrence pour $\alpha(n+1)$ et aussi pour $\psi(n+1)$.

Comme la variable $\alpha(n+1)$ est disponible, on peut l'utiliser pour calculer les erreurs *a posteriori* $\varepsilon_a(n+1)$ et $\varepsilon_b(n+1)$ avec une seule multiplication, au lieu de M et autant d'additions pour les définitions.

L'erreur de prédiction arrière *a priori* peut être obtenue directement, par le partitionnement du vecteur gain à $M+1$ dimension :

$$G'_1(n+1) = \begin{bmatrix} M'(n+1) \\ m'(n+1) \end{bmatrix} \quad (4.91)$$

La dernière ligne de l'équation matricielle (4.81) donne :

$$m'(n+1) = \frac{e_b(n+1)}{E_b(n)} \quad (4.92)$$

ce qui fournit $e_b(n+1)$ avec une multiplication seulement. Néanmoins, en raison de problèmes d'arrondi, cette simplification est à éviter.

L'algorithme complet est donné dans le [Tableau 4.2], p.162.

L'initialisation des moindres carrés correspond à :

$$A(0) = B(0) = G'(0) = 0 ; E_a(0) = E_0 ; E_b(0) = \lambda^{-M} E_0 \quad (4.93)$$

où E_0 est une constante positive faible. La définition (4.86) donne aussi $\alpha(0) = \lambda$.

La mise à jour du gain nécessite $6M+9$ multiplications et trois divisions, sous forme de calcul d'inverse. La partie filtrage demande $2M+1$ multiplications. Il faut environ $6M+7$ mémoires. Globalement, ce second algorithme peut apporter une réduction de complexité par rapport au précédent appréciable pour les grandes valeurs de M .

4.2.5. Conditions de stabilité et initialisation

Pour un ensemble d'échantillons d'un signal non nul, les calculs de moindres carrés fournissent un ensemble unique de coefficients de prédiction. Les algorithmes récursifs correspondent à un calcul exact à chaque instant et, par suite, leur stabilité est garantie en théorie, quelque soit le facteur de pondération λ . Comme les algorithmes

Algorithme MCR2 à oubli λ	
Mémoire du filtre à l'instant n	
Etat du filtre : $\theta(n)$	
Prédicteurs Avant et Arrière : $A(n), B(n)$ Vecteur d'observation : $X(n)$	
Gain d'adaptation : $G'(n)$	
Résidu de prédiction avant et arrière : $E_a(n), E_b(n)$	
Nouvelles données à l'instant $n + 1$	
Signal d'entrée : $x(n + 1)$; Référence : $y(n + 1)$	
Calcul du Gain d'adaptation	
$e_a(n + 1) = x(n + 1) - A^t(n)X(n)$	
$A(n + 1) = A(n) + G'(n)e_a(n + 1)/\alpha(n)$	
$E_a(n + 1) = [E_a(n) + e_a(n + 1)e_a(n + 1)/\alpha(n)].\lambda$	
$G'_1(n + 1) = \begin{bmatrix} 0 \\ G'(n) \end{bmatrix} + \frac{e_a(n + 1)}{E_a(n)} \begin{bmatrix} 1 \\ -A(n) \end{bmatrix} = \begin{bmatrix} M(n + 1) \\ m(n + 1) \end{bmatrix}$	
$e_b(n + 1) = x(n + 1 - M) - B^t(n)X(n + 1)$	
$G'(n + 1) = M(n + 1) + m(n + 1)B(n)$	
$\alpha_1(n + 1) = \alpha(n) + e_a(n + 1)e_a(n + 1)/E_a(n)$	
$\alpha(n + 1) = \alpha_1(n + 1) - m(n + 1)e_b(n + 1)$	
$E_b(n + 1) = [E_b(n) + e_b(n + 1)e_b(n + 1)/\alpha(n + 1)].\lambda$	
$B(n + 1) = B(n) + G'(n + 1)e_b(n + 1)/\alpha(n + 1)$	
Filtre adaptatif	
$e(n + 1) = y(n + 1) - \theta^t(n)X(n + 1)$	
$\theta(n + 1) = \theta(n) + G'(n + 1)e(n + 1)/\alpha(n + 1)$	

Tableau 4.2. Algorithme rapide basé sur l'ensemble des erreurs de prédiction

rapides sont mathématiquement équivalents aux algorithmes récursifs, ils jouissent de la même propriété. Leur stabilité est même garantie pour un signal nul, pourvu que les énergies d'erreur de prédiction initiales soient strictement positives.

Cette propriété théorique très importante et intéressante disparaît malheureusement dans les réalisations, à cause de la précision finie des machines [Lin 84, Ljung 85, Cioffi 87].

Les algorithmes rapides tirent leur efficacité d'une représentation des paramètres de moindres carrés, les estimations de l'inverse de la matrice AC du signal d'entrée et du vecteur d'intercorrélation entrée-référence, qui est réduite au nombre minimal de variables. Avec la précision limitée des opérations arithmétiques, cette représentation est forcément approchée. Ainsi, l'estimation de la matrice AC inverse $R_M^{-1}(n)$ apparaît dans les algorithmes MCR, par son produit par le vecteur de données $X(n)$, qui donne le gain d'adaptation $G(n)$. Comme le vecteur de données est une quantité exacte par définition, les erreurs d'arrondi engendrées lors du calcul amènent des écarts de l'estimation de la matrice AC inverse par rapport à sa valeur idéale, en précision infinie. En fait, les erreurs aléatoires sur les éléments de la matrice AC ne modifient pas de

manière significative les valeurs propres, mais elles changent les directions propres. Par contre, un biais dans l'estimation de la fonction AC fait varier les valeurs propres.

Quand le vecteur de données $X(n)$ est multiplié par la matrice théorique $R_M^{-1}(n)$, le vecteur résultant a une grandeur limitée, car $X(n)$ appartient au sous-espace signal de la matrice. Par contre, si c'est une approximation de $R_M^{-1}(n)$ qui est utilisée, le vecteur de données peut avoir une projection importante en dehors de l'espace signal de la matrice ; dans ce cas, la grandeur du vecteur résultant n'est plus maîtrisée, ce qui peut faire sortir les variables des limites qui leur sont imposées par le principe des moindres carrés. De plus, les valeurs propres peuvent devenir négatives, à cause de l'accumulation à long terme des erreurs d'arrondi.

Plusieurs variables ont un domaine limité dans les algorithmes MCR. Une des opérations est le calcul des erreurs *a posteriori*, à partir des coefficients mis à jour à l'aide du gain d'adaptation et des erreurs *a priori*. Dans ces conditions, la qualité de la représentation de $R_M^{-1}(n)X(n)$ par $G(n)$ peut être vérifiée par le rapport $\psi(n)$ des erreurs *a priori* et *a posteriori*. Dans les réalisations, la variable $\psi(n)$ correspond, d'après sa définition, à :

$$\psi(n) = 1 - X^t(n)[R_M^q(n)]^{-1}X(n) \quad (4.94)$$

où $R_M^q(n)$ est la matrice utilisée à la place de la matrice idéale $R_M(n)$. La variable $\psi(n)$ peut dépasser l'unité si les valeurs propres de $R_M^q(n)$ deviennent négatives ; elle peut devenir négative si le scalaire $X^t(n)[R_M^q(n)]^{-1}X(n)$ dépasse l'unité.

L'accumulation des erreurs d'arrondi, quand elle se produit, demande du temps. La première précaution à prendre dans la mise en œuvre des algorithmes rapides consiste à s'assurer que le scalaire $X^t(n)[R_M^q(n)]^{-1}X(n)$ ne dépasse pas l'unité.

Soit, d'abord, un signal d'entrée qui est un bruit blanc gaussien de moyenne nulle et de puissance σ_x^2 . On vérifie directement que, pour n suffisamment grand, il vient :

$$R_M(n) \approx \frac{\sigma_x^2}{1-\lambda} I_M \quad (4.95)$$

Au voisinage de l'origine des temps, on peut faire l'hypothèse que, l'accumulation n'étant pas encore significative, la matrice $R_M^q(n)$ ne diffère de $R_M(n)$ que par l'addition d'erreurs aléatoires, ce qui a pour effet de découpler, dans une certaine mesure, $R_M^q(n)$ et $X(n)$. D'où la justification de l'approximation suivante :

$$X^t(n)[R_M^q(n)]^{-1}X(n) \approx \frac{1-\lambda}{\sigma_x^2} X^t(n)X(n) \quad (4.96)$$

La variable $X^t(n)X(n)$ est gaussienne, de moyenne $M\sigma_x^2$ et de variance $2M\sigma_x^4$. En prenant un facteur de crête égal à 4, une condition pour garder positif le rapport des

erreurs de prédiction est la suivante :

$$(1 - \lambda)(M + 4\sqrt{2M}) < 1 \quad (4.97)$$

Cette inégalité montre qu'une limite inférieure est imposée à λ . Par exemple, si $M = 10$, alors il faut prendre : $\lambda > 0.95$.

Ensuite, pour un signal d'entrée plus général, la situation extrême se produit quand le vecteur des données $X(n)$ se trouve dans la direction propre associée à la plus petite valeur propre de la matrice $R_M^q(n)$. Toujours sous l'hypothèse d'addition d'erreurs aléatoires de moyenne nulle et en négligeant l'accumulation à long terme, l'approximation suivante peut être faite :

$$\lambda_{\min}^q(n) \approx \frac{\lambda_{\min}}{1 - \lambda} \quad (4.98)$$

où λ_{\min} est la plus petite valeur propre de la matrice AC du signal d'entrée. En remplaçant $X^t(n)X(n)$ par $M\sigma_x^2$, la condition sur $\psi(n)$ devient :

$$(1 - \lambda) \frac{M\sigma_x^2}{\lambda_{\min}} < 1 \quad (4.99)$$

Une telle condition peut apparaître extrêmement restrictive, étant donné que le rapport $\sigma_x^2/\lambda_{\min}$ peut prendre des valeurs très grandes. Par exemple, si $x(n)$ est un signal déterministe dans un bruit additif et que l'ordre du prédicteur est suffisamment grand, il s'agit du rapport signal à bruit. Bien entendu, étant donné que les inégalités (4.97) et (4.99) ont été obtenues en faisant des hypothèses très restrictives sur les erreurs d'arrondi, il faut les utiliser avec précaution. Néanmoins, elles montrent clairement que le facteur de pondération λ ne peut pas être pris arbitrairement petit.

Les réalisations récursives des algorithmes de moindres carrés pondérés nécessitent l'initialisation des variables d'état. Si le signal n'est pas connu avant $n = 0$, il est raisonnable de supposer qu'il est nul ainsi que les coefficients de prédiction. Cependant, il faut donner une valeur positive, E_0 , à l'énergie d'erreur de prédiction avant. Pour que l'algorithme démarre correctement, il faut que les conditions initiales correspondent à une situation de moindres carrés.

Une énergie de prédiction avant positive quand les coefficients sont nuls peut être interprétée comme correspondant à un signal dont tous les échantillons précédents sont nuls sauf un. Si, de plus, le gain $G(0)$ est nul, alors la séquence d'entrée est la suivante :

$$x(-M) = (\lambda^{-M} E_0)^{1/2} \quad x(n) = 0 \text{ si } M + 1 \leq n \leq 0 \quad (4.100)$$

La valeur correspondante pour l'énergie d'erreur de prédiction arrière est $E_B(o) = x^2(-M) = \lambda^{-M} E_o$, d'où l'initialisation (4.93). Dans ces conditions, la valeur initiale de l'estimation de la matrice AC s'écrit :

$$R_M(0) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \lambda^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda^{-(M-1)} \end{bmatrix} E_o \quad (4.101)$$

et la matrice réellement utilisée pour estimer la matrice AC d'entrée a pour valeur :

$$R_M^*(n) = R_M(n) + \lambda^n R_M(0) \quad (4.102)$$

La plus petite valeur propre de l'espérance de $R_M^*(n)$, notée $\lambda_{\min}^*(n)$, est obtenue, en utilisant (4.4), par :

$$\lambda_{\min}^*(n) = \frac{1 - \lambda^n}{1 - \lambda} \lambda_{\min} + \lambda^n E_o \quad (4.103)$$

Dans cette expression, le premier terme du membre de droite croît avec n, alors que le second décroît. La phase transitoire et l'état stationnaire sont dans la même situation quand à la stabilité si une borne inférieure est imposés à E_o . L'équation (4.103) peut se réécrire :

$$\lambda_{\min}^*(n) = \frac{\lambda_{\min}}{1 - \lambda} + \lambda^n (E_o - \frac{\lambda_{\min}}{1 - \lambda}) \quad (4.104)$$

Alors $\lambda_{\min}^*(n)$ est au moins égal à $\lambda_{\min}/1 - \lambda$ si E_o est lui-même supérieur ou égal à cette quantité. En se reportant à la condition (4.99), il vient :

$$E_o \geq M \sigma_x^2 \quad (4.105)$$

Cette condition, obtenue sous des hypothèses très restrictives, est en général beaucoup trop pessimiste et on peut en pratique prendre des valeurs plus petites, pour ne pas trop ralentir la convergence. La représentation de la matrice $R_M(n)$ dans le système peut rester très précise pendant une durée supérieure à la phase transitoire, si la précision de la machine est suffisante. Par exemple, des expériences répétées sur un microprocesseur à 16 bits et avec une arithmétique à virgule fixe ont montré qu'une valeur minimale pour E_o était environ $0.01 \sigma_x^2$ [Alcantara 86]. Si le nombre de bits dans la machine est plus faible, il faut augmenter E_o .

Finalement, les développements ci-dessus montrent que les énergies des erreurs initiales ne peuvent pas être choisies arbitrairement petites.

4.2.6. Contrôle de l'accumulation des erreurs d'arrondi

Les erreurs d'arrondi sont engendrées par les opérations de quantification qui sont généralement effectuées après les multiplications et les divisions. Elles sont considérées comme des sources indépendantes, à spectre plat et de variance $q^2/12$, q désignant le pas de quantification, lié au nombre de bits des données dans la machine utilisée. La particularité des algorithmes MCR introduits dans les paragraphes précédents est qu'une accumulation peut se produire [Lin 84, Ljung 85, Cioffi 87].

Globalement, l'algorithme MCR1 par exemple, est formé de trois boucles qui se recouvrent. La boucle de mise à jour du gain d'adaptation fait la liaison entre les boucles de mise à jour des coefficients de prédiction avant et arrière et les récurrences correspondantes peuvent produire une accumulation d'erreurs d'arrondi.

Considérons par exemple, au temps n , l'addition au vecteur des coefficients de prédiction arrière $B(n)$, d'un vecteur erreur $\Delta B(n)$. Alors, au temps $n+1$, en négligeant le facteur scalaire dans (4.71) et en prenant l'algorithme MCR1, l'écart s'écrit :

$$\begin{aligned} \Delta B(n+1) = & [I_M [1 + m(n+1)e_b(n+1)] - G(n+1)X^t(n+1)] \Delta B(n) \\ & - \Delta B(n) [\Delta B(n)]^t m(n+1)X(n+1) \quad (4.106) \end{aligned}$$

Si $\Delta B(n)$ est un vecteur aléatoire de moyenne nulle, ce qui est le cas pour l'arrondi, la moyenne de $\Delta B(n+1)$ n'est pas nulle en raison de la présence de la matrice $\Delta B(n) [\Delta B(n)]^t$ dans (4.106) et parce que, $m(n+1)$ étant lié au signal d'entrée, l'espérance du produit $m(n+1)X(n+1)$ est en général non nulle. Le facteur de $\Delta B(n)$ est proche de la matrice unité, il peut même avoir des valeurs propres supérieures à un, et, donc, l'introduction des vecteurs d'erreurs $\Delta B(n)$ à chaque instant produit une dérive des coefficients. L'effet est un décalage des coefficients par rapport à l'optimum, ce qui dégrade les performances. Cependant, si la valeur propre minimal $\lambda_{1 \min}$ de la matrice AC d'ordre $M+1$ du signal d'entrée est proche de la puissance du signal σ_x^2 , la puissance de l'erreur de prédiction, également proche de σ_x^2 , est une fonction des coefficients variant peu et la dérive peut se poursuivre jusqu'au point où l'écart qui en résulte sur les valeurs et vecteurs propres de la matrice $R_M^q(n)$ fait que la variable $\psi(n)$ dépasse ses bornes (4.27). Alors, l'algorithme n'est plus dans une configuration de moindres carrés et devient instable.

A noter que l'accumulation à long terme des erreurs d'arrondi affecte les coefficients de prédiction arrière, mais, sauf à l'ordre $M = 1$, pratiquement pas les coefficients de la prédiction avant. Ce fait est dû au décalage des éléments du vecteur gain apporté par l'équation (4.65).

Pour maîtriser l'accumulation d'erreurs d'arrondi, une méthode efficace consiste à trouver une variable représentative et à l'utiliser pour limiter la dérive des coefficients

[Botto 89],[Slock 91]. La variable $\xi(n)$ suivante :

$$\xi(n+1) = [x(n+1-M) - B^t(n+1)X(n+1)] - m(n+1)E_b(n+1) \quad (4.107)$$

est représentative de ce phénomène d'accumulation car c'est la différence entre deux quantités mises à jour dans des boucles distinctes, la prédiction arrière et la prédiction avant. Le vecteur $B(n)$ figure dans le calcul de $\xi(n+1)$ et une technique de moindres carrés peut être appliquée pour corriger les coefficients de prédiction arrière.

Comme le gain d'adaptation $G(n)$ est une variable disponible, l'opération consiste simplement à remplacer $e_b(n+1)$ par $[e_b(n+1) + \xi(n+1)]$ dans l'équation de mise à jour des coefficients de prédiction arrière [14]. Ainsi, dans l'algorithme MCR-2, la substitution suivante est faite :

$$e_b(n+1) = 2 [x(n+1-M) - B^t(n)X(n+1)] - m'(n+1)E_b(n) \quad (4.108)$$

L'algorithme MCR-1 peut être stabilisé de la même manière. En effet, dans la variable $\xi(n+1)$, le second terme peut s'exprimer en fonction de l'énergie d'erreur de prédiction avant. L'erreur de prédiction arrière s'écrit :

$$e_b(n+1) = \frac{m(n+1)E_b(n+1)}{\psi(n+1)} \quad (4.109)$$

Ensuite, par application du calcul matriciel, les équations de la prédiction donnent, en utilisant la première ligne dans la relation (4.64) et la dernière ligne dans la relation (4.66) :

$$\frac{\det(R_M(n))}{\det(R_{M+1}(n+1))} E_a(n+1) = \frac{\det(R_M(n+1))}{\det(R_{M+1}(n+1))} E_b(n+1) = 1 \quad (4.110)$$

La relation (4.27) fournit alors le résultat suivant :

$$\psi(n+1) = \lambda^M \frac{E_b(n+1)}{E_a(n+1)} \quad (4.111)$$

et, finalement, il vient :

$$e_b(n+1) = \lambda^{-M} E_a(n+1) m(n+1) \quad (4.112)$$

L'erreur de prédiction arrière est ainsi obtenue en utilisant uniquement les variables de la prédiction avant. Le contrôle d'erreur d'arrondi se fait alors en remplaçant dans l'algorithme MCR-1 l'erreur de prédiction arrière par :

$$e_b(n+1) = 2 [x(n+1-M) - B^t(n)X(n+1)] - \lambda^{-M} E_a(n+1) m(n+1) \quad (4.113)$$

Il a été montré expérimentalement que ces techniques permettent de faire fonctionner correctement les algorithmes dans certaines applications. Cependant, il faut bien noter que ce contrôle d'erreur d'arrondi ne garantit pas la stabilité numérique de l'algorithme, en théorie [Sayed 2003].

4.2.7. Moindres carrés avec décimation

Dans certaines applications, notamment en transmission numérique, la fréquence d'échantillonnage du signal de référence est inférieure à celle du signal d'entrée et une réduction de fréquence d'échantillonnage intervient en sortie du filtre adaptatif. Les algorithmes de moindres carrés rapides peuvent être adaptés à cette situation [Slock 92]. La présentation est faite dans le cas d'une réduction par 2 de la fréquence d'échantillonnage.

La séquence d'entrée est décomposée en deux suites entrelacées $x_1(n)$ et $x_2(n)$ et les vecteurs de données d'entrée sont définis par :

$$X_{2M}(n) = \begin{bmatrix} x_2(n) \\ x_1(n) \\ \vdots \\ x_1(n+1-M) \end{bmatrix} ; \quad X_{1,2M}(n+1) = \begin{bmatrix} x_1(n+1) \\ x_2(n) \\ \vdots \\ x_2(n+1-M) \end{bmatrix} \quad (4.114)$$

La fonction coût s'écrit alors :

$$J(n) = \sum_{p=1}^n \lambda^{n-p} [y(p) - \theta_{2M}^t(n) X_{2M}(p)]^2 \quad (4.115)$$

où $\theta_{2M}(n)$ est le vecteur des coefficients à $2M$ éléments. Le filtrage adaptatif avec division par 2 de la fréquence d'échantillonnage correspond aux deux équations :

$$\begin{aligned} e(n+1) &= y(n+1) - \theta_{2M}^t(n) X_{2M}(n+1) \\ \theta_{2M}(n+1) &= \theta_{2M}(n) + G_{2M}(n+1)e(n+1) \end{aligned} \quad (4.116)$$

Le vecteur gain d'adaptation $G_{2M}(n)$ est lui-même défini à partir de la matrice AC :

$$R_{2M}(n) = \sum_{p=1}^n \lambda^{n-p} X_{2M}(p) X_{2M}^t(p)$$

comme suit :

$$G_{2M}(n) = R_{2M}^{-1}(n) X_{2M}(n)$$

Dans l'algorithme des moindres carrés rapides, le vecteur gain d'adaptation est mis à jour par prédiction linéaire. Une première énergie d'erreur de prédiction est définie par :

$$E_{1a}(n) = \sum_{p=1}^n \lambda^{n-p} [x_1(p) - A_{1,2M}^t(n) X_{2M}(p-1)]^2$$

et elle conduit à l'équation matricielle :

$$R_{2M+1}(n+1) \begin{bmatrix} 1 \\ -A_{1,2M}(n+1) \end{bmatrix} = \begin{bmatrix} E_{1a}(n+1) \\ 0 \end{bmatrix}$$

où la matrice à $(2M + 1) \times (2M + 1)$ éléments s'exprime par :

$$R_{2M+1}(n+1) = \sum_{p=1}^{n+1} \lambda^{n+1-p} \begin{bmatrix} x_1(p) \\ X_{2M}(p-1) \end{bmatrix} [x_1(p), X_{2M}^t(p-1)]$$

La procédure consiste à calculer un gain d'adaptation étendu $G_{1,2M+1}(n+1)$ à partir de la prédiction linéaire avant et à l'utiliser avec la prédiction linéaire arrière pour mettre à jour le gain $G_{1,2M}(n+1)$. Ensuite, la même démarche est appliquée en utilisant la donnée suivante $x_2(n+1)$, ce qui fournit un autre gain d'adaptation au temps $n+1$, $G_{2M}(n+1)$. L'enchaînement des opérations est présenté dans le [Tableau 4.3] p.170. En fait, l'algorithme à une dimension est effectué deux fois dans la partie prédiction.

La technique se généralise à une réduction de fréquence d'échantillonnage d'un facteur M supérieur à 2. Elle s'applique également au cas de signaux d'entrée multi-dimensionnels.

4.3. Algorithmes en treillis

Alors que les algorithmes de moindres carrés rapides pour les structures adaptatives transversales sont essentiellement basés sur des récurrences temporelles, les algorithmes pour les structures en treillis font appel à la fois aux récurrences temporelles et sur l'ordre. Pour un ordre du filtre fixé M , ils demandent davantage d'opérations que l'équivalent en transversal. Par contre, ils fournissent tous les filtres adaptatifs intermédiaires, ce qui est intéressant notamment dans les applications où l'ordre n'est pas connu d'avance ou imposé et où il faut procéder par essais successifs.

4.3.1. Récurrences sur l'ordre pour les coefficients de prédiction

Soit $A_M(n)$, $B_M(n)$, $E_{aM}(n)$, $E_{bM}(n)$ et $G_M(n)$ les vecteurs des coefficients de prédiction du signal d'entrée, les énergies des erreurs et le vecteur gain d'adaptation respectivement, au temps n pour un filtre d'ordre M .

L'équation matricielle de prédiction linéaire avant à l'ordre $M-1$ s'écrit :

$$R_M(n) \begin{bmatrix} 1 \\ -A_{M-1}(n) \end{bmatrix} = \begin{bmatrix} E_{a(M-1)}(n) \\ 0 \end{bmatrix} \quad (4.117)$$

De même pour la prédiction arrière :

$$R_M(n) \begin{bmatrix} -B_{M-1}(n) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ E_{b(M-1)}(n) \end{bmatrix} \quad (4.118)$$

Algorithme MCR 2D-1D
<p style="text-align: center;">Mémoire à l'instant n :</p> <p>Coefficients du filtre : $\theta_{2M}(n)$ Coefficients de prédiction avant : $A_{1,2M}(n)$; $A_{2,2M}(n)$ Coefficients de prédiction arrière : $B_{1,2M}(n)$; $B_{2,2M}(n)$ Vecteur d'observation : $X_{2n}(n)$ Gains d'adaptation : $G_{2M}(n)$; $G_{1,2M}(n)$ Résidus de prédiction avant : $E_{1a}(n)$; $E_{2a}(n)$</p> <p style="text-align: center;">Nouvelles données à l'instant $n + 1$:</p> <p>Entrée : $x_1(n + 1)$, $x_2(n + 1)$; référence : $y(n + 1)$</p> <p style="text-align: center;">Partie prédiction :</p> $e_{1a}(n + 1) = x_1(n + 1) - A_{1,2M}^t(n)X_{2M}(n)$ $A_{1,2M}(n + 1) = A_{1,2M}(n) + G_{2M}(n)e_{1a}(n + 1)$ $\varepsilon_{1a}(n + 1) = x_1(n + 1) - A_{1,2M}^t(n + 1)X_{2M}(n)$ $E_{1a}(n + 1) = \lambda E_{1a}(n) + e_{1a}(n + 1)\varepsilon_{1a}(n + 1)$ $G_{1,2M+1}(n + 1) = \begin{bmatrix} 0 \\ G_{2M}(n) \end{bmatrix} + \frac{\varepsilon_{1a}(n + 1)}{E_{1a}(n + 1)} \begin{bmatrix} 1 \\ -A_{1,2M}(n + 1) \end{bmatrix} = \begin{bmatrix} M_1(n + 1) \\ m_1(n + 1) \end{bmatrix}$ $e_{1b}(n + 1) = x_1(n + 1 - M) - B_{1,2M}^t(n)X_{1,2M}(n + 1)$ $G_{1,2M}(n + 1) = \frac{1}{1 - m_1(n + 1)e_{1b}(n + 1)} [M_1(n + 1) + m_1(n + 1)B_{1,2M}(n)]$ $B_{1,2M}(n + 1) = B_{1,2M}(n) + G_{1,2M}(n + 1)e_{1b}(n + 1)$ $e_{2a}(n + 1) = x_2(n + 1) - A_{2,2M}^t(n + 1)X_{1,2M}(n + 1)$ $A_{2,2M}(n + 1) = A_{2,2M}(n) + G_{1,2M}(n + 1)e_{2a}(n + 1)$ $\varepsilon_{2a}(n + 1) = x_2(n + 1) - A_{2,2M}^t(n + 1)X_{1,2M}(n + 1)$ $E_{2a}(n + 1) = \lambda E_{2a}(n) + e_{2a}(n + 1)\varepsilon_{2a}(n + 1)$ $G_{2,2M+1}(n + 1) = \begin{bmatrix} 0 \\ G_{1,2M}(n + 1) \end{bmatrix} + \frac{\varepsilon_{2a}(n + 1)}{E_{2a}(n + 1)} \begin{bmatrix} 1 \\ -A_{2,2M}(n + 1) \end{bmatrix} = \begin{bmatrix} M_2(n + 1) \\ m_2(n + 1) \end{bmatrix}$ $e_{2b}(n + 1) = x_2(n + 1 - M) - B_{2,2M}^t(n)X_{2M}(n + 1)$ $G_{2M}(n + 1) = \frac{1}{1 - m_2(n + 1)e_{2b}(n + 1)} [M_2(n + 1) - m_2(n + 1)B_{2,2M}(n)]$ $B_{2,2M}(n + 1) = B_{2,2M}(n) + G_{2M}(n + 1)e_{2b}(n + 1)$ $e(n + 1)y(n + 1) - \theta_{2M}^t(n)X(n + 1)$ $\theta_{2M}(n + 1) = \theta_{2M}(n) + G_{2M}(n + 1)e(n + 1)$ <p style="text-align: center;">Partie filtrage :</p> $e(n + 1)y(n + 1) - \theta_{2M}^t(n)X(n + 1)$ $\theta_{2M}(n + 1) = \theta_{2M}(n) + G_{2M}(n + 1)e(n + 1)$

Tableau 4.3. Algorithme MCR 2D/1D pour filtrage adaptatif avec décimation par 2.

En reprenant la décomposition de $R_{M+1}(n)$, il vient :

$$\begin{bmatrix} R_M(n) & r_M^b(n) \\ [r_M^b(n)]^t & R_1(n - M) \end{bmatrix} \begin{bmatrix} 1 \\ -A_{M-1}(n) \\ 0 \end{bmatrix} = \begin{bmatrix} E_{\alpha(M-1)}(n) \\ 0 \\ K_M(n) \end{bmatrix} \quad (4.119)$$

où la variable $K_M(n)$ dans la dernière ligne, a pour expression :

$$K_M(n) = \sum_{p=1}^n \lambda^{n-p} x(p)x(p-M) - A_{M-1}^t(n)R_{M-1}(n-1)B_{M-1}(n-1) \quad (4.120)$$

Dans cette expression, les coefficients de prédiction avant et arrière apparaissent de manière équilibrée. Il en résulte que la même variable $K_M(n)$ figure aussi dans l'équation matricielle de la prédiction arrière :

$$\begin{bmatrix} R_1(n) & [r_M^a(n)]^t \\ r_M^a(n) & R_M(n-1) \end{bmatrix} \begin{bmatrix} 0 \\ -B_{M-1}(n-1) \\ 1 \end{bmatrix} = \begin{bmatrix} K_M(n) \\ 0 \\ E_{b(M-1)}(n-1) \end{bmatrix} \quad (4.121)$$

comme on le vérifie directement en explicitant la première ligne. En multipliant les deux membres par le scalaire $K_M(n) / E_{b(M-1)}(n-1)$, on obtient :

$$R_{M+1}(n) \begin{bmatrix} 0 \\ \begin{bmatrix} -B_{M-1}(n-1) \\ 1 \end{bmatrix} \frac{K_M(n)}{E_{b(M-1)}(n-1)} \end{bmatrix} = \begin{bmatrix} \frac{K_M^2(n)}{E_{b(M-1)}(n-1)} \\ 0 \\ K_M(n) \end{bmatrix} \quad (4.122)$$

et il est alors intéressant de soustraire l'équation (4.122) de l'équation (4.119) pour se ramener à l'équation de la prédiction avant (4.117) d'ordre M . L'identification fournit la relation de récurrence suivante pour les vecteurs des coefficients de prédiction avant :

$$A_M(n) = \begin{bmatrix} A_{M-1}(n) \\ 0 \end{bmatrix} - \frac{K_M(n)}{E_{b(M-1)}(n-1)} \begin{bmatrix} B_{M-1}(n-1) \\ -1 \end{bmatrix} \quad (4.123)$$

La première ligne donne une récurrence sur l'ordre pour les énergies d'erreurs de prédiction avant :

$$E_{aM}(n) = E_{a(M-1)}(n) - \frac{K_M^2(n)}{E_{b(M-1)}(n-1)} \quad (4.124)$$

La même technique s'applique aux équations de la prédiction arrière. L'équation matricielle (4.119) peut être réécrite :

$$R_{M+1}(n) \begin{bmatrix} \begin{bmatrix} 1 \\ -A_{M-1}(n) \end{bmatrix} \frac{K_M(n)}{E_{a(M-1)}(n)} \\ 0 \end{bmatrix} = \begin{bmatrix} K_M(n) \\ 0 \\ \frac{K_M^2(n)}{E_{a(M-1)}(n)} \end{bmatrix} \quad (4.125)$$

En soustrayant l'équation (4.125) de l'équation (4.121) et en identifiant à l'équation de la prédiction arrière (4.118) à l'ordre M , on obtient les relations de récurrence pour les vecteurs des coefficients de prédiction arrière :

$$B_M(n) = \begin{bmatrix} 0 \\ B_{M-1}(n-1) \end{bmatrix} - \frac{K_M(n)}{E_{a(M-1)}(n)} \begin{bmatrix} -1 \\ A_{M-1}(n) \end{bmatrix} \quad (4.126)$$

ainsi que la récurrence sur l'ordre pour l'énergie d'erreur de prédiction arrière :

$$E_{bM}(n) = E_{b(M-1)}(n-1) - \frac{K_M^2(n)}{E_{a(M-1)}(n)} \quad (4.127)$$

Les définitions de l'erreur de prédiction avant *a priori* :

$$e_{aM}(n+1) = x(n+1) - A_M^t(n)X(n)$$

et de l'erreur arrière :

$$e_{bM}(n+1) = x(n+1-M) - B_M^t(n)X(n+1)$$

en liaison avec les récurrences (4.123) et (4.126), conduisent à la structure de prédicteur en treillis, qui relie les erreurs aux ordres M et $M-1$

$$e_{aM}(n+1) = e_{a(M-1)}(n+1) - \frac{K_M(n)}{E_{b(M-1)}(n-1)} e_{b(M-1)}(n) \quad (4.128)$$

et

$$e_{bM}(n+1) = e_{b(M-1)}(n) - \frac{K_M(n)}{E_{a(M-1)}(n)} e_{a(M-1)}(n+1) \quad (4.129)$$

De même, avec les erreurs *a posteriori* :

$$\varepsilon_{aM}(n+1) = x(n+1) - A_M^t(n+1)X(n)$$

et

$$\varepsilon_{bM}(n+1) = x(n+1-M) - B_M^t(n+1)X(n+1)$$

les opérations du treillis s'écrivent :

$$\varepsilon_{aM}(n+1) = \varepsilon_{a(M-1)}(n+1) - k_{bM}(n+1)\varepsilon_{b(M-1)}(n) \quad (4.130)$$

et

$$\varepsilon_{bM}(n+1) = \varepsilon_{b(M-1)}(n) - K_{aM}(n+1)\varepsilon_{a(M-1)}(n+1) \quad (4.131)$$

où les variables $k_{aM}(n+1)$ et $k_{bM}(n+1)$, données par :

$$K_{aM}(n+1) = \frac{K_M(n+1)}{E_{a(M-1)}(n+1)} ; \quad K_{bM}(n+1) = \frac{K_M(n+1)}{E_{b(M-1)}(n)} \quad (4.132)$$

sont les estimations de corrélation partielle, ou coefficients de réflexion.

Des récurrences sur l'ordre semblables peuvent être obtenues pour les coefficients des filtres adaptatifs, le gain d'adaptation et le rapport des erreurs *a priori* et *a posteriori*.

4.3.2. Relations de récurrence sur l'ordre pour les coefficients du filtre

Un filtre adaptatif à M coefficients produit le signal d'erreur de sortie :

$$e_M(n+1) = y(n+1) - \theta_M^t(n)X(n+1) \quad (4.133)$$

Le vecteur des coefficients $\theta_M(n)$ qui minimisent l'énergie de cette erreur au temps n s'écrit :

$$\theta_M(n) = R_M^{-1}(n)r_{yxM}(n) \quad (4.134)$$

avec :

$$r_{yxM}(n) = \sum_{p=1}^n \lambda^{n-p} y(p) X_M(p)$$

Pour un filtre à $M+1$ coefficients, les équations sont :

$$e_{M+1}(n+1) = y(n+1) - \theta_{M+1}^t(n)X_{M+1}(n+1)$$

et

$$R_{M+1}(n)\theta_{M+1}(n) = \begin{bmatrix} r_{yxM}(n) \\ \sum_{p=1}^n \lambda^{n-p} y(p)x(p-M) \end{bmatrix} \quad (4.135)$$

Le vecteur des coefficients $\theta_{M+1}(n)$ s'obtient à partir de $\theta_M(n)$ à l'aide de la décomposition de la matrice AC du signal d'entrée. Comme précédemment, on considère l'équation :

$$\begin{aligned} R_{M+1}(n) \begin{bmatrix} \theta_M(n) \\ 0 \end{bmatrix} &= \begin{bmatrix} R_M(n) & r_M^b(n) \\ [r_M^b(n)]^t & R_1(n-M) \end{bmatrix} \begin{bmatrix} \theta_M(n) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} r_{yxM}(n) \\ [r_M^b(n)]^t \theta_M(n) \end{bmatrix} \end{aligned} \quad (4.136)$$

La dernière ligne s'écrit aussi :

$$[r_M^b(n)]^t \theta_M(n) = B_M^t(n)R_M(n)\theta_M(n) = B_M^t(n)r_{yxM}(n) \quad (4.137)$$

En soustrayant l'équation (4.136) de (4.135), il vient :

$$R_{M+1}(n) \left[\theta_{M+1}(n) - \begin{bmatrix} \theta_M(n) \\ 0 \end{bmatrix} \right] = \begin{bmatrix} 0 \\ K_{fM}(n) \end{bmatrix} \quad (4.138)$$

où

$$K_{fM}(n) = \sum_{p=1}^n \lambda^{n-p} y(p) [x(p-M) - B_M^t(n)X(p)] \quad (4.139)$$

L'identification de (4.138) avec l'équation de la prédiction arrière donne la récurrence suivante pour les coefficients du filtre :

$$\theta_{M+1}(M) = \begin{bmatrix} \theta_M(n) \\ 0 \end{bmatrix} - \frac{K_{fM}(n)}{E_{bM}(n)} \begin{bmatrix} B_M(n) \\ -1 \end{bmatrix} \quad (4.140)$$

En reportant (4.140) dans la définition de l'erreur, on fait apparaître une récurrence sur l'ordre pour les erreurs de sortie *a priori* :

$$e_{M+1}(n+1) = e_M(n+1) - \frac{K_{fM}(n)}{E_{bM}(n)} e_{bM}(n+1) \quad (4.141)$$

L'équation correspondante pour les erreurs *a posteriori* s'écrit :

$$\varepsilon_{M+1}(n+1) = \varepsilon_M(n+1) - \frac{K_{fM}(n+1)}{E_{bM}(n+1)} \varepsilon_{bM}(n+1) \quad (4.142)$$

L'ensemble des équations (4.128), (4.129) et (4.141) forme le filtre adaptatif en treillis *a priori*, alors que les équations (4.130), (4.131) et (4.142) donnent la version *a posteriori* de ce filtre.

L'énergie de l'erreur se calcule aussi par récurrence sur l'ordre. D'après la définition de l'erreur de sortie du filtre, on a :

$$E_{M+1}^{(n)} = \sum_{p=1}^n \lambda^{n-p} y^2(p) - \theta_{M+1}^t(n) R_{M+1}(n) \theta_{M+1}(n) \quad (4.143)$$

En reportant dans cette équation la récurrence (4.140) et en utilisant l'équation de la prédiction arrière, il vient :

$$E_{M+1}(n) = E_M(n) - \frac{K_{fM}^2(n)}{E_{bM}(n)} \quad (4.144)$$

Il est évident que $E_{M+1}(n) \leq E_M(n)$ et l'énergie de l'erreur décroît quand l'ordre du filtre croît, ce qui est conforme à la logique.

Au paragraphe 4.2.3, il a été montré que le gain d'adaptation du filtre transverse se calcule également par une récurrence sur l'ordre avec les variables de la prédiction arrière. La prédiction avant conduit à une récurrence mixte, à la fois sur l'ordre et temporelle. Ces expressions peuvent être utilisées pour calculer par récurrence le rapport $\psi_M(n)$ des erreurs *a priori* et *a posteriori*, qui a été défini par :

$$\psi_M(n) = \frac{\varepsilon_M(n)}{e_M(n)} = 1 - X_M^t(n) R_M^{-1}(n) X_M(n) = 1 - X_M^t(n) G_M(n)$$

La substitution directe donne :

$$\psi_M(n) = \psi_{M-1}(n) - \frac{\varepsilon_{b(M-1)}^2(n)}{E_{b(M-1)}(n)} \quad (4.145)$$

et pour la cellule initiale, $M = 1$:

$$\psi_1(n) = \psi_0(n) - \frac{\varepsilon_{b0}^2(n)}{E_{b0}(n)} = 1 - \frac{x^2(n)}{\sum_{p=1}^n \lambda^{n-p} x^2(p)} \quad (4.146)$$

Donc, pour calculer $\psi_M(n)$ par récurrence, il suffit de prendre $\psi_0(n) = 1$ et d'itérer l'équation (4.145). Dans les algorithmes en treillis, cette variable a un rôle très particulier, car elle établit un lien essentiel entre les récurrences dans le temps et sur l'ordre.

4.3.3. Relations de récurrence temporelle

Pour un filtre d'ordre fixe M , la variable du treillis $k_M(n)$ se calcule par récurrence dans le temps. En effet, d'après la définition (4.120) il vient :

$$K_{M+1}(n+1) = \lambda \sum_{p=1}^n \lambda^{n-p} x(p)x(p-M-1) + x(n+1)x(n-M) - A_M^t(n+1)R_M B_M(n) \quad (4.147)$$

En utilisant les relations de récurrence établies précédemment pour $A_M(n+1)$, $R_M(n)$ et $B_M(n)$, on obtient, après un certain nombre de manipulations algébriques, la mise à jour suivante :

$$K_{M+1}(n+1) = \lambda K_{M+1}(n) + e_{aM}(n+1)\varepsilon_{bM}(n) \quad (4.148)$$

A noter, que les relations entre les erreurs *a priori* et *a posteriori* fournissent la relation duale :

$$K_{M+1}(n+1) = \lambda K_{M+1}(n) + \varepsilon_{aM}(n+1)e_{bM}(n)$$

Il est clair que la variable $K_{M+1}(n)$ représente une estimation de l'intercorrélation entre les erreurs de prédiction avant et arrière à l'ordre M . Evidemment, l'équation (4.148) est semblable aux relations obtenues pour la mise à jour des énergies d'erreur de prédiction et exploitées dans les algorithmes pour filtre transverse. Quant à l'énergie de l'erreur de sortie du filtre, $E_M(n)$, elle satisfait bien entendu une relation du même type. L'équation (4.143) à l'ordre M et au temps $n+1$ correspond à :

$$E_M(n+1) = \sum_{p=1}^{n+1} \lambda^{n+1-p} y^2(p) - \theta_M^t(n+1)R_M(n+1)\theta_M(n+1) \quad (4.149)$$

En reportant l'équation de mise à jour des coefficients :

$$\theta_M(n+1) = \theta_M(n) + G_M(n+1)e_M(n+1) \quad (4.150)$$

dans l'équation (4.149) et en simplifiant, il vient :

$$E_M(n+1) = \lambda E_M(n) + e_M(n+1)\varepsilon_M(n+1) \quad (4.151)$$

De même, pour la variable de la partie filtrage $K_{fM}(n+1)$, la définition (4.139) peut être réécrite :

$$K_{fM}(n+1) = \sum_{p=1}^{n+1} \lambda^{n+1-p} y(p)x(p-M) - [B_M^t(n) + G_M^t(n+1)e_{bM}(n+1)] \cdot \left[\begin{array}{c} \lambda r_{yxM}(n) + \\ y(n+1)X_M(n+1) \end{array} \right] \quad (4.152)$$

ce qui, toutes simplifications faites, donne :

$$K_{fM}(n+1) = \lambda K_{fM} + \varepsilon_{bM}(n+1)e_M(n+1) \quad (4.153)$$

A noter que la variable $K_{fM}(n+1)$, qui, d'après la définition (4.139), représente une estimation de l'intercorrélation entre le signal de référence et l'erreur de prédiction signal de référence et l'erreur de prédiction arrière, peut se calculer comme une estimation de l'intercorrélation entre l'erreur de sortie du filtre et l'erreur de prédiction arrière. Cette propriété est une conséquence de la décorrélation entre les erreurs de prédiction et le vecteur des données.

Les relations de récurrence ci-dessus peuvent servir à élaborer des algorithmes de moindres carrés rapides pour les filtres à structure en treillis.

4.3.4. Algorithmes MCR pour les structures en treillis

Les algorithmes combinent les récurrences dans le temps et sur l'ordre, pour calculer, quand un ensemble de nouvelles valeurs des signaux d'entrée et de référence devient disponible, les coefficients du treillis, les erreurs de prédiction et de sortie du filtre, ainsi que leurs énergies et intercorrélation. Dans un filtre d'ordre M , les opérations se répartissent en deux ensembles, pour la prédiction et le filtrage.

D'abord, il convient d'examiner l'initialisation. Comme il y a deux types de récurrences, il faut distinguer deux types d'initialisations. Les initialisations des récurrences sur l'ordre sont obtenues d'une manière directe : les erreurs de prédiction ont comme valeur initiale le nouvel échantillon du signal d'entrée, les énergies d'erreur de prédiction, la puissance de ce signal et la variable $\psi_0(n)$ est mise à un. Quant aux récurrences temporelles, la méthode pour initialiser les variables d'état du filtre en

treillis d'ordre M consiste à généraliser la méthode de l'équation (4.93). Le signal d'entrée pour $n \leq 0$ est supposé constitué d'une impulsion unique au temps $-M$, ce qui donne :

$$\begin{aligned} e_{ai}(0) &= e_{bi}(0) = \varepsilon_{ai}(0) = \varepsilon_{bi}(0) = 0 ; & 0 \leq i \leq M-1 \\ E_{ai}(0) &= \lambda^M E_0 & ; & 0 \leq i \leq M-1 \\ E_{bi}(0) &= \lambda^{M-i} E_0 & ; & 0 \leq i \leq M-1 \\ K_i(0) &= 0 & ; & 1 \leq i \leq M \end{aligned} \quad (4.154)$$

où E_0 est un scalaire réel positif. On peut vérifier que les récurrences sur l'ordre de la partie prédiction et notamment les relations sur les énergies (4.124) et (4.127) sont satisfaites pour $n = 0$. Bien sûr, dans ces conditions, l'impact du choix de la valeur E_0 , énergie d'erreur initiale, sur les performances du filtre est le même qu'en structure transversale et les résultats sont toujours valables.

De nombreux algorithmes plus ou moins différents peuvent être élaborés à partir des récurrences de base, suivant le choix des variables internes et selon que la préférence est donnée au calcul d'erreurs *a priori* ou *a posteriori* et aux récurrences dans le temps ou sur l'ordre. En fait, il existe des règles générales pour concevoir des algorithmes robustes et efficaces, les principales s'énoncent comme suit :

- limiter au minimum le nombre des variables d'état
- donner la préférence aux récurrences temporelles
- s'assurer que des variables de contrôle fiables sont disponibles, afin d'être capable de vérifier en permanence le fonctionnement correct du filtre.

Ainsi l'algorithme du treillis donné ci-après évite de faire appel aux variables d'intercorrélation $K_i(n)$ et il est basé sur la mise à jour directe des coefficients de réflexion.

En incorporant la récurrence temporelle (4.148) et l'équation de mise à jour de l'énergie d'erreur dans la définition (4.132), il vient :

$$\begin{aligned} [E_{ai}(n+1) - e_{ai}(n+1)\varepsilon_{ai}(n+1)] K_{a(i+1)}(n) = \\ K_{i+1}(n+1) - \varepsilon_{ai}(n+1)e_{bi}(n) \end{aligned} \quad (4.155)$$

D'où, en appliquant encore la définition (4.132) au temps $n+1$:

$$K_{a(i+1)}(n+1) = K_{a(i+1)}(n) + \frac{\varepsilon_{ai}(n+1)}{E_{ai}(n+1)} [e_{ai}(n) - K_{a(i+1)}(n)e_{ai}(n+1)] \quad (4.156)$$

Ensuite, la récurrence temporelle (4.129) donne :

$$K_{a(i+1)}(n+1) = K_{a(i+1)}(n) + \frac{\varepsilon_{ai}(n+1)e_{b(i+1)}(n+1)}{E_{ai}(n+1)} \quad (4.157)$$

ce qui correspond à une mise à jour des coefficients de réflexion uniquement avec les variables d'erreur. La même méthode, en faisant appel aux récurrences temporelles (4.148) et (4.128), aboutit à la mise à jour des autres coefficients de réflexion de la partie prédiction :

$$K_{b(i+1)}(n+1) = K_{b(i+1)}(n) + \frac{\varepsilon_{bi}(n)e_{a(i+1)}(n+1)}{E_{bi}(n)} \quad (4.158)$$

Quant à la partie filtrage, en posant :

$$k_{fM}(n) = \frac{K_{fM}(n)}{E_{bM}(n)} \quad (4.159)$$

la même méthode, faisant appel à la récurrence temporelle (4.138) et à la relation de mise à jour de l'énergie d'erreur, conduit à :

$$K_{fi}(n+1) = K_{fi}(n) + \frac{\varepsilon_{bi}(n+1)e_{i+1}(n+1)}{E_{bi}(n+1)} \quad (4.160)$$

L'agencement des opérations du filtre adaptatif en treillis basé sur les erreurs *a priori* est donné dans le Tableau 4.4.

Un algorithme en treillis basé sur les erreurs *a posteriori* peut être obtenu de façon analogue [Ling 86].

La complexité de l'algorithme ci-dessus s'élève à $16M + 2$ multiplications et $3M$ divisions (calculs d'inverse). Il faut environ $7M$ mémoires. Dans cet algorithme, la partie filtrage est parfois, en raison de sa forme, dite en échelle et l'ensemble est appelé filtre adaptatif en structure treillis-échelle.

Les relations qui ont abouti à cet algorithme font intervenir des élévations au carré et des multiplications de variables. Il en résulte une extension de la dynamique des signaux et une complexité accrue des circuits. Le développement d'algorithmes basés sur des variables normalisées permet une réduction de la dynamique des signaux [Lee 81].

4.3.5. Algorithmes pour treillis avec normalisation

La variable $K_i(n)$ définie par l'équation (4.120) et mise à jour par la récurrence (4.148) correspond à un calcul d'intercorrélation. En fait, le vrai coefficient d'intercorrélation, dont la valeur se trouve dans l'intervalle $[-1,1]$, est obtenu en divisant cette variable par les énergies des signaux d'erreur, ce qui conduit à la variable normalisée $k_i(n)$ définie par :

$$k_{i+1}(n) = \frac{K_{i+1}(n)}{\sqrt{E_{ai}(n)E_{bi}(n-1)}} \quad (4.162)$$

ALGORITHME M.C.R.T.

Conditions initiales

$$\begin{aligned}
 e_{bi}(o) = k_{ai}(o) = k_{bi}(o) = k_{fi}(o) = 0 ; \quad 0 \leq i \leq M-1 \\
 \psi_i(o) = 1 ; \quad E_{ai}(0) = \lambda^M E_0 ; \quad E_{bi}(0) = \lambda^{M-i} E_o ; \quad 0 \leq i \leq M-1
 \end{aligned} \quad (4.161)$$

DISPONIBLE AU TEMPS n :

 Vecteurs des coefficients de réflexion : $K_a(n), K_b(n)$

 Vecteur des coefficients du filtre : $K_f(n)$

 Vecteur des erreurs de prédiction arrière : $|e_b(n)|$

 Vecteurs des énergies d'erreurs de prédiction : $|E_a(n)|, |E_b(n)|$

 Facteur de pondération : λ
NOUVELLES DONNEES AU TEMPS n :

 Signal d'entrée : $x(n+1)$; Référence $y(n+1)$
INITIALISATIONS :

$$\begin{aligned}
 e_{ao}(n+1) = e_{bo}(n+1) = x(n+1) ; \quad e_o(n+1) = y(n+1) \\
 \psi_o(n+1) = 1 ; \quad E_{ao}(n+1) = E_{bo}(n+1) = \lambda E_{ao}(n) + x^2(n+1)
 \end{aligned}$$

$$0 \leq i \leq M-1$$

PARTIE PREDICTION :

$$\begin{aligned}
 e_{a(i+1)}(n+1) &= e_{ai}(n+1) - k_{b(i+1)}(n) e_{bi}(n) \\
 e_{b(i+1)}(n+1) &= e_{bi}(n) - k_{a(i+1)}(n) e_{ai}(n+1) \\
 k_{a(i+1)}(n+1) &= k_{a(i+1)}(n) + e_{ai}(n+1) \psi_i(n) e_{b(i+1)}(n+1) / E_{ai}(n+1) \\
 k_{b(i+1)}(n+1) &= k_{b(i+1)}(n) + e_{a(i+1)}(n+1) e_{bi}(n) \psi_i(n) / E_{bi}(n) \\
 E_{a(i+1)}(n+1) &= \lambda E_{a(i+1)}(n) + e_{a(i+1)}^2(n+1) \psi_{i+1}(n) \\
 \psi_{i+1}(n+1) &= \psi_i(n+1) - \psi_i^2(n+1) e_{bi}^2(n+1) / E_{bi}(n+1) \\
 E_{b(i+1)}(n+1) &= \lambda E_{b(i+1)}(n) + e_{b(i+1)}^2(n+1) \psi_{i+1}(n+1)
 \end{aligned}$$

PARTIE FILTRE :

$$\begin{aligned}
 e_{i+1}(n+1) &= e_i(n+1) - k_{fi}(n) e_{bi}(n+1) \\
 k_{fi}(n+1) &= k_{fi}(n) + e_{bi}(n+1) e_{i+1}(n+1) \psi_i(n+1) / E_{bi}(n+1)
 \end{aligned}$$

Tableau 4.4. Les opérations d'un filtre adaptatif en treillis

Une relation de récurrence dans le temps est obtenue, à partir de la relation (4.148) par :

$$k_{i+1}(n+1) = [E_{ai}(n+1)]^{-\frac{1}{2}} [\lambda K_{i+1}(n) + e_{ai}(n+1) \varepsilon_{bi}(n)] [E_{bi}(n)]^{-\frac{1}{2}} \quad (4.163)$$

Pour faire apparaître $k_{i+1}(n)$ dans cette expression, il faut faire appel aux rapports des énergies d'erreurs. Les équations de mise à jour se réécrivent :

$$\lambda \frac{E_{ai}(n)}{E_{ai}(n+1)} = 1 - \frac{e_{ai}^2(n+1)}{E_{ai}(n+1)} \psi_i(n) \quad (4.164)$$

et :

$$\lambda \frac{E_{bi}(n-1)}{E_{bi}(n)} = 1 - \frac{\varepsilon_{bi}^2(n)}{E_{bi}(n)} \frac{1}{\psi_i(n)} \quad (4.165)$$

En définissant l'erreur de prédiction avant normalisée par :

$$e_{nai}(n+1) = e_{ai}(n+1) \sqrt{\frac{\psi_i(n)}{E_{ai}(n+1)}} = \varepsilon_{ai}(n+1) [\psi_i(n) E_{ai}(n+1)]^{-1} \quad (4.166)$$

et l'erreur de prédiction arrière normalisée par :

$$e_{nbi}(n) = \varepsilon_{nbi}(n) [\psi_i(n) E_{bi}(n)]^{-\frac{1}{2}} \quad (4.167)$$

la récurrence (4.163) devient :

$$k_{i+1}(n+1) = k_{i+1}(n) [(1 - e_{nai}^2(n+1)) (1 - e_{nbi}^2(n))]^{\frac{1}{2}} + e_{nai}(n+1) e_{nbi}(n) \quad (4.168)$$

Il est clair, avec les définitions, que les variables d'erreur normalisées sont intermédiaire entre les erreurs *a priori* et *a posteriori*. Pour faire un algorithme, il faut établir des récurrences sur les erreurs de prédiction normalisées. La récurrence sur l'ordre (4.130) pour les erreurs avant *a posteriori* se réécrit :

$$\psi_{i+1}(n) e_{a(i+1)}(n+1) = \psi_i(n) e_{ai}(n+1) - \frac{K_{i+1}(n+1)}{E_{bi}(n)} \varepsilon_{bi}(n) \quad (4.169)$$

En reportant les erreurs normalisées dans cette équation, il vient :

$$e_{na(i+1)}(n+1) = \left[\frac{E_{ai}(n+1)}{E_{a(i+1)}(n+1)} \right]^{\frac{1}{2}} \left[\frac{\psi_i(n)}{\psi_{i+1}(n)} \right]^{\frac{1}{2}} e_{nai}(n+1) - k_{i+1}(n+1) e_{nbi}(n) \quad (4.170)$$

Les variables normalisées peuvent être introduites dans les récurrences sur l'ordre (4.124) et (4.145) ce qui donne :

$$E_{a(i+1)}(n+1) = E_{ai}(n+1) [1 - k_{i+1}^2(n+1)] \quad (4.171)$$

et :

$$\psi_{i+1}(n) = \psi_i(n) [1 - e_{nbi}^2(n)] \quad (4.172)$$

En reportant dans l'équation (4.170), on obtient la forme finale des relations de récurrence dans le temps pour les erreurs de prédiction avant normalisées :

$$e_{na(i+1)}(n+1) = \frac{e_{nai}(n+1) - k_{i+1}(n+1) e_{nbi}(n)}{\sqrt{[1 - k_{i+1}^2(n+1)] [1 - e_{nbi}^2(n)]}} \quad (4.173)$$

La même méthode s'applique aux erreurs de prédiction arrière. La récurrence sur l'ordre (4.129) s'exprime en fonction des variables normalisées par :

$$e_{nb(i+1)}(n+1) = \left[\frac{E_{bi}(n)}{E_{b(i+1)}(n+1)} \right]^{\frac{1}{2}} \left[\frac{\psi_i(n)}{\psi_{i+1}(n+1)} \right]^{\frac{1}{2}} [e_{nbi}(n) - k_{i+1}(n+1)e_{nai}(n+1)] \quad (4.174)$$

et l'équation (11) pour l'énergie par :

$$E_{b(i+1)}(n+1) = E_{bi}(ni) [1 - k_{i+1}^2(n+1)] \quad (4.175)$$

Une équation reliant $\psi_{i+1}(n+1)$ et $\psi_i(n)$ peut être obtenue à l'aide de la récurrence sur le gain d'adaptation des algorithmes en structure transverse, ce qui donne :

$$\psi_{i+1}(n+1) = \psi_i(n) - \frac{\varepsilon_{ai}^2(n+1)}{E_{ai}(n+1)} \quad (4.176)$$

et donc

$$\psi_{i+1}(n+1) = \psi_i(n) [1 - e_{nai}^2(n+1)] \quad (4.177)$$

D'où la forme finale de la récurrence temporelle pour l'erreur de prédiction arrière normalisée :

$$e_{nb(i+1)}(n+1) = \frac{e_{nbi}(n) - k_{i+1}(n+1)e_{nai}(n+1)}{\sqrt{[1 - k_{i+1}^2(n+1)] [1 - e_{nai}^2(n+1)]}} \quad (4.178)$$

Finalement, les équations (4.168), (4.173) et (4.178) constituent un algorithme pour prédicteur adaptatif en treillis normalisé.

Les variables normalisées s'introduisent aussi dans la section filtrage. Les erreurs de sortie du filtre normalisées sont définies par :

$$e_{ni}(n) = e_i(n) \left[\frac{\psi_i(n)}{E_i(n)} \right]^{\frac{1}{2}} = \varepsilon_i(n) [\psi_i(n)E_i(n)]^{-\frac{1}{2}} \quad (4.179)$$

Alors la récurrence sur l'ordre (4.141) donne :

$$e_{n(i+1)}(n) = \left[\frac{E_i(n)}{E_{i+1}(n)} \right]^{\frac{1}{2}} \left[\frac{\psi_i(n)}{\psi_{i+1}(n)} \right]^{\frac{1}{2}} \left[e_{ni}(n) - \frac{K_{fi}(n)}{\sqrt{E_i(n)\psi_i(n)}} \frac{\varepsilon_{bi}(n)}{E_{bi}(n)} \right] \quad (4.180)$$

En définissant les coefficients normalisés par :

$$k_{fi}(n) = \frac{K_{fi}(n)}{\sqrt{E_{bi}(n)E_i(n)}} \quad (4.181)$$

la récurrence sur l'ordre (4.144) pour les énergies d'erreurs se réécrit :

$$E_{i+1}(n) = E_i(n) [1 - k_{fi}^2(n)] \quad (4.182)$$

En reportant les relations (4.172) et (4.182) dans l'équation (4.180), on obtient la récurrence sur l'ordre pour les erreurs de sortie du filtre :

$$e_{n(i+1)}(n) = [1 - k_{fi}^2(n)]^{-\frac{1}{2}} [1 - e_{nbi}^2(n)]^{-\frac{1}{2}} [e_{ni}(n) - k_{fi}(n)e_{nbi}(n)] \quad (4.183)$$

Ensuite, il faut calculer les coefficients normalisés eux-mêmes. En introduisant les variables normalisées dans les récurrences temporelles (4.153), il vient :

$$k_{fi}(n+1) = \left[\frac{E_{bi}(n)}{E_{bi}(n+1)} \right]^{\frac{1}{2}} \left[\frac{E_i(n)}{E_i(n+1)} \right]^{\frac{1}{2}} \lambda k_{fi}(n) + e_{nbi}(n+1)e_{ni}(n+1) \quad (4.184)$$

La récurrence temporelle pour les énergies d'erreurs de sortie du filtre se réécrit :

$$\lambda \frac{E_i(n)}{E_i(n+1)} = 1 - \frac{e_i^2(n+1)\psi_i(n+1)}{E_i(n+1)} = 1 - e_{ni}^2(n+1) \quad (4.185)$$

En reportant les relations (4.165) et (4.185), on aboutit à la récurrence temporelle pour les coefficients normalisés :

$$k_{fi}(n+1) = k_{fi}(n) [1 - e_{nbi}^2(n+1)]^{\frac{1}{2}} [1 - e_{ni}^2(n+1)]^{\frac{1}{2}} + e_{nbi}(n+1)e_{ni}(n+1) \quad (4.186)$$

qui complète l'algorithme pour le filtre en treillis normalisé. Les initialisations se font selon la définition des variables normalisées, ce qui implique pour la prédiction :

$$e_{na0}(n+1) = \frac{x(n+1)}{\sqrt{E_{a0}(n+1)}} = e_{nb0}(n+1) \quad (4.187)$$

et pour la partie filtrage :

$$e_{n0}(n+1) = \frac{y(n+1)}{\sqrt{E_{f0}(n+1)}} \quad ; \quad E_{f0}(n+1) = \lambda E_{f0}(n) + y^2(n+1) \quad (4.188)$$

Pour les autres variables, on utilise les initialisations (4.161) avec l'équation supplémentaire :

$$E_{f0}(0) = E_0$$

L'agencement des calculs du filtre adaptatif en treillis normalisé est donné dans le [Tableau 4.5] p.183.

Bien qu'il se présente sous une forme très concise, cet algorithme nécessite davantage de calculs que la version non normalisée. Il faut, dans la partie prédiction, $10M+2$ multiplications, $2M+1$ divisions et $3M+1$ racines carrées, et dans la partie filtrage,

ALGORITHME M.C.R.T.N.
DISPONIBLE AU TEMPS n :

 Vecteur des coefficients de réflexion : $K(n)$

 Vecteur des coefficients du filtre : $K_f(n)$

 Vecteur des erreurs de prédiction arrière : $[e_i(n)]$

 Energies des signaux : E_{ao} , E_{fo}
NOUVELLES DONNEES AU TEMPS n :

 Signal d'entrée : $x(n+1)$; Référence : $y(n+1)$
INITIALISATION :

$$\begin{aligned} E_{ao}(n+1) &= \lambda E_{ao}(n) + x^2(n+1) \\ e_{nao}(n+1) &= e_{nbo}(n+1) = x(n+1) / [E_{ao}(n+1)]^{1/2} \\ E_{fo}(n+1) &= \lambda E_{fo}(n) + y^2(n+1) \\ e_{no}(n+1) &= y(n+1) / [E_{fo}(n+1)]^{1/2} \end{aligned}$$

$$0 \leq i \leq M-1$$

PARTIE PREDICTION :

$$\begin{aligned} k_{i+1}(n+1) &= k_{i+1}(n) [(1 - e_{nai}^2(n+1))(1 - e_{nbi}^2(n))]^{1/2} + e_{nai}(n+1)e_{nbi}(n) \\ e_{na(i+1)}(n+1) &= \frac{[e_{nai}(n+1) - k_{i+1}(n+1)e_{nbi}(n)]}{(1 - k_{i+1}^2(n+1))^{1/2}(1 - e_{nbi}^2(n))^{1/2}} \\ e_{nb(i+1)}(n+1) &= \frac{[e_{nbi}(n) - k_{i+1}(n+1)e_{nai}(n+1)]}{(1 - k_{i+1}^2(n+1))^{1/2}(1 - e_{nai}^2(n+1))^{1/2}} \end{aligned}$$

PARTIE FILTRE :

$$\begin{aligned} k_{fi}(n+1) &= k_{fi}(n)(1 - e_{nbi}^2(n+1))^{1/2}(1 - e_{nai}^2(n+1))^{1/2} + e_{nbi}(n+1)e_{nai}(n+1) \\ e_{n(i+1)}(n+1) &= \frac{[e_{nai}(n+1) - k_{fi}(n+1)e_{nbi}(n+1)]}{(1 - k_{fi}^2(n+1))^{1/2}(1 - e_{nbi}^2(n+1))^{1/2}} \end{aligned}$$

Tableau 4.5. Les opérations d'un filtre adaptatif en treillis normalisé

$6M + 2$ multiplications, $M + 1$ divisions et $2M + 1$ racines carrées. Au total donc, la quantité de calculs de l'algorithme normalisé s'élève à $16M + 4$ multiplications, $3M + 2$ divisions et $5M + 2$ racines carrées. Evidemment, la présence de racines carrées est un aspect important pour la réalisation en raison de la complexité de cette opération. La quantité de mémoires est d'environ $3M$.

L'algorithme normalisé peut être intéressant pour traiter les signaux non stationnaires en arithmétique à virgule fixe, car il comporte un cadrage automatique de ses variables. La robustesse aux erreurs d'arrondi qui en résulte se trouve renforcée par le fait qu'un seul ensemble de coefficients de prédiction est calculé [Samson 83].

Sur le plan général, un avantage essentiel de la structure en treillis réside dans le fait qu'elle conduit à mettre en œuvre un ensemble de M filtres adaptatifs, avec tous

les ordres de 1 à M . Dans ces conditions il est intéressant d'établir des relations avec les coefficients et les gains d'adaptation des filtres transversaux correspondants.

4.3.6. Calcul des coefficients des filtres transversaux

La conversion entre les coefficients de prédiction des filtres en structures en treillis et transversale repose sur les relations de récurrence sur l'ordre (4.123) :

$$\begin{aligned} A_{i+1}(n+1) &= \begin{bmatrix} A_i(n+1) \\ 0 \end{bmatrix} - K_{b(i+1)}(n+1) \begin{bmatrix} B_i(n) \\ -1 \end{bmatrix} \\ B_{i+1}(n+1) &= \begin{bmatrix} 0 \\ B_i(n) \end{bmatrix} - K_{a(i+1)}(n+1) \begin{bmatrix} -1 \\ A_i(n+1) \end{bmatrix} \end{aligned} \quad (4.189)$$

Alors les coefficients des filtres transversaux se calculent par récurrence de l'ordre 2 à l'ordre M . Cependant, il peut être commode de remplacer $B_i(n)$ par $B_i(n+1)$ pour avoir des variables homogènes dans le temps.

En reportant les récurrences temporelles des coefficients de prédiction avant et arrière dans l'équation (4.189) et en ajoutant la récurrence temporelle pour le gain d'adaptation, le système d'équations pour la conversion devient :

$$\begin{aligned} A_{i+1}(n+1) &= \begin{bmatrix} A_i(n+1) \\ 0 \end{bmatrix} - k_{b(i+1)}(n+1) \begin{bmatrix} B_i(n+1) \\ -1 \end{bmatrix} \\ &+ k_{b(i+1)}(n+1) e_{bi}(n+1) \begin{bmatrix} G_i(n+1) \\ 0 \end{bmatrix} \\ B_{i+1}(n+1) &= \begin{bmatrix} 0 \\ B_i(n+1) \end{bmatrix} - e_{bi}(n+1) \begin{bmatrix} 0 \\ G_i(n+1) \end{bmatrix} \\ &- k_{a(i+1)}(n+1) \begin{bmatrix} -1 \\ A_i(n+1) \end{bmatrix} \\ G_{i+1}(n+1) &= \begin{bmatrix} G_i(n+1) \\ 0 \end{bmatrix} + \frac{\varepsilon_{bi}(n+1)}{E_{bi}(n+1)} \begin{bmatrix} -B_i(n+1) \\ 1 \end{bmatrix} \end{aligned}$$

Quant aux coefficients $\theta_i(n)$ de la partie filtrage transversal, ils sont obtenus par récurrence à partir de l'équation (4.140).

4.4. Algorithmes de décomposition QR

La rotation possède une propriété très intéressante en traitement numérique : elle préserve la norme des vecteurs et des signaux. Par suite, un algorithme à base de rotations conserve la dynamique du signal au lieu de la doubler comme les algorithmes avec calcul de moindres carrés.

De même que les algorithmes en treillis reposent sur une triangularisation de la matrice d'autocorrélation du signal reçu, les algorithmes à rotations reposent sur une

triangularisation de la matrice des échantillons d'entrée. En analyse numérique, l'opération est dite décomposition QR. Le problème des moindres carrés est alors résolu en deux phases. D'abord, une matrice orthogonale est utilisée pour transformer en une matrice triangulaire la matrice des échantillons reçus. Ensuite, les valeurs optimales des coefficients du filtre adaptatif sont obtenues en résolvant un système triangulaire d'équations.

Avant de décrire la décomposition QR, il peut être utile de présenter l'opération de rotation.

4.4.1. L'opération de rotation

Soit un vecteur X de données réelles à 3 éléments et deux rotations définies par deux matrices :

$$X = \begin{bmatrix} x_2 \\ x_1 \\ x_0 \end{bmatrix} ; R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_1 & -\sin \theta_1 \\ 0 & \sin \theta_1 & \cos \theta_1 \end{bmatrix} ; R_2 = \begin{bmatrix} \cos \theta_2 & 0 & -\sin \theta_1 \\ 0 & 1 & 0 \\ \sin \theta_1 & 0 & \cos \theta_1 \end{bmatrix} \quad (4.190)$$

Les matrices R_1 et R_2 sont choisies pour annuler les 2 éléments supérieurs du vecteur X . Il faut alors que :

$$\begin{bmatrix} \cos \theta_1 & -\sin \theta_1 \\ \sin \theta_1 & \cos \theta_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_0 \end{bmatrix} = \begin{bmatrix} 0 \\ s_1 \end{bmatrix} \quad (4.191)$$

où la variable $s_1 = \sqrt{x_1^2 + x_0^2}$ est la norme du vecteur de données qui est invariante dans l'opération. Par suite, il vient :

$$\cos \theta_1 = \frac{x_0}{s_1} ; \sin \theta_1 = \frac{x_1}{s_1} \quad (4.192)$$

La même procédure s'applique à l'élément x_2 , avec l'angle θ_2 :

$$\begin{bmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} x_2 \\ s_1 \end{bmatrix} = \begin{bmatrix} 0 \\ s_2 \end{bmatrix} \quad (4.193)$$

avec

$$s_2 = \sqrt{x_2^2 + s_1^2} ; \cos \theta_2 = \frac{s_1}{s_2} ; \sin \theta_2 = \frac{x_2}{s_2}$$

La nouvelle variable s_2 est la norme du vecteur X et l'opération globale est décrite par l'équation matricielle

$$\begin{bmatrix} 0 \\ 0 \\ s_2 \end{bmatrix} = R_2 R_1 \begin{bmatrix} x_2 \\ x_1 \\ x_0 \end{bmatrix} \quad (4.194)$$

Ainsi la norme d'un vecteur peut se calculer par itérations avec des rotations, définie par les éléments du vecteur. Dans ces rotations le cosinus est le rapport des normes et le sinus introduit la nouvelle donnée .

4.4.2. La décomposition QR

En filtrage adaptatif transverse par moindres carrés, avec M coefficients, la fonction coût a été définie par :

$$J(n) = \sum_{p=0}^n \lambda^{n-p} [y(p) - X^t(p) H(n)]^2 = \sum_{p=0}^n \lambda^{n-p} e^2(p) \quad (4.195)$$

Dans le calcul de l'ensemble des valeurs $e(p)$, qui forment un vecteur dont $J(n)$ est le carré de la norme, cette fonction coût fait intervenir la matrice de l'ensemble des données $X_M(n)$ définie par :

$$X_M(n) = \begin{bmatrix} x(n) & x(n-1) & x(n+1-M) \\ \lambda^{1/2}x(n-1) & \lambda^{1/2}x(n-2) & \lambda^{1/2}x(n-M) \\ \vdots & \vdots & \vdots \\ \lambda^{n/2}x(0) & 0 & 0 \end{bmatrix} \quad (4.196)$$

C'est une matrice à $(x+1) \times M$ éléments. Les données d'entrée sont supposées nulles pour $n < 0$.

La première phase de la décomposition QR consiste à multiplier cette matrice par une matrice de rotation, $Q_M(n)$, pour réduire les éléments non nuls à une matrice triangulaire. Deux approches sont possibles pour atteindre cet objectif, correspondant à la prédiction linéaire avant et arrière [Cioffi 90, Proudler 90]. Avec la prédiction avant, on obtient :

$$Q_M(n) X_M(n) = \begin{bmatrix} 0 & \dots & \dots & 0 \\ \vdots & & & \vdots \\ 0 & \dots & \dots & 0 \\ & S_M(n) & & \vdots \\ & \dots & \dots & 0 \end{bmatrix} \quad (4.197)$$

et avec la prédiction arrière :

$$Q_M(n) X_M(n) = \begin{bmatrix} 0 & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 \\ \vdots & & T_M(n) & \\ 0 & \dots & \dots & \end{bmatrix} \quad (4.198)$$

Les matrices $S_M(n)$ et $T_M(n)$ sont des matrices triangulaires à $M \times M$ éléments. La matrice $Q_M(n)$ se construit par itérations à partir des données, comme on le verra par la suite.

Une fois obtenues les équations de triangularisation, le calcul de la solution des moindres carrés s'effectue directement. En effet, la fonction coût $J(n)$ est égale au carré de la norme du vecteur d'erreur suivant :

$$\begin{bmatrix} e_n(n) \\ \lambda^{1/2} e_{n-1}(n) \\ \vdots \\ \vdots \\ \lambda^{n/2} e_0(n) \end{bmatrix} = \begin{bmatrix} y(n) \\ \lambda^{1/2} y(n-1) \\ \vdots \\ \vdots \\ \lambda^{n/2} y(0) \end{bmatrix} - X_M(n) H(n) \quad (4.199)$$

Ensuite, les deux membres sont multipliés par la matrice de rotation $Q_M(n)$. Considérons alors l'équation (4.198) de la prédiction arrière. Le vecteur optimal des coefficients minimise la norme du vecteur des erreurs et c'est donc celui qui annule les M derniers termes du vecteur obtenu après rotation :

$$Q_M(n) \begin{bmatrix} e_n(n) \\ \lambda^{1/2} e_{n-1}(n) \\ \vdots \\ \vdots \\ \lambda^{n/2} e_0(n) \end{bmatrix} = \begin{bmatrix} e_{qn}(n) \\ e_{q_{n-1}}(n) \\ \vdots \\ e_{q_M}(n) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.200)$$

Puisque les rotations conservent les normes, la fonction coût s'écrit aussi :

$$J(n) = \sum_{p=M}^n e_{qp}^2(n) \quad (4.201)$$

Finalement, les éléments d'erreur de la fonction coût ont été obtenus par rotation du vecteur de référence, par la matrice $Q_M(n)$.

Ensuite, la matrice triangulaire $T_M(n)$ est utilisée pour obtenir un algorithme rapide. Cependant, il est utile de clarifier d'abord la signification de ses éléments.

4.4.3. Rotations en prédiction linéaire arrière

L'énergie d'erreur de prédiction arrière au temps n , qui s'écrit :

$$E_{bM}(n) = \sum_{p=0}^M \lambda^{n-p} [x(p-M) - X^t(p) B_M(n)]^2 \quad (4.202)$$

est aussi la norme du vecteur d'erreur suivant :

$$\begin{bmatrix} e_{bn}(n) \\ \lambda^{1/2}e_{b(n-1)}(n) \\ \vdots \\ \vdots \\ \lambda^{n/2}e_{bo}(n) \end{bmatrix} = \begin{bmatrix} x(n-M) \\ \lambda^{1/2}x(n-1-M) \\ \vdots \\ \vdots \\ \lambda^{n/2}x(-M) \end{bmatrix} - X_M(n)B_M(n) \quad (4.203)$$

où $B_M(n)$ est le vecteur des coefficients de prédiction arrière. En multipliant par la matrice de rotation $Q_M(n)$, on obtient, comme précédemment, à l'optimum des coefficients du prédiction arrière :

$$Q_M(n) \begin{bmatrix} e_{bn}(n) \\ \lambda^{1/2}e_{b(n-1)}(n) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \lambda^{1/2}e_{bo}(n) \end{bmatrix} = \begin{bmatrix} e_{bqn}(n) \\ e_{bq(n-1)}(n) \\ \vdots \\ \vdots \\ e_{bqM}(n) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.204)$$

Les $(n+1-M)$ valeurs supérieures du vecteur ainsi obtenu résultent de la rotation des erreurs de prédiction arrière et l'énergie est donnée par :

$$E_{bM}(n) = \sum_{p=M}^n e_{bqp}^2(n) \quad (4.205)$$

Les coefficients de prédiction sont calculés en annulant les M derniers termes. Compte tenu de la relation (4.198), ils sont tels que :

$$X_{bqM}(n) = T_M(n)B_M(n) \quad (4.206)$$

où le vecteur à M éléments $X_{bqM}(n)$ représente les M derniers termes du vecteur obtenu par rotation du vecteur des données dans l'équation (4.203).

Les valeurs des éléments de la matrice triangulaire $T_M(n)$ se déterminent par itération sur l'ordre.

Soit $X_{M+1}(n)$ la matrice des données à $M + 1$ colonnes. Elle contient la matrice $X_M(n)$ et la rotation $Q_M(n)$ donne :

$$Q_M(n) \begin{bmatrix} & x(n-M) \\ & \lambda^{1/2}x(n-1-M) \\ X_M(n) & \vdots \\ & \vdots \\ & \lambda^{n/2}x(-M) \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & e_{bqn}(n) \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & e_{bqM}(n) \\ T_M(n) & & X_{bqM}(n) \end{bmatrix} \quad (4.207)$$

Pour obtenir la matrice triangulaire à l'ordre $M + 1$, il faut compléter la matrice de rotation $Q_M(n)$ par une matrice de rotation qui accumule les erreurs de prédiction sur le dernier terme dans la dernière colonne, d'où la récurrence :

$$T_{M+1}(n) = \begin{bmatrix} 0 & \cdots & 0 & E_{bM}^{1/2}(n) \\ \vdots & & \vdots & \vdots \\ & & T_M(n) & X_{bqM}(n) \end{bmatrix} \quad (4.208)$$

et, par suite, en considérant les ordre successifs :

$$T_{M+1}(n) = \begin{bmatrix} 0 & 0 & E_{bM}^{1/2}(n) \\ \vdots & \vdots & \vdots \\ 0 & E_{b1}^{1/2}(n) & \cdots & X_{bqM}(n) \\ E_{b0}^{1/2}(n) & X_{bq1}(n) & \cdots \end{bmatrix} \quad (4.209)$$

Ainsi la matrice $T_{M+1}(n)$ est constituée, sur sa diagonale, des racines des énergies d'erreur de prédiction arrière et, en dehors de la diagonale, des vecteurs de données d'entrée, après rotation, de l'ordre 1 à l'ordre M .

La matrice de rotation $Q_M(n)$ se calcule par récurrence dans le temps. Si la matrice $Q_M(n-1)$, telle que :

$$Q_M(n-1)X_M(n-1) = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \\ T_M(n-1) \end{bmatrix} \quad (4.210)$$

est connue, on peut calculer $Q_M(n)$ en remarquant que la matrice des données satisfait la récurrence :

$$X_M(n) = \begin{bmatrix} x(n) \dots \dots \dots x(n+1-M) \\ \lambda^{1/2} X_M(n-1) \end{bmatrix} \quad (4.211)$$

En effet, on peut écrire :

$$\begin{bmatrix} 1 & 0 \\ 0 & Q_M(n-1) \end{bmatrix} X_M(n) = \begin{bmatrix} x(n) & \dots & x(n+1-M) \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \\ & \lambda^{1/2} T_M(n-1) & \end{bmatrix} \quad (4.212)$$

et annule la première ligne par un ensemble de rotations qui conduisent à la matrice $T_M(n)$. La première de ces rotations s'écrit :

$$R_1 = \begin{bmatrix} \cos \theta_1 & 0 \dots \dots \dots 0 & -\sin \theta_1 \\ 0 & 1 \dots \dots \dots 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 \dots \dots \dots 1 & 0 \\ \sin \theta_1 & 0 \dots \dots \dots 0 & \cos \theta_1 \end{bmatrix} \quad (4.213)$$

Elle annule $x(n)$ et l'angle θ_1 est tel que :

$$\cos \theta_1 = \frac{\lambda^{1/2} E_{b0}^{1/2}(n-1)}{E_{b0}^{1/2}(n)} \quad ; \quad \sin \theta_1 = \frac{x(n)}{E_{b0}^{1/2}(n)} \quad (4.214)$$

A noter que $E_{b0}(n)$ est l'énergie du signal d'entrée. L'élément suivant, qui était $x(n-1)$ a été modifié par la première rotation. En le désignant par u , la seconde rotation, définie par :

$$R_2 = \begin{bmatrix} \cos \theta_2 & 0 \dots \dots \dots -\sin \theta_2 & 0 \\ 0 & 1 \dots \dots \dots 0 & 0 \\ \vdots & & \vdots & \vdots \\ \sin \theta_2 & 0 \dots \dots \dots \cos \theta_2 & 0 \\ 0 & 0 \dots \dots \dots 0 & 1 \end{bmatrix} \quad (4.215)$$

correspond à l'angle θ_2 tel que :

$$u \sin \theta_2 + \lambda^{1/2} E_{b1}^{1/2}(n-1) \cos \theta_2 = E_{b1}^{1/2}(n)$$

c'est à dire :

$$\cos \theta_2 = \frac{\lambda^{1/2} E_{b1}^{1/2}(n-1)}{E_{b1}^{1/2}(n)} \quad ; \quad \sin \theta_2 = \frac{u}{E_{b1}^{1/2}(n)} \quad (4.216)$$

Pour clarifier la signification de cette variable u , il faut considérer la récurrence dans le temps de l'énergie de prédiction :

$$E_{b1}(n) = \lambda E_{b1}(n-1) + e_{b1q}^2(n) \tag{4.217}$$

qui montre que la variable u est, en fait, l'erreur de prédiction linéaire arrière après rotation $e_{b1q}(n)$. En itérant la procédure, il vient :

$$Q_M(n) = R_M \dots R_2 R_1 \begin{bmatrix} 1 & 0 \\ 0 & Q_M(n-1) \end{bmatrix} \tag{4.218}$$

les angles étant tels que :

$$\cos \theta_{i+1} = \frac{\lambda^{1/2} E_{bi}^{1/2}(n-1)}{E_{bi}^{1/2}(n)} ; \sin \theta_{i+1} = \frac{e_{biq}(n)}{E_{bi}^{1/2}(n)} \tag{4.219}$$

Il faut souligner que cette factorisation n'est possible que parce que la matrice $T_M(n-1)$ est triangulaire.

4.4.4. Rotation en prédiction linéaire avant

Comme pour la structure transversale, les algorithmes rapides combinent les prédictions avant et arrière et la récurrence entre $T_{M+1}(n)$ et $T_M(n)$ s'obtient aussi pour la prédiction avant :

L'énergie d'erreur de prédiction avant $E_{aM}(n)$, donnée par :

$$E_{aM}(n) = \sum_{p=0}^n \lambda^{n-p} [x(p) - X^t(p-1)A_M(n)]^2 \tag{4.220}$$

est le carré de la norme du vecteur d'erreurs :

$$\begin{bmatrix} e_{an}(n) \\ \lambda^{1/2} e_{a(n-1)}(n) \\ \vdots \\ \lambda^{n/2} e_{ao}(n) \end{bmatrix} = \begin{bmatrix} x(n) \\ \lambda^{1/2} x(n-1) \\ \vdots \\ \lambda^{n/2} x(0) \end{bmatrix} - \begin{bmatrix} X_M(n-1) \\ 0 \dots \dots \dots 0 \end{bmatrix} A_M(n) \tag{4.221}$$

La matrice des données qui intervient est la matrice $X_M(n-1)$ complétée par une ligne de zéros, pour rendre compte du décalage temporel de la prédiction avant. Pour faire apparaître les erreurs de prédiction avant après rotation, il faut considérer le produit :

$$\begin{bmatrix} Q_M(n-1) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(n) \\ \lambda^{1/2} x(n-1) \\ \vdots \\ \lambda^{n/2} x(0) \end{bmatrix} = \begin{bmatrix} e_{aqn}(n) \\ \vdots \\ e_{aq(M+1)}(n) \\ X_{aqM}(n) \\ \lambda^{n/2} x(0) \end{bmatrix} \tag{4.222}$$

Comme précédemment, le vecteur des coefficients de prédiction avant $A_M(n)$ est la solution du système :

$$X_{aqM}(n) = T_M(n-1) A_M(n) \tag{4.223}$$

où le vecteur $X_{aqM}(n)$ représente le résultat de la rotation des données d'entrée.

Comme pour les algorithmes en structure transversale, une récurrence entre les matrices $T_{M+1}(n)$ et $T_M(n-1)$ est obtenue d'abord, à partir de la matrice des données augmentée $X_{M+1}(n)$. Soit le produit :

$$\begin{bmatrix} Q_M(n-1) & 0 \\ 0 & 1 \end{bmatrix} X_{M+1}(n) = \begin{bmatrix} e_{aqn}(n) & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ e_{aq(M+1)}(n) & 0 & \dots & 0 \\ X_{aqM}(n) & T_M(n-1) & & \\ w^{n/2} x(0) & 0 & \dots & 0 \end{bmatrix} \tag{4.224}$$

La triangularisation est obtenue en deux étapes. D'abord, il faut accumuler les erreurs de prédiction après rotation sur le dernier élément $w^{n/2} x(0)$, qui se trouve remplacé par la racine carrée de l'énergie d'erreur de prédiction avant.

Ensuite, il faut accumuler le vecteur $X_{aqM}(n)$ sur le dernier élément, ce qui est obtenu par la matrice $Q_\alpha(n)$ telle que :

$$Q_\alpha(n) \begin{bmatrix} X_{aqM}^{(n)} & 0 & \dots & \\ \vdots & T_M(n-1) & & \\ E_{aM}^{1/2}(n) & 0 & \dots & 0 \end{bmatrix} = T_{M+1}(n) \tag{4.225}$$

Cette matrice est le produit de M rotations R_{α_i} avec :

$$R_{\alpha_i} = \begin{bmatrix} 1 & \dots & \cos \alpha_i & \dots & -\sin \alpha_i \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & \sin \alpha_i & \dots & \cos \alpha_i \end{bmatrix} \tag{4.226}$$

Les angles $\alpha_i (1 \leq i \leq M)$ se déterminent en utilisant la définition des termes diagonaux des matrices $T_{M+1}(n)$ et $T_M(n-1)$. Comme la matrice $T_M(n-1)$ a été complétée par une ligne inférieure de zéros, il vient :

$$E_{b(i+1)}^{1/2}(n) = \cos \alpha_{i+1} E_{b_i}^{1/2}(n-1) \tag{4.227}$$

Si l'on rapproche ce résultat de la récurrence sur l'ordre correspondante pour l'algorithme du treillis normalisé :

$$E_{b(i+1)}^{1/2}(n) = [1 - k_{i+1}^2(n)] E_{b_i}^{1/2}(n-1) \tag{4.228}$$

il apparaît que les angles sont liés aux coefficients du treillis par :

$$\sin \alpha_i = k_i(n) \quad ; \quad 1 \leq i \leq M \quad (4.229)$$

La procédure de calcul des angles se définit par la relation simplifiée.

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ E_{b0}^{1/2}(n) \end{bmatrix} = Q_\alpha(n) \begin{bmatrix} X_{aqM}(n) \\ E_{aM}^{1/2}(n) \end{bmatrix} \quad (4.230)$$

A noter la relation obtenue en inversant cette équation

$$\prod_{i=1}^M \cos \alpha_i = E_{aM}^{1/2}(n) / E_{b0}^{1/2}(n) \quad (4.231)$$

De même, les valeurs des angles sont données par :

$$\sin \alpha_i = \frac{x_{aqMi}(n)}{E_{aM}(n) + \sum_{j=1}^M x_{aqMj}^2(n)} = k_i(n) \quad (4.232)$$

Les récurrences ayant été explicitées, l'algorithme rapide peut être défini.

4.4.5. L'algorithme QR rapide

L'algorithme doit effectuer le calcul des angles en fonction de la nouvelle donnée au temps n , $x(n)$. Deux types d'angles sont à considérer, ceux qui sont liés à la matrice $Q_M(n)$ et ceux de la matrice Q_α . Le calcul des angles de la matrice Q_α est défini par la relation (4.230), quand les éléments de la prédiction avant sont disponibles. Il faut donc d'abord obtenir ces éléments.

Compte tenu de la relation de récurrence (4.218) pour la matrice $Q_M(n)$, les angles θ_i peuvent être représenté au temps n par les variables $\gamma_M(n)$ et $G_M(n)$ définies par :

$$Q_M(n) \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \gamma_M(n) \\ 0 \\ \vdots \\ 0 \\ G_M(n) \end{bmatrix} \quad (4.233)$$

En effet, il vient pour $\gamma_M(n)$:

$$\gamma_M(n) = \prod_{i=1}^M \cos \theta_i \quad (4.234)$$

L'interprétation physique de ce paramètre est obtenue à partir de la relation (4.203) qui, combinée à la relation (4.233), donne :

$$e_n(n) = [e_{qn}(n) \dots e_{qM}(n) 0 \dots 0] Q_M(n) \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \gamma_M(n) e_{qn}(n) \quad (4.235)$$

Ainsi le scalaire $\gamma_M(n)$ est le rapport de l'erreur *a posteriori* à l'erreur après rotation. C'est la racine carrée du rapport $\varphi_M(n)$ des erreurs *a priori* et *a posteriori*.

Quant aux M éléments du vecteur $G_M(n)$ ils sont obtenus en appliquant les définitions des rotations $R_1 \dots R_{M_1}$, ce qui conduit à :

$$\begin{aligned} g1(n) &= \sin \theta_1 = x(n) / E_{b0}^{1/2}(n) \\ g2(n) &= \sin \theta_2 \cos \theta_1 = \varepsilon_{bi}(n) / E_{b1}^{1/2}(n) \\ &\dots\dots\dots \\ gM(n) &= \varepsilon_{b(M-1)} / E_{b(M-1)}^{1/2}(n) \end{aligned} \quad (4.236)$$

où la variable $\varepsilon_{bi}(n)$ est l'erreur de prédiction arrière *a posteriori* d'ordre i , liée à l'erreur après rotation par :

$$\varepsilon_{bi}(n) = \gamma_i(n) e_{biq}(n) \quad (4.237)$$

Ce vecteur $G_M(n)$ fait la liaison entre la prédiction arrière et la prédiction avant, comme le gain d'adaptation pour la structure transversale. Il obéit à une équation de récurrence qui est obtenue en reprenant le calcul de la prédiction avant. En effet, le calcul qui aboutit à la matrice triangulaire $T_{M+1}(n)$ revient à représenter la matrice de rotation $Q_{M+1}(n)$ par trois facteurs :

$$Q_{M+1}(n) = \begin{bmatrix} I_{M-M} & 0 \\ 0 & Q_\alpha(n) \end{bmatrix} Q_{ea}(n) \begin{bmatrix} Q_M(n-1) & 0 \\ 0 & 1 \end{bmatrix} \quad (4.238)$$

La matrice $Q_{ea}(n)$ représente les rotations qui accumulent les erreurs de prédiction arrière pour fournir l'énergie $E_{aM}(n)$. En multipliant les deux membres par le vecteur nul sauf le premier élément égal à l'unité, il vient :

$$\begin{bmatrix} \gamma_{M+1}(n) \\ 0 \\ \vdots \\ 0 \\ G_{M+1}(n) \end{bmatrix} = \begin{bmatrix} I_{n-M} & 0 \\ 0 & Q_\alpha(n) \end{bmatrix} Q_{ea}(n) \begin{bmatrix} \gamma_M(n-1) \\ 0 \\ \vdots \\ 0 \\ G_M(n-1) \\ 0 \end{bmatrix} \quad (4.239)$$

La matrice $Q_{ea}(n)$ intervient simplement pour projeter le premier élément $\gamma_M(n-1)$ à la dernière position et fournir $\gamma_{M+1}(n)$. En se limitant aux $M+1$ derniers éléments, il vient :

$$Q_\alpha(n) \begin{bmatrix} G_M(n-1) \\ \gamma_M(n-1) e_{aqn}(n) / E_{aM}^{1/2}(n) \end{bmatrix} = \begin{bmatrix} g_{M+1}(n) \\ G_M(n) \end{bmatrix} \quad (4.240)$$

Un algorithme est alors obtenu en combinant les relations (4.240), (4.233), (4.230) et (4.218). Dans les vecteurs à n éléments, seuls le premier et les M derniers sont concernés par ces opérations et il est possible de simplifier en limitant la représentation à des vecteurs à $M+1$ éléments et des matrices $(M+1) \times (M+1)$. Ainsi dans l'algorithme du Tableau 4.6, le produit $R_1 \dots R_M$ est représenté par la matrice $Q_a(n)$.

Les calculs numériques à effectuer sont de 2 types : les rotations et les calculs d'angle. Une rotation est comparable à une multiplication complexe, un calcul d'angle peut se faire avec une table.

L'algorithme présenté dans le [Tableau 4.6] p.196 est numériquement stable [Regalia 93]. En fait, on peut montrer qu'il incorpore un mécanisme de contrôle des erreurs d'arrondi. La précision des calculs peut être visualisée, par exemple en utilisant la variable $g_1(n+1)$ qui est obtenue avec la dernière des rotations de la matrice Q_α et peut être aussi calculée directement à partir des données d'entrée par la relation (4.236).

L'approche par décomposition QR peut être étendue aux signaux complexes [Terré 99]. Elle se généralise également aux signaux multidimensionnels [Bellanger 91]

4.4.6. Equivalence avec le treillis

L'algorithme présenté ci-dessus peut être rendu plus régulier par une normalisation du signal d'entrée, correspondant à la rotation .

$$\begin{bmatrix} 0 \\ E_{b0}^{1/2}(n+1) \end{bmatrix} = \begin{bmatrix} \cos \Psi(n+1) & -\sin \Psi(n+1) \\ \sin \Psi(n+1) & \cos \Psi(n+1) \end{bmatrix} \begin{bmatrix} x(n+1) \\ \lambda^{1/2} E_{b0}^{1/2}(n) \end{bmatrix} \quad (4.241)$$

avec :

$$\sin \Psi(n+1) = \frac{x(n+1)}{E_{b0}^{1/2}(n+1)} ; \quad \cos \Psi(n+1) = \frac{\lambda^{1/2} E_{b0}^{1/2}(n)}{E_{b0}^{1/2}(n+1)}$$

Dans l'algorithme, $x(n+1)$ est alors remplacé par $\sin \Psi(n+1)$ et $\lambda^{1/2}$ est remplacé par $\cos \Psi(n+1)$. Alors, le premier calcul d'angle devient :

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = Q_\alpha(n+1) \begin{bmatrix} X_q(n+1) \\ E_a^{1/2}(n+1) \end{bmatrix} \quad (4.242)$$

ALGORITHME MCR-QRDisponible au temps n :Vecteur de données d'entrée après rotation : $X_q(n)$
Vecteur de référence à M éléments après rotation : $Y_q(n)$ Vecteur unité après rotation : $G_M(n)$ Energie d'erreur de prédiction avant : $E_a(n)$ Rapport des erreurs de prédiction : $\gamma_M(n)$ Facteur de pondération : λ Nouvelles données au temps n Signal d'entrée : $x(n+1)$; Référence : $y(n+1)$ Prédiction :

$$\begin{vmatrix} e_{aq}(n+1) \\ X_q(n+1) \end{vmatrix} = Q_a(n) \begin{vmatrix} x(n+1) \\ \lambda^{1/2} X_q(n) \end{vmatrix}$$

$$E_a(n+1) = \lambda E_a(n) + e_{aq}^2(n+1)$$

$$\begin{vmatrix} 0 \\ E_{bo}^{1/2}(n+1) \end{vmatrix} = Q_\alpha(n+1) \begin{vmatrix} X_q(n+1) \\ E_a^{1/2}(n+1) \end{vmatrix}$$

$$u = \gamma_M(n) e_{aq}(n+1) / E_a^{1/2}(n+1)$$

$$\begin{vmatrix} g_{M+1}(n+1) \\ G_M(n+1) \end{vmatrix} = Q_\alpha(n+1) \begin{vmatrix} G_M(n) \\ u \end{vmatrix}$$

$$\begin{vmatrix} \gamma_M(n+1) \\ G_M(n+1) \end{vmatrix} = Q_a(n+1) \begin{vmatrix} 1 \\ 0 \end{vmatrix}$$

Filtre

$$\begin{vmatrix} e_q(n+1) \\ Y_q(n+1) \end{vmatrix} = Q_a(n+1) \begin{vmatrix} y(n+1) \\ \lambda^{1/2} Y_q(n) \end{vmatrix}$$

$$\varepsilon(n+1) = \gamma_M(n+1) e_q(n+1)$$

Tableau 4.6. Algorithme de moindres carrés avec rotation

équation semblable au calcul de $Q_a(n+1)$. On peut même pousser encore la similitude en inversant la relation (4.242) et en permutant les variables pour obtenir :

$$\begin{bmatrix} E_a^{1/2}(n+1) \\ X_q(n+1) \end{bmatrix} = J_M Q_\alpha^{-1}(n+1) J_M \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (4.243)$$

Un algorithme basé sur les récurrences sur l'ordre peut être obtenu, qui est en fait un algorithme du treillis normalisé. Les deux approches, treillis et QR , utilisent les mêmes variables normalisées.

4.5. Conclusion

A partir de la même fonction coût quadratique $J(n)$ pour un filtre d'ordre M , trois familles d'algorithmes rapides ont été développées :

1) les algorithmes de moindres carrés récursifs en structure transverse directe, qui sont obtenus en éliminant la matrice d'autocorrélation (AC) par combinaison des prédictions avant et arrière. Ils fournissent les coefficients du filtre adaptatif et les coefficients de prédiction et leur complexité arithmétique est de l'ordre de $10M$. Ils sont numériquement instables, cependant, les erreurs d'arrondi peuvent être contrôlées.

2) les algorithmes en treillis qui correspondent à une factorisation en 2 matrices triangulaires de la matrice AC et qui conduisent à une structure en treillis-échelle. Ils sont basés sur une combinaison de récurrences sur l'ordre et de récurrences temporelles. Ils fournissent les coefficients de réflexion pour tous les filtres d'ordre intermédiaire, avec une complexité numérique de l'ordre de $16M$. Bien que leur stabilité numérique ne soit pas prouvée, l'expérience montre qu'ils sont peu sensibles aux erreurs d'arrondi.

3) Les algorithmes à rotation, basés sur une décomposition QR de la matrice des données. Ils peuvent éviter toute élévation au carré des variables et, par suite, opérer avec la dynamique des données. Les opérations à mettre en œuvre sont des rotations, comparables à des multiplications complexes et des calculs d'angle. Les variables internes sont liées à celles de l'algorithme en treillis normalisé. Ces algorithmes sont numériquement stables.

Finalement, ces trois familles d'algorithmes donnent au concepteur de filtres adaptatifs les moyens de mettre en œuvre les techniques de moindres carrés avec différents niveaux de complexité et de robustesse, lui permettant de s'adapter au mieux à ses objectifs et à ses contraintes.

4.6. Bibliographie

- Alcantara, Prado and Gueguen, 1986, "Fixed Point Implementation of the Fast Kalman Algorithm Using a TMS 32010 Microprocessor", Proceedings of EUSIPCO-86, Morth Holland, The Hague, Sept. 1986, pp.1335-8.
- Bellanger, Regalia, 1991, "The FLS-QR Algorithm for Adaptive Filtering : the Case of Multichannel Signals", Signal Processing, vol.22, pp.115-126.
- Botto and Moustakides, 1989, "Stabilizing the Fast Kalman Algorithms", IEEE Trans., vol.ASSP-37, 1989, pp. 1342-8.
- Carayannis, Manolakis et Kalouptsidis, 1983, "A Fast Sequential Algorithm for LS Filtering and Prediction", IEEE Trans., vol ASSP-31, n° 6, December 1983, pp. 1394-1402.
- Cioffi et Kailath, 1984, "Fast recursive Least Squares Transversal Filters for Adaptive Filtering", IEEE Trans., vol. ASSP-32, n°2, April 1984, pp.304-337.

- Cioffi, 1987, "*Limited Precision Effects in Adaptive Filtering*", IEEE Trans., vol. CAS-34, July 1987, pp.821-833.
- Cioffi, 1990, "*The Fast Adaptive Rotors RLS Algorithm*", IEEE Trans., vol. ASSP-38, pp.631-653.
- Falconer et L. Jung 1978, "*Application of Fast Kalman Estimation to Adaptive Equalization*", IEEE Trans., vol. COM-26, n° 10, October 1978, pp. 1439-1446.
- Giordano et Hsu, 1989, *Least Squares Estimation with Application to Digital Signal Processing*, John Wiley, Mew-York, 1989.
- Kalouptsidis and Theodoridis, 1993, *Adaptive System Identification and Signal Processing Algorithms*, Prentice-Hall, London, 1993.
- Lee, Friedlander et Morf, 1981, "*Recursive Square Root Ladder Estimation Algorithms*", IEEE Trans., vol. ASSP-29, pp.627-641.
- Lin, 1984, "*On Digital Implementation of the Fast Kalman Algorithms*", IEEE Trans., vol. ASSP-32, n°5, October 1984, pp.998-1005.
- Ling, Manolakis et Proakis, 1986, "*Numerically Robust L.S.Lattice-Ladder Algorithms with Direct Updating of the Coefficients*", IEEE Trans., vol. ASSP-34, pp.837-845.
- Ljung et Ljung, 1985, "*Error Propagation Properties of Recursive Least Squares Adaptation algorithms*", Automatica, vol.21, n°2, 1985, pp.157-167.
- Proudler, McWhirter, Sheperd, 1990, "*Computationally Efficient QR Decomposition Approach to Least Squares Adaptive Filtering*", IEE Proceedings, vol.138, M°4, pp.341-353.
- Regalia, 1993a, "*Numerical stability issues in fast least-squares adaptation algorithms*", Optical Engineering, vol. 31, pp.1144-1152, juin 1992.
- Regalia, 1993b, "*Numerical Stability Properties of QR-based Fast Least Squares Algorithms*", IEEE Trans.vol.SP-41, pp.2096-2109.
- Samson et Reddy, 1983, "*Fixed Point Error Analysis of the Normalised Ladder Algorithm*", IEEE Trans., vol. ASSP-31, pp.1177-1191.
- Sayed, 2003, *Fundamentals of Adaptive Filtering*, John Wiley, Hoboken, M.J., 2003.
- Slock and Kailath, 1991, "*Numerically Stable Fast Transversal Filters for Recursive Least Squares Adaptive Filtering*", IEEE Trans., vol. ASSP-39, 1991, pp. 92-114.
- Slock and Kailath, 1992, "*A Modular Multichannel Multiexperiment Fast Transversal Filter RLS Algorithm*", Signal Processing, vol.28, 1992, pp.25-45.
- Terré, Bellanger, 1999, "*A Fast Least Squares QRD-based Algorithm for Complex Data*" IEEE Trans., vol.SP-42, pp.3272-3273.
- Tylavsky et Sohie, 1986, "*Generalization of the Matrix Inversion Lemma*", Proceedings of the IEEE, vol. 74, n° 7, July 1986, pp.1050-1052.

Chapitre 5

Filtrage RII Adaptatif

5.1. Introduction

Le développement des filtres adaptatifs à réponse impulsionnelle infinie (RII) remonte à quelques décennies déjà, motivé par des applications où l'emploi d'un filtre à réponse impulsionnelle finie (RIF) nécessiterait un très grand nombre de coefficients. Les filtres RII, ayant comme fonctions de transfert des fractions rationnelles, semblent offrir des capacités de modélisation plus souples, surtout lorsque l'on cherche à modéliser des réponses impulsionnelles longues et/ou des spectres avec des pics prononcés. Leurs propriétés de convergence, en revanche, sont moins bien maîtrisées que celles applicables aux filtres RIF adaptatifs qui présentent des critères quadratiques à optimiser. Les critères applicables aux filtres RII sont non quadratiques en général, et peuvent présenter des minima locaux ou d'autres irrégularités. Par conséquent, le temps de convergence peut être, dans certains cas, assez lent.

Nous présentons dans ce chapitre les algorithmes principaux en filtrage RII adaptatif (dont certains ont été présentés rapidement dans le chapitre 3, comme illustration des méthodes générales). Après un rappel des principes d'identification du système, nous examinons les méthodes en erreur d'équation et leurs capacités de modélisation. Nous passons ensuite aux algorithmes pilotés par une erreur de sortie, comprenant la méthode de gradient, la méthode de Steiglitz-McBride, et les algorithmes à base d'hyperstabilité. Le chapitre se termine en présentant des liens avec le problème d'approximation rationnelle, qui intervient lorsqu'on cherche à qualifier l'adéquation entre le filtre optimal et l'environnement signal.

Chapitre rédigé par Phillip REGALIA .

5.2. Principes d'Identification du Système

Le cadre traditionnel pour étudier les filtres RII adaptatifs est celui d'identification du système, où l'on suppose que les signaux d'entrée et de référence sont liés par un système dynamique inconnu ; le filtre adaptatif tente d'identifier le système.

Pour commencer, supposons que les entrées $\{u(n)\}$ et les sorties $\{y(n)\}$ du système à identifier vérifient une équation aux différences

$$\begin{aligned} y(n) + a_1 y(n-1) + \dots + a_M y(n-M) \\ = b_0 u(n) + b_1 u(n-1) + \dots + b_M u(n-M) \end{aligned} \quad (5.1)$$

pour un certain jeu de paramètres $\{a_k\}_{k=1}^M$ et $\{b_k\}_{k=0}^M$. Par la transformée en z , cette équation aux différences donne un fraction rationnelle comme fonction de transfert :

$$\underbrace{\sum_k y(k) z^{-k}}_{Y(z)} = \underbrace{\frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + \dots + a_M z^{-M}}}_{H(z) = \frac{B(z)}{A(z)}} \underbrace{\sum_k u(k) z^{-k}}_{U(z)} \quad (5.2)$$

Notons que les deux polynômes $A(z)$ et $B(z)$ sont écrits avec le même degré M ; ceci est sans pertes de généralités, car le cas où $d^\circ A(z) \neq d^\circ B(z)$ s'obtient en permettant que certains coefficients s'annulent. Si, par exemple,

$$\begin{aligned} A(z) = 1 + 1.2z^{-1} + 0.8z^{-2} &\Rightarrow d^\circ A(z) = 2 \\ B(z) = 0.4 + 0.6z^{-1} &\Rightarrow d^\circ B(z) = 1 \end{aligned}$$

on peut réécrire $B(z)$ sous la forme

$$B(z) = 0.4 + 0.6z^{-1} + 0z^{-2}$$

pour retrouver un polynôme de degré 2, dont par chance un des coefficients s'annule : $b_2 = 0$.

5.3. Méthode d'erreur d'équation

Considérons maintenant le problème de déterminer les coefficients $\{a_k\}$ et $\{b_k\}$ à partir des observations $\{u(n)\}$ et $\{y(n)\}$, sous l'hypothèse que les coefficients vérifient l'équation (5.1). Pour ceci, introduisons des estimées des paramètres, notées

$\{\hat{a}_k\}$ et $\{\hat{b}_k\}$, qui seront choisis de façon à minimiser un certain signal d'erreur, appelé traditionnellement "erreur d'équation", sous la forme

$$e(n) = y(n) + \underbrace{\sum_{k=1}^M \hat{a}_k y(n-k) - \sum_{k=0}^M \hat{b}_k u(n-k)}_{\text{estimée de } -y(n)}$$

Nous constatons que les choix $\hat{a}_k = a_k$ et $\hat{b}_k = b_k$ conduisent à $e(n) \equiv 0$ en vue de (5.1).

Or, le signal d'erreur peut être écrit en format vectoriel de la manière suivante :

$$\begin{aligned} e(n) &= \underbrace{\begin{bmatrix} y(n) & y(n-1) & \cdots & y(n-M) \end{bmatrix}}_{\mathbf{y}^t(n)} \underbrace{\begin{bmatrix} 1 \\ \hat{a}_1 \\ \vdots \\ \hat{a}_M \end{bmatrix}}_{\hat{\mathbf{a}}} \\ &\quad - \underbrace{\begin{bmatrix} u(n) & \cdots & u(n-M) \end{bmatrix}}_{\mathbf{u}^t(n)} \underbrace{\begin{bmatrix} \hat{b}_0 \\ \vdots \\ \hat{b}_M \end{bmatrix}}_{\hat{\mathbf{b}}} \\ &= \begin{bmatrix} \mathbf{y}^t(n) & \mathbf{u}^t(n) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}} \\ -\hat{\mathbf{b}} \end{bmatrix} \end{aligned}$$

Ainsi, pour un jeu de paramètres fixes, on peut choisir comme fonction de coût $J(\hat{\mathbf{a}}, \hat{\mathbf{b}})$ la variance $E[e^2(n)]$ de l'erreur, qui s'écrit

$$\begin{aligned} J(\hat{\mathbf{a}}, \hat{\mathbf{b}}) = E[e^2(n)] &= \begin{bmatrix} \hat{\mathbf{a}}^t & (-\hat{\mathbf{b}})^t \end{bmatrix} E \left\{ \begin{bmatrix} \mathbf{y}(n) \\ \mathbf{u}(n) \end{bmatrix} \begin{bmatrix} \mathbf{y}^t(n) & \mathbf{u}^t(n) \end{bmatrix} \right\} \begin{bmatrix} \hat{\mathbf{a}} \\ -\hat{\mathbf{b}} \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{a}}^t & (-\hat{\mathbf{b}})^t \end{bmatrix} \begin{bmatrix} \mathbf{R}_{yy} & \mathbf{R}_{yu} \\ \mathbf{R}_{uy} & \mathbf{R}_{uu} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}} \\ -\hat{\mathbf{b}} \end{bmatrix} \end{aligned} \quad (5.3)$$

avec les correspondances

$$\begin{aligned} \mathbf{R}_{yy} &= E[\mathbf{y}(n) \mathbf{y}^t(n)] \\ \mathbf{R}_{yu} &= E[\mathbf{y}(n) \mathbf{u}^t(n)] = \mathbf{R}_{uy}^t \\ \mathbf{R}_{uu} &= E[\mathbf{u}(n) \mathbf{u}^t(n)] \end{aligned}$$

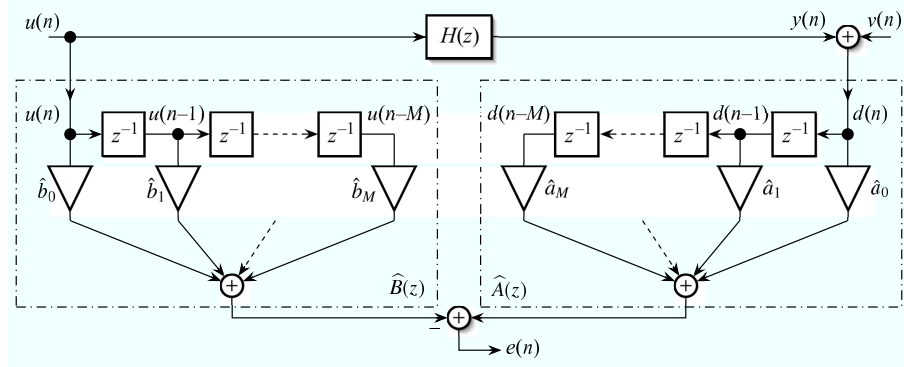


Figure 5.1. Schéma pour la génération de l'erreur d'équation $e(n)$.

Pour caractériser le minimum, notons que $E[e^2(n)]$ est une fonction quadratique des coefficients du vecteur $\hat{\mathbf{b}}$. Le minimum par rapport à $\hat{\mathbf{b}}$ (et pour $\hat{\mathbf{a}}$ fixe) s'obtient en annulant les dérivées, ce qui mène à

$$\frac{\partial J(\hat{\mathbf{a}}, \hat{\mathbf{b}})}{\partial \hat{\mathbf{b}}} = 0 \quad \Leftrightarrow \quad 2(\mathbf{R}_{uy}\hat{\mathbf{a}} - \mathbf{R}_{uu}\hat{\mathbf{b}}) = 0$$

ou encore $\hat{\mathbf{b}} = \mathbf{R}_{uu}^{-1}\mathbf{R}_{uy}\hat{\mathbf{a}}$, sous l'hypothèse raisonnable que l'inverse \mathbf{R}_{uu}^{-1} existe.¹ La substitution de cette expression pour le $\hat{\mathbf{b}}$ optimum (en fonction de $\hat{\mathbf{a}}$) permet d'écrire $J(\hat{\mathbf{a}}) = J(\hat{\mathbf{a}}, \hat{\mathbf{b}})|_{\hat{\mathbf{b}}=\hat{\mathbf{b}}_{\text{opt}}}$ sous une forme qui ne dépend que de $\hat{\mathbf{a}}$:

$$J(\hat{\mathbf{a}}) = \hat{\mathbf{a}}^t \underbrace{[\mathbf{R}_{yy} - \mathbf{R}_{yu}\mathbf{R}_{uu}^{-1}\mathbf{R}_{uy}]}_{\triangleq \mathbf{R}_{y/u}} \hat{\mathbf{a}}, \quad \text{si } \hat{\mathbf{b}} = \mathbf{R}_{uu}^{-1}\mathbf{R}_{uy}\hat{\mathbf{a}}.$$

Nous introduisons la notation $\mathbf{R}_{y/u}$ pour la matrice extraite comme indiquée de la matrice de covariance $\begin{bmatrix} \mathbf{R}_{yy} & \mathbf{R}_{yu} \\ \mathbf{R}_{uy} & \mathbf{R}_{uu} \end{bmatrix}$ qui figure dans l'équation (5.3); la forme $\mathbf{R}_{y/u}$ s'appelle le *complément de Schur* (par rapport à \mathbf{R}_{uu}) de cette dernière.

La formulation jusqu'ici suppose que la sortie du système à déterminer est observée sans bruit. Si l'on y ajoute du bruit, comme suggéré dans la figure 5.1, la sortie observable, notée $\{d(n)\}$, devient

$$d(n) = y(n) + v(n), \quad v(n) = \text{bruit de sortie.}$$

1. Cette hypothèse est vérifiée dès lors que le signal $\{u(n)\}$ vérifie la propriété d'*excitation persistante* de degré $M+1$, c'est à dire que sa description ne se résume pas en une somme de $M+1$ cisoïdes ou moins.

La variance de l'erreur d'équation se développe d'une manière analogue à celle ci-dessus, en remplaçant $y(n)$ par $d(n)$. Avec l'hypothèse que $v(n)$ et $u(m)$ sont décorrélés pour tout n et m , l'expression (5.3) pour la variance d'erreur d'équation devient

$$\begin{aligned} J(\hat{\mathbf{a}}, \hat{\mathbf{b}}) &= E[e^2(n)] = [\hat{\mathbf{a}}^t \quad (-\hat{\mathbf{b}})^t] E \left\{ \begin{bmatrix} \mathbf{d}(n) \\ \mathbf{u}(n) \end{bmatrix} \begin{bmatrix} \mathbf{d}^t(n) & \mathbf{u}^t(n) \end{bmatrix} \right\} \begin{bmatrix} \hat{\mathbf{a}} \\ -\hat{\mathbf{b}} \end{bmatrix} \\ &= [\hat{\mathbf{a}}^t \quad (-\hat{\mathbf{b}})^t] \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{du} \\ \mathbf{R}_{ud} & \mathbf{R}_{uu} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}} \\ -\hat{\mathbf{b}} \end{bmatrix} \end{aligned}$$

avec les correspondances

$$\begin{aligned} \mathbf{R}_{dd} &= E[\mathbf{d}(n) \mathbf{d}^t(n)] = \mathbf{R}_{yy} + \mathbf{R}_{vv} \\ \mathbf{R}_{du} &= E[\mathbf{d}(n) \mathbf{u}^t(n)] = \mathbf{R}_{yu} \\ \mathbf{R}_{ud} &= E[\mathbf{u}(n) \mathbf{d}^t(n)] = \mathbf{R}_{du}^t \end{aligned}$$

L'optimisation de $\hat{\mathbf{b}}$ par rapport à $\hat{\mathbf{a}}$ mène à $\hat{\mathbf{b}} = \mathbf{R}_{uu}^{-1} \mathbf{R}_{ud} \hat{\mathbf{a}} = \mathbf{R}_{uu}^{-1} \mathbf{R}_{uy} \hat{\mathbf{a}}$, ce qui donne la même formule que précédemment.

Concernant le complément de Schur, notons que

$$\begin{aligned} \mathbf{R}_{d/u} &\triangleq \mathbf{R}_{dd} - \mathbf{R}_{du} \mathbf{R}_{uu}^{-1} \mathbf{R}_{ud} \\ &= \mathbf{R}_{yy} + \mathbf{R}_{vv} - \mathbf{R}_{yu} \mathbf{R}_{uu}^{-1} \mathbf{R}_{uy} \\ &= \mathbf{R}_{y/u} + \mathbf{R}_{vv} \end{aligned}$$

Ainsi, l'expression pour la variance réduite se simplifie en

$$\begin{aligned} J(\hat{\mathbf{a}}) &= \hat{\mathbf{a}}^t \mathbf{R}_{d/u} \hat{\mathbf{a}} \\ &= \hat{\mathbf{a}}^t \mathbf{R}_{y/u} \hat{\mathbf{a}} + \hat{\mathbf{a}}^t \mathbf{R}_{vv} \hat{\mathbf{a}}. \end{aligned} \quad (5.4)$$

Le premier terme du membre à droite est la contribution induite par l'entrée $\{u(n)\}$, tandis que le deuxième est la contribution induite par le bruit $\{v(n)\}$.

La minimisation par rapport à $\hat{\mathbf{a}}$, sous la contrainte que son premier élément soit 1, se caractérise par l'équation suivante :

$$\mathbf{R}_{d/u}\hat{\mathbf{a}} = \begin{bmatrix} \sigma_e^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5.5)$$

ce qui montre que $\hat{\mathbf{a}}$ est orthogonale à toutes les lignes de $\mathbf{R}_{d/u}$ sauf la première ; la valeur σ_e^2 restante est reconnue comme la variance d'erreur d'équation ainsi minimisée.

La solution optimale varie, en général, avec le rapport signal-à-bruit en sortie, car selon (5.4) le $\hat{\mathbf{a}}$ optimal est contraint de trouver un compromis entre une configuration qui minimise le contribution induite par le signal d'entrée $\{u(n)\}$ et celle induite par le bruit $\{v(n)\}$. Ceci confirme que la présence de bruit risque de biaiser la solution minimum, sauf si la minimisation du terme induit par le bruit donne la même solution qu'à la minimisation du terme induit par le signal.

Considérons le cas où le bruit de sortie est blanc. Dans ce cas, $\mathbf{R}_{vv} = \sigma_v^2 \mathbf{I}$ où σ_v^2 désigne la variance du bruit. La fonction de coût réduite dans (5.4) devient

$$J(\hat{\mathbf{a}}) = \hat{\mathbf{a}}^t \mathbf{R}_{y/u} \hat{\mathbf{a}} + \sigma_v^2 (a_0^2 + a_1^2 + \dots + a_M^2)$$

Nous avons retenu la contrainte $a_0 = 1$ jusqu'ici (appelé contrainte *monique*) pour éviter la solution triviale $\hat{\mathbf{a}} = \mathbf{0}$. Cette dernière forme, toutefois, suggère qu'une contrainte mieux appropriée serait $a_0^2 + a_1^2 + \dots + a_M^2 = 1$, c'est-à-dire une contrainte à *norme unité*, car la contribution induite par le bruit se réduirait à une constante σ_v^2 , ce qui ne change pas le point minimum de la fonction de coût réduite. La solution optimum avec cette nouvelle contrainte se caractérise par une équation propre :

$$\mathbf{R}_{d/u} \hat{\mathbf{a}} = \lambda_{\min} \hat{\mathbf{a}} \quad (5.6)$$

où λ_{\min} est la valeur propre minimum de $\mathbf{R}_{d/u}$; la valeur de la fonction de coût ainsi minimisée devient $J_{\min} = \hat{\mathbf{a}}^t \mathbf{R}_{d/u} \hat{\mathbf{a}} = \lambda_{\min}$.

5.3.1. Développement Algorithmique

La minimisation de $J(\hat{\mathbf{a}}, \hat{\mathbf{b}})$ peut se faire en estimant les différentes matrices de covariance du paragraphe précédent, suivi de la résolution d'un système d'équations linéaires en (5.5) (avec la contrainte monique) où d'une équation propre en (5.6) (avec la contrainte à norme unité). On peut aussi envisager une recherche par gradient stochastique donnant lieu à un filtre adaptatif, comme nous le poursuivons à présent.

Considérons d'abord la contrainte monique ($a_0 = 1$). Un algorithme de recherche par gradient est obtenue en ajustant les coefficients dans la direction négative du gradient :

$$\begin{aligned} a_k(n+1) &= a_k(n) - \frac{\mu}{2} \frac{\partial E[e^2(n)]}{\partial a_k} \Big|_{a_k = a_k(n)}, & k = 1, 2, \dots, M; \\ b_k(n+1) &= b_k(n) - \frac{\mu}{2} \frac{\partial E[e^2(n)]}{\partial b_k} \Big|_{b_k = b_k(n)}, & k = 0, 1, \dots, M. \end{aligned}$$

Puisque $e(n) = \hat{\mathbf{a}}^t(n) \mathbf{d}(n) - \hat{\mathbf{b}}^t(n) \mathbf{u}(n)$, l'algorithme s'exprime en fonction des signaux filtrés par

$$\begin{aligned} a_k(n+1) &= a_k(n) - \mu E[e(n) d(n-k)], & k = 1, 2, \dots, M; \\ b_k(n+1) &= b_k(n) + \mu E[e(n) u(n-k)], & k = 0, 1, \dots, M. \end{aligned}$$

Les intercorrélations $E[e(n) d(n-k)]$ et $E[e(n) u(n-k)]$ n'étant pas disponibles en pratiques, elles sont remplacées par leurs estimées instantanées, ce qui donne un algorithme de gradient stochastique :

$$\begin{aligned} a_k(n+1) &= a_k(n) - \mu e(n) d(n-k), & k = 1, 2, \dots, M; \\ b_k(n+1) &= b_k(n) + \mu e(n) u(n-k), & k = 0, 1, \dots, M. \end{aligned}$$

L'algorithme peut être vu comme un filtre adaptatif RIF à deux canaux (voir fig. 5.1), et une étude de convergence est très proche de celle de l'algorithme LMS (voir chapitres 2 pour LMS et 3 pour LMS-RII). Pour l'algorithme à norme unité, la modification de l'algorithme devient

$$\begin{aligned} \alpha_k(n+1) &= a_k(n) - \mu E\{e(n) [d(n-k) - a_k(n)]\} \\ a_k(n+1) &= \alpha_k(n+1) / \sqrt{\sum_{i=0}^M \alpha_i^2(n+1)}, & k = 0, 1, \dots, M; \\ b_k(n+1) &= b_k(n) + \mu E[e(n) u(n-k)], & k = 0, 1, \dots, M. \end{aligned}$$

qui nécessite une étape de normalisation pour imposer la contrainte à norme unité à chaque pas. En remplaçant encore les intercorrélations par leurs estimées instantanées, on arrive à un algorithme de gradient stochastique :

$$\begin{aligned}\alpha_k(n+1) &= \alpha_k(n) - \mu e(n) [d(n-k) - \alpha_k(n)] \\ \alpha_k(n+1) &= \alpha_k(n+1) / \sqrt{\sum_{i=0}^M \alpha_i^2(n+1)}, \quad k = 0, 1, \dots, M; \\ b_k(n+1) &= b_k(n) + \mu e(n) u(n-k), \quad k = 0, 1, \dots, M.\end{aligned}$$

5.3.2. Propriétés de Modélisation

Le développement jusqu'ici a été motivé par le cas où le système inconnu $H(z)$ a une fonction de transfert sous forme rationnelle, et de degré connu. Dans ce cas précis, les deux versions de l'algorithme (avec contrainte monique ou contrainte à norme unitaire) identifient correctement les paramètres du système, si le bruit $\{v(n)\}$ s'annule : la fonction de transfert

$$\hat{H}(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \dots + a_M z^{-M}}$$

coïncide avec la fonction $H(z)$ rencontrée précédemment dans l'équation (5.2), à condition que l'entrée $\{u(n)\}$ soit à excitation persistante de degré $2M + 1$ ou supérieur.²

En pratique, pourtant, une description exacte d'un système inconnu n'admet que très rarement une fonction de transfert rationnelle d'un degré convenable, en raison par exemple de la présence des effets de paramètres distribués et/ou des non linéarités mêmes subtiles. Dans ces cas, il n'est pas justifié de parler d'"identification" de système, car aucun réglage des coefficients $\{\hat{a}_k\}$ et $\{\hat{b}_k\}$ ne permet d'éliminer l'erreur de modélisation.

Il est important, alors, de vérifier si le modèle $\hat{H}(z)$ obtenu en un point minimum pourrait néanmoins prétendre à une forme d'approximation utile. Nous élaborons dans

2. Pour être sûr, les correspondances entre les coefficients sont

$$\begin{aligned}a_k &= \hat{a}_k / \hat{a}_0, \quad k = 1, 2, \dots, M; \\ b_k &= \hat{b}_k / \hat{a}_0, \quad k = 0, 1, \dots, M;\end{aligned}$$

pour l'algorithme à contrainte à norme unité. Avec la contrainte monique, on a déjà $\hat{a}_0 = 1$.

ce paragraphe certains résultats inattendus dans ce sens lorsque le vrai système est linéaire et invariant dans le temps [admettant donc un fonction de transfert notée $H(z)$], mais de degré strictement supérieur au degré M choisi pour le modèle $\widehat{H}(z)$; on parle alors du cas *sous modélisé*.

Le première question à examiner porte sur la stabilité de la fonction $\widehat{H}(z) = \widehat{B}(z)/\widehat{A}(z)$ obtenu en un point minimum, ce qui revient à examiner si son dénominateur

$$\widehat{A}(z) = \hat{a}_0 + \hat{a}_1 z^{-1} + \dots + \hat{a}_M z^{-M}$$

est à minimum de phase, c'est-à-dire, ayant toutes ces racines à l'intérieur du disque unité du plan complexe. Quelques exemples soumis dans [SÖD 81] ont permis d'apprécier que parfois la réponse est non dans le cas sous modélisé, ce qui donne une conclusion très défavorable dans la mesure où un modèle instable est sans utilité.

Heureusement, en limitant l'entrée $\{u(n)\}$ à une classe restreinte des signaux, la stabilité de $\widehat{H}(z)$ au point minimum peut être théoriquement démontrée [MUL 76], [REG 94], [LOP 99]. Plus précisément, nous rappelons qu'un processus $\{u(n)\}$ est autoregressif de degré N , s'il existe des coefficients $\gamma_1, \dots, \gamma_N$ tel qu'un signal résidu $\epsilon(n)$ formé selon

$$\epsilon(n) = u(n) + \gamma_1 u(n-1) + \dots + \gamma_N u(n-N)$$

soit blanc : $E[\epsilon(n)\epsilon(m)] = 0$ pour tout $n \neq m$, et $E[\epsilon^2(n)] > 0$. Notons qu'un processus autoregressif de degré $N-1$ est inclus dans cette définition comme cas particulier (obtenue avec $\gamma_N = 0$), et ainsi de suite pour tous les degrés inférieurs à N , jusqu'à degré 0 ($\gamma_1 = \dots = \gamma_N = 0$) obtenue lorsque la séquence $\{u(n)\}$ est déjà blanche.

Le résultat suivant a été démontré dans [MUL 76] pour une entrée blanche, et puis étendu aux processus autoregressifs dans [REG 94] et [LOP 99] :

Résultat 1 *Soit M le degré choisi pour le modèle $\widehat{H}(z)$. Si l'entrée $\{u(n)\}$ est un processus autoregressif de degré $M+1$ (ou inférieur), le polynôme $\widehat{A}(z)$ obtenu au point minimum est à minimum de phase (donc $\widehat{H}(z)$ sera stable), qu'on utilise la contrainte monique ou la contrainte à norme unité.*

Nous examinons par la suite certaines caractérisations de la solution en termes de la réponse impulsionnelle et la fonction d'autocorrélation associées à $\widehat{H}(z)$. Par souci de simplicité, nous nous limitons au cas où l'entrée $\{u(n)\}$ est un bruit blanc; des généralisations au cas où l'entrée est autoregressif sont toutefois disponibles dans [REG 94].

Ecrivons le vrai système $H(z)$ et son modèle $\hat{H}(z)$ en termes de leurs réponses impulsionnelles respectives :

$$H(z) = \sum_{k=0}^{\infty} h_k z^{-k} \quad \text{et} \quad \hat{H}(z) = \sum_{k=0}^{\infty} \hat{h}_k z^{-k}$$

Mullis et Roberts [MUL 76] ont montré que les débuts de ces séquences coïncident toujours :

$$\hat{h}_k = h_k, \quad k = 0, 1, \dots, M, \quad (5.7)$$

où l'on rappelle que M est le degré choisi pour notre modèle $\hat{H}(z)$. Introduisons maintenant les fonctions d'autocorrélation pour les deux systèmes :

$$r_k = \sum_{i=0}^{\infty} h_i h_{i+k} \quad \text{et} \quad \hat{r}_k = \sum_{i=0}^{\infty} \hat{h}_i \hat{h}_{i+k}$$

En utilisant la contrainte à norme unité, les débuts de ces deux séquences sont liées par une formule simple [REG 94] :

$$r_k - \hat{r}_k = \begin{cases} E[v(n)v(n+k)], & k = 1, 2, \dots, M; \\ \lambda_{\min}(\mathbf{R}_{d/u}), & k = 0. \end{cases} \quad (5.8)$$

Si le bruit de sortie est blanc, alors $E[v(n)v(n+k)] = 0$ pour $k \neq 0$, ce qui donne la coïncidence entre les termes d'indices 1 jusqu'à M dans les fonctions d'autocorrélation.

En utilisant la contrainte monique, l'expression devient plus compliquée [MUL 76] :

$$r_k - \hat{r}_k = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} \frac{e^{jk\omega}}{|\hat{A}(e^{j\omega})|^2} d\omega, \quad k = 0, 1, \dots, M. \quad (5.9)$$

Notons que les écarts entre les termes des fonctions d'autocorrélation dépendent de la solution $\hat{A}(z)$ recherchée, ce qui rend cette formule moins explicite à évaluer.

Nous notons enfin que ces équations caractérisent les solutions recherchées : Une fraction rationnelle $\hat{H}(z) = \hat{B}(z)/\hat{A}(z)$ vérifie les deux propriétés d'interpolation (5.7) et (5.8) (respectivement, (5.7) et (5.9)) si et seulement si elle est obtenue en minimisant la variance de l'erreur d'équation avec une contrainte à norme unité (resp., contrainte monique), lorsque l'entrée est blanche. Cette caractérisation se généralise aux entrées autoregressifs [REG 94], bien que certaines formules paraissent légèrement plus compliquées.

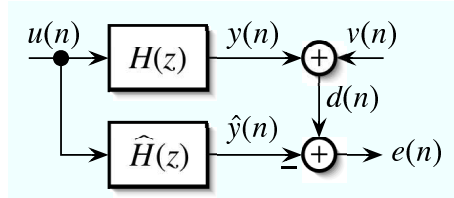


Figure 5.2. Configuration en erreur de sortie.

5.4. Méthode de minimisation de l'erreur de sortie

Bien que l'algorithme de minimisation de l'erreur d'équation soit simple, la stabilité du modèle $\hat{H}(z)$ au point minimum n'est garantie que si l'entrée $\{u(n)\}$ est autoregressive d'un degré convenable. Sinon, le modèle $\hat{H}(z)$ obtenu n'a pas de raison fondamentale d'être une bonne approximation du système inconnu $H(z)$. Nous examinons donc une autre critère qui consiste à minimiser directement la variance de l'erreur de sortie :

$$e(n) = d(n) - \hat{y}(n),$$

dont un schéma est donné dans la Figure 5.2. Cette minimisation est intimement liée à une méthode d'identification basée sur le principe de maximum de vraisemblance [SÖD 75], [SÖD 82].

La fonction de coût associée est la variance de l'erreur de sortie, obtenue lorsque les coefficients $\{\hat{a}_k\}$ et $\{\hat{b}_k\}$ sont fixes :

$$J(\hat{\mathbf{a}}, \hat{\mathbf{b}}) = E\{[d(n) - \hat{y}(n)]^2\}.$$

Si le bruit de sortie $\{v(n)\}$ est décorrélé de tous les échantillons d'entrée :

$$E[v(n) u(m)] = 0, \quad \text{pour tout } m, n ;$$

alors il est décorrélé des sorties $y(n)$ et $\hat{y}(n)$ aussi, car

$$\begin{aligned} E[v(n) y(n)] &= E\left(v(n) \sum_{k=0}^{\infty} h_k u(n-k)\right) \\ &= \sum_{k=0}^{\infty} h_k \underbrace{E[v(n) u(n-m)]}_0 = 0, \end{aligned}$$

avec une vérification analogue pour montrer que $E[v(n) \hat{y}(n)] = 0$. Ainsi, la présence d'un bruit de sortie ne modifie la fonction de coût que par l'addition d'une constante

$E[v^2(n)] :$

$$\begin{aligned} J(\hat{\mathbf{a}}, \hat{\mathbf{b}}) &= E\{[d(n) - \hat{y}(n)]^2\} \\ &= E\{y(n) - \hat{y}(n) + v(n)\}^2 \\ &= E\{[y(n) - \hat{y}(n)]^2\} + \underbrace{2 E\{[y(n) - \hat{y}(n)] v(n)\}}_0 + E[v^2(n)] \end{aligned}$$

Minimiser $J(\hat{\mathbf{a}}, \hat{\mathbf{b}})$ revient alors à minimiser $E\{[y(n) - \hat{y}(n)]^2\}$ qui, à son tour, s'exprime dans le domaine fréquentiel comme une norme L_2 pondérée de la différence entre la fonction inconnu $H(z)$ et son modèle $\hat{H}(z)$:

$$E\{[y(n) - \hat{y}(n)]^2\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{S}_{uu}(e^{i\omega}) |H(e^{i\omega}) - \hat{H}(e^{i\omega})|^2 d\omega.$$

La fonction de pondération $\mathcal{S}_{uu}(e^{i\omega})$ est la densité spectrale de puissance du signal d'entrée :

$$\mathcal{S}_{uu}(e^{i\omega}) = \sum_{k=-\infty}^{\infty} E[u(n) u(n-k)] e^{ik\omega}$$

Si l'on arrive à trouver le minimum du critère, on est assuré que la fonction $\hat{H}(z)$ résultante sera une meilleure approximation de la fonction $H(z)$ originelle.

L'inconvénient principal de cette approche concerne l'absence de l'unicité du minimum dans le cas général. En effet, la fonction de coût n'est pas quadratique vis à vis des coefficients $\{\hat{a}_k\}$ du dénominateur du modèle $\hat{H}(z)$, et des minima locaux peuvent paraître dans bien de cas. Certaines conditions assurent toutefois l'unicité : si le degré de $H(z)$ est connu, et si l'on choisit ce même degré pour $\hat{H}(z)$, la fonction de coût aura un minimum unique [caractérisé par $\hat{H}(z) = H(z)$] dès lors que l'entrée $\{u(n)\}$ est un bruit blanc ou un processus autoregressif de degré 1 [REG 95]. En admettant que ces conditions sont un peu spéciales, le problème de minima locaux dans le cas général demeure. Toutefois, la pertinence du modèle résultant au minimum global justifie l'intérêt de la méthode.

Le calcul des gradients est parfois plus facile à mener en utilisant les fonctions de transfert dans le domaine fréquentiel, bien que le développement des algorithmes adaptatifs passe plus directement dans le domaine temporel. Nous utilisons dans la suite alors une notation mixte, qui se comprend sans difficulté en interprétant " z^{-1} " comme un opérateur de retard : $z^{-1}u(n) = u(n-1)$. Ainsi, si $H(z) = \sum_k h_k z^{-k}$, le produit $H(z)u(n)$ s'interprète comme une somme de convolution, car

$$H(z)u(n) = \sum_{k=0}^{\infty} h_k z^{-k}u(n) = \sum_{k=0}^{\infty} h_k u(n-k)$$

ce qui donne la sortie $y(n)$ d'un filtre attaqué par $u(n)$ et ayant comme fonction de transfert $H(z)$.

Avec cette notation mixte, on peut écrire, pour l'erreur de sortie

$$e(n) = y(n) - \hat{y}(n) + v(n) = [H(z) - \hat{H}(z)] u(n) + v(n).$$

En paramétrant $\hat{H}(z)$ comme une fraction rationnelle :

$$\hat{H}(z) = \frac{\hat{B}(z)}{\hat{A}(z)} = \frac{\hat{b}_0 + \hat{b}_1 z^{-1} + \dots + \hat{b}_M z^{-M}}{1 + \hat{a}_1 z^{-1} + \dots + \hat{a}_M z^{-M}}$$

il reste à développer un algorithme de réactualisation des coefficients, en suivant le gradient négatif de la fonction de coût $J = E[e^2(n)]$:

$$\begin{aligned} \hat{a}_k(n+1) &= \hat{a}_k(n) - \frac{\mu}{2} \left. \frac{\partial E[e^2(n)]}{\partial \hat{a}_k} \right|_{\hat{a}_k = \hat{a}_k(n)} \\ \hat{b}_k(n+1) &= \hat{b}_k(n) - \frac{\mu}{2} \left. \frac{\partial E[e^2(n)]}{\partial \hat{b}_k} \right|_{\hat{b}_k = \hat{b}_k(n)} \end{aligned}$$

Pour le calcul de gradient, notons que

$$\begin{aligned} \frac{\partial E[e^2(n)]}{\partial \hat{a}_k} &= 2 E \left(e(n) \frac{\partial \hat{y}(n)}{\partial \hat{a}_k} \right) \\ \frac{\partial E[e^2(n)]}{\partial \hat{b}_k} &= 2 E \left(e(n) \frac{\partial e(n)}{\partial \hat{b}_k} \right) = -2 E \left(e(n) \frac{\partial \hat{y}(n)}{\partial \hat{b}_k} \right) \end{aligned}$$

ce qui fait apparaître les signaux de sensibilité $\partial \hat{y}(n) / \partial \hat{a}_k$ et $\partial \hat{y}(n) / \partial \hat{b}_k$. Par un calcul direct, ceux-ci deviennent

$$\begin{aligned} \frac{\partial \hat{y}(n)}{\partial \hat{a}_k} &= \frac{\partial \hat{H}(z)}{\partial \hat{a}_k} u(n) = \frac{-z^{-k}}{\hat{A}(z)} \hat{H}(z) u(n) \\ \frac{\partial \hat{y}(n)}{\partial \hat{b}_k} &= \frac{\partial \hat{H}(z)}{\partial \hat{b}_k} u(n) = \frac{z^{-k}}{\hat{A}(z)} u(n) \end{aligned}$$

La génération des signaux de sensibilité est illustrée dans la figure 5.3, et l'algorithme de gradient se résume alors comme suit :

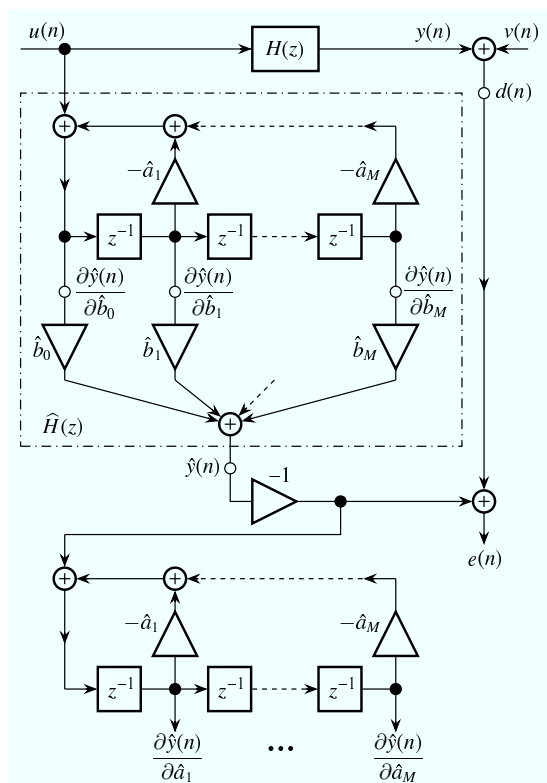


Figure 5.3. Génération des signaux de sensibilité pour la configuration en erreur de sortie.

- Disponible à l'instant n :
signaux d'état $\partial \hat{y}(n-1)/\partial \hat{b}_k$ et $\partial \hat{y}(n-1)/\partial \hat{a}_k$, $k = 0, 1, \dots, M-1$.
- Récurrence d'état :

$$\frac{\partial \hat{y}(n)}{\partial \hat{b}_k} = \frac{\partial \hat{y}(n-1)}{\partial \hat{b}_{k-1}} \quad k = 1, 2, \dots, M ;$$

$$\frac{\partial \hat{y}(n)}{\partial \hat{b}_0} = u(n) - \sum_{k=1}^M \hat{a}_k(n-1) \frac{\partial \hat{y}(n)}{\partial \hat{b}_k}$$
- Sortie du filtre :

$$\hat{y}(n) = \sum_{k=0}^M \hat{b}_k(n-1) \frac{\partial \hat{y}(n)}{\partial \hat{b}_k}$$

$$e(n) = d(n) - \hat{y}(n)$$
- Post filtre :

$$\frac{\partial \hat{y}(n)}{\partial \hat{a}_k} = \frac{\partial \hat{y}(n-1)}{\partial \hat{a}_{k-1}} \quad k = 1, 2, \dots, M ;$$

$$\frac{\partial \hat{y}(n)}{\partial \hat{a}_0} = \hat{y}(n) - \sum_{k=1}^M \hat{a}_k(n-1) \frac{\partial \hat{y}(n)}{\partial \hat{a}_k}$$

- Réactualisation des coefficients :

$$\hat{a}_k(n+1) = \hat{a}_k(n) + \mu e(n) \frac{\partial \hat{y}(n)}{\partial \hat{a}_k}, \quad k = 1, 2, \dots, M ;$$

$$\hat{b}_k(n+1) = \hat{b}_k(n) + \mu e(n) \frac{\partial \hat{y}(n)}{\partial \hat{b}_k}, \quad k = 0, 1, \dots, M ;$$

En pratique, l'algorithme devrait être accompagné d'un test de stabilité, qui consiste à examiner périodiquement si le polynôme "gelé" $\hat{A}(z, n) = 1 + \sum_k \hat{a}_k(n) z^{-k}$ est à minimum de phase.³

L'inconvénient principal de la méthode de gradient en erreur de sortie est la possibilité que l'algorithme se piège dans un minimum local. Cette considération nous motive pour examiner d'autres algorithmes qui sont pilotés par l'erreur de sortie, sans toutefois suivre le gradient de sa variance. Ainsi, un minimum local dans la variance de sortie n'a aucune raison de paraître comme un point attracteur d'un algorithme qui ne suit pas ce gradient. Par ce même raisonnement, on constate que le minimum global n'aura pas de raison non plus de paraître comme un point attracteur, à cette différence près que, au moins dans le cas de modélisation exacte, l'erreur de sortie s'annule identiquement au minimum global si le bruit de sortie est absent. Nous présentons deux approches alternatives dans les paragraphes suivants qui visent à éliminer l'erreur de sortie : la méthode de Steiglitz-McBride, et les algorithmes fonctionnant selon le principe d'hyperstabilité.

5.5. Méthode de Steiglitz et McBride

Cette méthode a été proposée d'abord comme une méthode itérative [STE 65] qui fonctionne en temps différé, puis réinterprétée comme un algorithme en filtrage adaptatif [STO 81], [FAN 86], capable de fonctionner en temps réel. Nous développons d'abord la version en temps différé, et puis nous expliquons comment l'algorithme en temps réel en résulte.

Revenons à la minimisation d'une erreur d'équation, qui utilise non pas les séquences d'entrée et de référence directement, mais plutôt des versions pré-filtrées de

3. A parler proprement, ce test de stabilité n'est pas valable dans le cas non stationnaire général (voir, par exemple, [REG 95, exemple 6.3 et problème 6.9]), mais s'avère utilisable si les coefficients varient suffisamment lentement, condition obtenue lorsque le pas d'adaptation μ est choisi suffisamment petit.

ces dernières. Plus précisément, soit k un indice d'itération, et supposons qu'il nous reste un préfiltre $1/\hat{A}_{k-1}(z)$ depuis l'itération précédente :

$$\frac{1}{\hat{A}_{k-1}(z)} = \frac{1}{1 + \hat{a}_{1,k-1} z^{-1} + \dots + \hat{a}_{M,k-1} z^{-M}}$$

Nous appliquons ce préfiltre aux séquences d'entrée $\{u(n)\}$ et de référence $\{d(n)\}$ afin de générer des signaux préfiltrés, notés $\{u'(n)\}$ et $\{d'(n)\}$ [voir figure 5.4(a)]. A partir de ces séquences préfiltrées, nous cherchons deux polynômes, notés

$$\begin{aligned}\hat{A}_k(z) &= 1 + \hat{a}_{1,k} z^{-1} + \dots + \hat{a}_{M,k} z^{-M} \\ \hat{B}_k(z) &= \hat{b}_{0,k} + \hat{b}_{1,k} z^{-1} + \dots + \hat{b}_{M,k} z^{-M}\end{aligned}$$

qui minimisent la variance d'une erreur d'équation :

$$\begin{aligned}e(n) &= \hat{A}_k(z) d'(n) - \hat{B}_k(z) u'(n) \\ &= d'(n) + \sum_{j=1}^M \hat{a}_{j,k} d'(n-j) - \sum_{j=0}^M \hat{b}_{j,k} u'(n-j)\end{aligned}$$

Le polynôme $\hat{A}_k(z)$ obtenu au point minimum est inversé en $1/\hat{A}_k(z)$, ce dernier remplaçant le préfiltre pour la prochaine itération. La procédure continue jusqu'à ce qu'un point stationnaire s'obtienne, caractérisé par $A_k(z) = A_{k-1}(z)$; comme nous voyons dans la figure 5.4(b), à un point stationnaire, la branche à droite se simplifie en un chemin directe, tandis que la branche à gauche rend le modèle rationnel $\hat{H}(z) = \hat{B}_k(z)/\hat{A}_k(z)$. Ainsi, à un point stationnaire, l'erreur d'équation avec préfiltrage se transforme en une erreur de sortie.

5.5.1. Développement Algorithmique

Considérons d'abord le cas où le préfiltre $1/\hat{A}_{k-1}(z)$ est fixe. La minimisation de l'erreur d'équation peut faire appel à un algorithme de gradient, adapté directement depuis le paragraphe 5.3.1. D'abord, les signaux préfiltrés sont générés par les récurrences autoregressives utilisant les valeurs finales de l'itération $k-1$:

$$\begin{aligned}d'(n) &= d(n) - \sum_{j=1}^M \hat{a}_{j,k-1} d'(n-j) \\ y'(n) &= y(n) - \sum_{j=1}^M \hat{a}_{j,k-1} u'(n-j)\end{aligned}\tag{5.10}$$

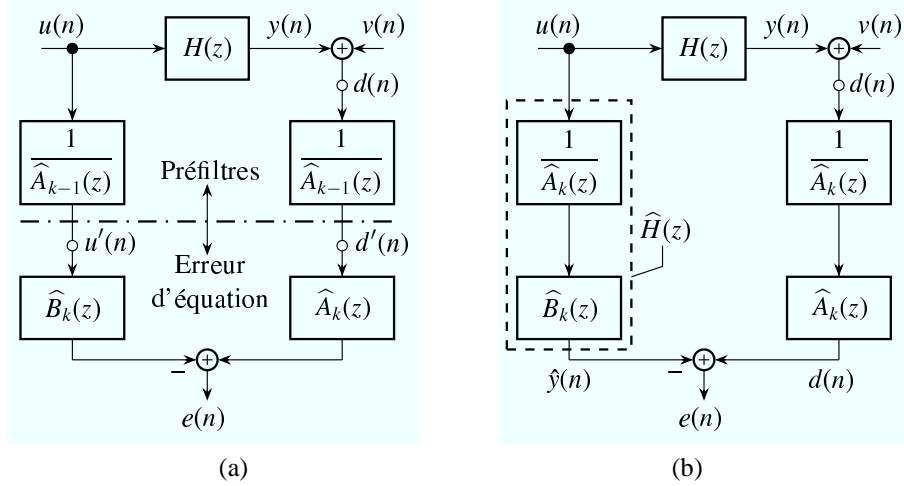


Figure 5.4. (a) Méthode en erreur d'équation appliquée aux signaux pré-filtrés ; (b) système en erreur de sortie obtenu en un point stationnaire.

L'algorithme de gradient s'écrit alors

$$e(n) = d'(n) + \sum_{j=1}^M \hat{a}_{j,k}(n) d'(n-j) - \sum_{j=0}^M \hat{b}_{j,k}(n) u'(n-j) \quad (5.11)$$

$$\hat{a}_{j,k}(n+1) = \hat{a}_{j,k}(n) - \mu e(n) d'(n-j), \quad j = 1, 2, \dots, M; \quad (5.12)$$

$$\hat{b}_{j,k}(n+1) = \hat{b}_{j,k}(n) + \mu e(n) u'(n-j), \quad j = 0, 1, \dots, M. \quad (5.13)$$

Comme écrit, il nous faut trois indices pour chaque coefficient : j est l'indice de coefficient dans le polynôme, k indique l'itération en cours, et n désigne l'échantillon temporel pendant la k -ième itération. En principe, on peut fixer k et laisser n augmenter jusqu'à ce que l'algorithme converge, après quoi les préfiltres seront réactualisés en $1/\hat{A}_k(z)$, le compteur n remis à 0, et le $k + 1$ -ième itération pourra commencer. Cette approche donne ainsi une séquence des problèmes à résoudre, avec le résultat d'un problème fournissant le préfiltre pour le problème suivant.

Une méthode pour accélérer la procédure est de réactualiser les préfiltres en même temps que les coefficients, ce qui est obtenu en remplaçant $\hat{a}_{j,k-1}$ dans le préfiltre par $\hat{a}_j(n)$ fourni par l'algorithme d'adaptation ; l'indice k (d'itération) devient alors

superflu. Ainsi, les séquence préfiltrées de (5.10) deviennent des récurrences non stationnaires (car à coefficients variables) :

$$d'(n) = d(n) - \sum_{j=1}^M \hat{a}_j(n) d'(n-j) \quad (5.14)$$

$$y'(n) = y(n) - \sum_{j=1}^M \hat{a}_j(n) u'(n-j) \quad (5.15)$$

La première ligne permet d'exprimer $d(n)$ sous la forme

$$d(n) = d'(n) + \sum_{j=1}^M \hat{a}_j(n) d'(n-j),$$

qui est reconnue comme le premier terme du membre à droite de (5.11) dans le calcul de l'erreur d'équation (après avoir enlevé l'indice k superflu). Ainsi, on constate que l'erreur d'équation avec préfiltrage se transforme en une erreur de sortie :

$$e(n) = \underbrace{d'(n) + \sum_{j=1}^M \hat{a}_j(n) d'(n-j)}_{d(n)} - \underbrace{\sum_{j=0}^M \hat{b}_j(n) u'(n-j)}_{\hat{y}(n)}$$

En retenant la réactualisation des coefficients de (5.12) et (5.13), et la réactualisation des signaux pré-filtrés de (5.14) et (5.15), nous avons l'algorithme de Steiglitz-McBride en forme de filtre RII adaptatif. L'algorithme est résumé ainsi :

Disponible à l'itération n :

Coefficients du filtre

$$\hat{a}_k(n), \quad k = 1, 2, \dots, M;$$

$$\hat{b}_k(n), \quad k = 0, 1, \dots, M;$$

Etats des préfiltres

$$u'(n-k) \text{ et } d'(n-k), \quad k = 1, 2, \dots, M;$$

Nouvelles données : échantillon d'entrée $u(n)$; échantillon de référence $d(n)$.

Calcul d'erreur de sortie :

$$u'(n) = u(n) - \sum_{k=1}^M \hat{a}_k(n) u'(n-k)$$

$$\hat{y}(n) = \sum_{k=0}^M \hat{b}_k(n) u'(n-k)$$

$$e(n) = d(n) - \hat{y}(n)$$

Réactualisation des coefficients :

$$\hat{a}_k(n+1) = \hat{a}_k(n) - \mu e(n) y'(n-k), \quad k = 1, 2, \dots, M;$$

$$\hat{b}_k(n+1) = \hat{b}_k(n) + \mu e(n) u'(n-k), \quad k = 0, 1, \dots, M;$$

Préfiltre de sortie :

$$y'(n) = y(n) - \sum_{k=0}^M \hat{a}_k(n+1) y'(n-k).$$

5.6. Algorithmes à Base d'Hyperstabilité

Les algorithmes dans cette classe remontent aux travaux de I. LANDAU [LAN 76] qui a reconnu l'intérêt d'appliquer la théorie de stabilité des systèmes non linéaires pour établir la convergence d'un système adaptatif.

Considérons la configuration illustrée dans la figure 5.5, comportant un système linéaire et invariant dans le temps, caractérisé par une fonction de transfert $\mathcal{F}(z)$, bouclé par une loi de contre réaction qui est, en général, non linéaire et non stationnaire. Le théorème d'hyperstabilité [POP 63], [AND 68] affirme que le système bouclé est asymptotiquement stable, en ce sens que $\epsilon(n) \xrightarrow{n \rightarrow \infty} 0$ pour toute condition initiale, si les deux conditions suivantes sont réunies :

1) La fonction de transfert de la partie linéaire et invariante dans le temps est strictement réelle positive (SPR) Ceci veut dire que la partie réelle de sa réponse fréquentielle est strictement positive :

$$\operatorname{Re} \mathcal{F}(e^{i\omega}) > 0, \quad \text{pour tout } \omega.$$

Cette contrainte équivaut à dire que la valeur absolue de la phase de $\mathcal{F}(e^{i\omega})$ est strictement inférieure à $\pi/2$ radians pour tout ω .

2) La loi de contre réaction vérifie l'inégalité de Popov :

$$\sum_{k=0}^n \epsilon(n) s(n) \geq -\gamma^2, \quad \text{pour tout } n,$$

pour une certaine constante γ qui peut dépendre de la condition initiale. L'inégalité indique que l'excursion négative de la somme est bornée ; l'excursion positive peut, en revanche, être non bornée.

Différentes démonstrations sont disponibles [POP 63], [AND 68], [REG 95] ; nous nous contentons ici de montrer comment le problème d'identification peut être reformulé afin de tirer profit de ce résultat.

Pour commencer, faisons l'hypothèse que le système inconnu est véritablement une fraction rationnelle d'un degré connu, disons M , et supposons dans un premier

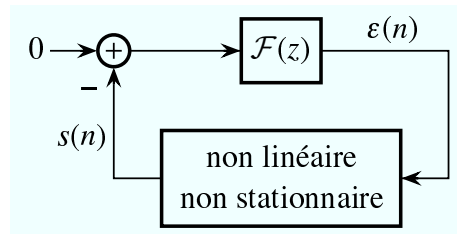


Figure 5.5. Système linéaire et invariant dans le temps $\mathcal{F}(z)$ bouclé par une loi de contre-réaction non linéaire et non stationnaire.

temps que le bruit de sortie $\{v(n)\}$ est absent. Dans ce cas, on retrouve $d(n) = y(n)$, où $y(n)$ vérifie

$$y(n) = \frac{B(z)}{A(z)} u(n)$$

ou encore

$$A(z) y(n) = B(z) u(n).$$

En écrivant explicitement les deux polynômes

$$\begin{aligned} A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M} \\ B(z) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M} \end{aligned}$$

et en se rappelant que z^{-1} désigne l'opérateur de retard, le comportement entrée-sortie prend la forme d'une équation aux différences :

$$y(n) = - \sum_{k=1}^M a_k y(n-k) + \sum_{k=0}^M b_k u(n-k).$$

Mettons maintenant en parallèle un modèle ajustable :

$$\hat{y}(n) = - \sum_{k=1}^M \hat{a}_k(n) \hat{y}(n-k) + \sum_{k=0}^M \hat{b}_k(n) u(n-k),$$

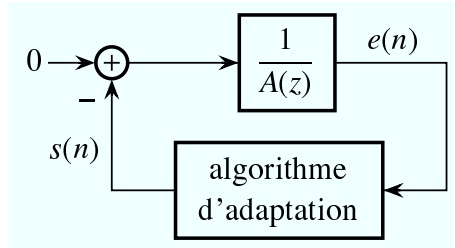


Figure 5.6. Première tentative d'écrire l'erreur de sortie dans la forme de la figure 5.5 ; la fonction $1/A(z)$ n'est pas toujours strictement réelle positive.

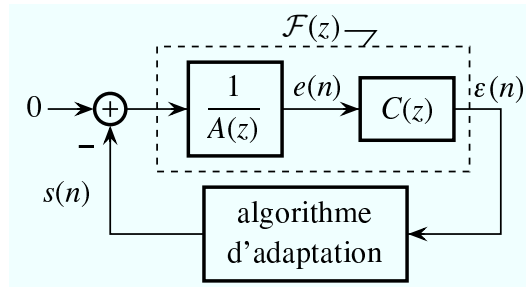


Figure 5.7. Version corrigée en ajoutant un filtre de compensation $C(z)$, choisi pour que $C(z)/A(z)$ soit strictement réelle positive.

où les coefficients $\{\hat{a}_k(n)\}$ et $\{\hat{b}_k(n)\}$ restent à déterminer. L'erreur de sortie $e(n) = y(n) - \hat{y}(n)$ vérifie ainsi une équation aux différences

$$\begin{aligned}
 e(n) &= - \sum_{k=1}^M a_k \underbrace{[y(n-k) - \hat{y}(n-k)]}_{e(n-k)} \\
 &+ \underbrace{\sum_{k=1}^M [\hat{a}_k(n) - a_k] \hat{y}(n-k) + \sum_{k=0}^M [b_k - \hat{b}_k(n)] u(n-k)}_{s(n)}
 \end{aligned}$$

Ceci admet une représentation schématique illustrée dans la figure 5.6. Le schéma commence à ressembler à celui de la figure 5.5, à une faute près : la fonction de transfert $1/A(z)$ n'est pas garantie d'être strictement réelle positive (SPR).

Puisque le signal d'erreur $e(n)$ est directement disponible, on peut envisager d'ajouter un filtre de compensation $C(z) = 1 + c_1 z^{-1} + \dots + c_M z^{-M}$ qui donne une

erreur filtrée, selon

$$\epsilon(n) = C(z) e(n) = e(n) + c_1 e(n-1) + \dots + c_M e(n-M),$$

comme illustré dans la figure 5.7. La partie linéaire et invariante dans le temps devient ainsi $C(z)/A(z)$, et en choisissant correctement les coefficients $\{c_k\}$, on peut s'assurer que la fonction $C(z)/A(z)$ devient strictement réelle positive. Pour s'en convaincre, notons qu'avec le choix $C(z) = A(z)$, on retrouve $C(z)/A(z) = 1$, qui est certainement strictement réelle positive. D'une manière plus générale, tant que les coefficients $\{c_k\}$ sont choisis dans un voisinage adéquat des coefficients $\{a_k\}$, la phase de $C(e^{i\omega})/A(e^{i\omega})$ a de bonnes chances de rester inférieur à $\pi/2$ radians, ce qui assure la réalité positive stricte.

Il reste maintenant à concevoir un algorithme d'adaptation qui impose la satisfaction de l'inégalité de Popov. Ceci a été achevé d'abord par Landau [LAN 76], qui a développé un algorithme dont la complexité de calcul était de $\mathcal{O}(M^2)$ opérations par itération. Un algorithme simplifié a été proposé par Johnson [JOH 79], algorithme qui s'avère être un cas limite de l'algorithme de Landau, mais dont la complexité se ramène à $\mathcal{O}(M)$ opérations par itération. L'algorithme a été baptisé HARF (= Hyperstable Adaptive Recursive Filter), et a la particularité de nécessiter la propagation des erreurs a priori (c'est-à-dire, ce qui résulte avant de réactualiser les coefficients) et a posteriori (après réactualisation les coefficients); il se résume ainsi :

Algorithme HARF

Disponible à l'itération n :

– Coefficients du filtre adaptatif :

$$\hat{a}_k(n), \quad k = 1, 2, \dots, M;$$

$$\hat{b}_k(n), \quad k = 0, 1, \dots, M.$$

– Etats du filtre adaptatif :

$$u(n-k) \quad \text{et} \quad x(n-k), \quad k = 1, 2, \dots, M;$$

– Coefficients et états du filtre de compensation :

$$c_k \quad \text{et} \quad \bar{e}(n-k), \quad k = 1, 2, \dots, M.$$

Nouvelles données : échantillon d'entrée $u(n)$; échantillon de référence $d(n)$.

Calcul d'erreur a priori :

$$\begin{aligned}\hat{y}(n) &= - \sum_{k=1}^M \hat{a}_k(n) x(n-k) + \sum_{k=0}^M \hat{b}_k(n) u(n-k) \\ e(n) &= d(n) - \hat{y}(n) \\ \epsilon(n) &= e(n) + \sum_{k=1}^M c_k \bar{e}(n-k)\end{aligned}$$

Réactualisation des coefficients :

$$\begin{aligned}\mu' &= \mu / \left(1 + \mu \sum_{k=1}^M x^2(n-k) + \mu \sum_{k=0}^M u^2(n-k) \right) \\ \hat{a}_k(n+1) &= \hat{a}_k(n) - \mu' \epsilon(n) x(n-k), \quad k = 1, 2, \dots, M ; \\ \hat{b}_k(n+1) &= \hat{b}_k(n) + \mu' \epsilon(n) u(n-k), \quad k = 0, 1, \dots, M.\end{aligned}$$

Erreur a posteriori

$$\begin{aligned}x(n) &= - \sum_{k=1}^M \hat{a}_k(n+1) x(n-k) + \sum_{k=0}^M \hat{b}_k(n+1) u(n-k) \\ \bar{e}(n) &= d(n) - x(n)\end{aligned}$$

Si le système inconnu est une fraction rationnelle de degré M , et si le bruit de sortie $\{v(n)\}$ est absent, on montre [JOH 79] que cet algorithme converge globalement pour tout $\mu > 0$ à condition, bien entendu, de choisir les coefficients $\{c_k\}$ du filtre de compensation de telle façon que la fonction $C(z)/A(z)$ soit strictement réelle positive. En présence d'un bruit de sortie, on montre [FAN 88] que les propriétés moyennes de convergence restent intactes ; en particulier, la présence d'un bruit de sortie ne peut pas biaiser le(s) point(s) de convergence.

5.7. Algorithme Hyperstable Simplifié

L'algorithme que nous venons de présenter nécessite le calcul des erreurs a priori et a posteriori, ainsi que des divisions. Nous montrons, toutefois, que lorsque μ est suffisamment petit, certaines approximations permettent d'arriver à un algorithme encore simplifié.

Pour commencer, notons que l'algorithme HARF utilise un pas d'adaptation normalisé, sous la forme

$$\mu' = \mu / \left(1 + \mu \sum_{k=1}^M x^2(n-k) + \mu \sum_{k=0}^M u^2(n-k) \right).$$

Pour μ petit, le dénominateur est presque constant, car

$$1 + \mu \sum_{k=1}^M x^2(n-k) + \mu \sum_{k=0}^M u^2(n-k) \approx 1, \quad \text{pour } \mu \text{ petit,}$$

ce qui donne $\mu' \approx \mu$.

Notons ensuite qu'une valeur faible pour μ implique que les coefficients ne changent que lentement :

$$\hat{a}_k(n+1) \approx \hat{a}_k(n) \quad \text{et} \quad \hat{b}_k(n+1) \approx \hat{b}_k(n)$$

Ainsi, la sortie a posteriori $x(n)$ est proche de la sortie a priori $y(n)$:

$$\begin{aligned} x(n) &= - \sum_{k=1}^M \hat{a}_k(n+1) x(n-k) + \sum_{k=0}^M \hat{b}_k(n+1) u(n-k) \\ &\approx - \sum_{k=1}^M \hat{a}_k(n) x(n-k) + \sum_{k=0}^M \hat{b}_k(n) u(n-k) \\ &= \hat{y}(n) \end{aligned}$$

A lieu de stocker donc les anciennes valeurs $\{x(n-k)\}_{k=1}^M$ dans le filtre récursif, on peut stocker leurs valeurs approchées $\{\hat{y}(n-k)\}_{k=1}^M$, sans trop changer le comportement du filtre adaptatif. De la même manière, les anciennes erreurs de sortie a posteriori $\{\bar{e}(n-k)\}_{k=1}^M$ cèdent leur place aux erreurs a priori $\{e(n-k)\}_{k=1}^M$. Avec ces modifications, on arrive au filtre hyperstable simplifié, baptisé SHARF (= Simplified Hyperstable Adaptive Recursive Filter) [LAR 80] :

Algorithme SHARF*Disponible à l'itération n :*

– Coefficients du filtre adaptatif :

$$\hat{a}_k(n), \quad k = 1, 2, \dots, M;$$

$$\hat{b}_k(n), \quad k = 0, 1, \dots, M.$$

– Etats du filtre adaptatif :

$$u(n-k) \quad \text{et} \quad y(n-k), \quad k = 1, 2, \dots, M;$$

– Coefficients et états du filtre de compensation :

$$c_k \quad \text{et} \quad e(n-k), \quad k = 1, 2, \dots, M.$$

Nouvelles données : échantillon d'entrée $u(n)$; échantillon de référence $d(n)$.*Calcul d'erreur de sortie :*

$$\hat{y}(n) = - \sum_{k=1}^M \hat{a}_k(n) \hat{y}(n-k) + \sum_{k=0}^M \hat{b}_k(n) u(n-k)$$

$$e(n) = d(n) - \hat{y}(n)$$

$$\epsilon(n) = e(n) + \sum_{k=1}^M c_k e(n-k)$$

Réactualisation des coefficients :

$$\hat{a}_k(n+1) = \hat{a}_k(n) - \mu \epsilon(n) \hat{y}(n-k), \quad k = 1, 2, \dots, M;$$

$$\hat{b}_k(n+1) = \hat{b}_k(n) + \mu \epsilon(n) u(n-k), \quad k = 0, 1, \dots, M.$$

5.8. Convergence des Algorithmes en Erreur de Sortie

A la différence de la méthode en erreur d'équation, dont la fonction de coût est quadratique, les méthodes en erreur de sortie soit présentent une fonction de coût non

quadratique (méthode de gradient en erreur de sortie), soit n'admettent pas d'interprétation en termes de recherche d'un point minimum (méthodes de Steiglitz-McBride et d'hyperstabilité). Leurs propriétés de convergence sont donc moins immédiates à déduire, et ne sont que partiellement maîtrisées dans le cas sous-modélisé [REG 95].

Toutefois, certaines propriétés de convergence peuvent être démontrées dans le cas de modélisation suffisante, grâce à la technique qui associe à un algorithme adaptatif une équation différentielle ordinaire. (voir Chapitre 3). Nous résumons dans ce paragraphe la stabilité des équations différentielles associées aux algorithmes en erreur de sortie, et présentons certains résultats applicables au cas sous-modélisé lors du paragraphe 5.9.

Commençons par ranger les coefficients $\{\hat{a}_k\}$ et $\{\hat{b}_k\}$ dans un vecteur \mathbf{w} :

$$\mathbf{w} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_M \\ \hat{b}_0 \\ \vdots \\ \hat{b}_M \end{bmatrix}$$

Les algorithmes présentés jusqu'ici prennent tous la forme

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \psi(n) \quad (5.16)$$

avec μ le pas d'adaptation, $e(n)$ un signal d'erreur, et $\psi(n)$ un vecteur de régresseurs filtrés, dont la forme exacte varie en fonction de l'algorithme. L'approche à base d'une équation différentielle associée s'applique lorsque le pas d'adaptation μ est choisi suffisamment petit et d'autres conditions de régularité sont vérifiées ; voir [BEN 87], [KUS 84], [FAN 88] pour des énoncés plus précis. Cette approche permet de faire un lien probabiliste entre l'algorithme de réactualisation (5.16) d'une part, et une équation différentielle d'autre part sous la forme

$$\frac{d\mathbf{w}(t)}{dt} = E[e(n) \psi(n) | \mathbf{w}] \triangleq f(\mathbf{w}) \quad (5.17)$$

où l'espérance du membre à droite est évaluée pour des coefficients fixes dans \mathbf{w} , le résultat variant en fonction de \mathbf{w} , ce qui donne le terme $f(\mathbf{w})$ qui pilote la dérivé temporelle $d\mathbf{w}/dt$ du membre à gauche. Pour μ petit, on montre [BEN 87], [KUS 84] qu'un point stationnaire, noté \mathbf{w}_* , sera un point attracteur de l'équation différentielle [5.17] si et seulement si ce même point est un point attracteur de l'équation de réactualisation [5.16], en ce sens "probabiliste" que

$$P(\|\mathbf{w}(n) - \mathbf{w}_*\| < C) > 1 - \delta, \quad \text{pour } n \text{ grand,}$$

où les constantes C et δ diminuent avec μ . D'une manière informelle, la convergence en probabilité se traduit par les estimées des coefficients $\mathbf{w}(n)$ s'approchant d'un voisinage d'un point attracteur \mathbf{w}_* et y restant par la suite, tout comme des mouches autour d'une ampoule électrique.

Nous examinons donc les équations différentielles qui s'associent aux algorithmes d'adaptations obtenus jusqu'ici.

5.8.1. Méthode de Gradient

Le vecteur régresseur de la méthode de gradient est choisi comme la dérivée négative de l'erreur de sortie :

$$\psi(n) = -\frac{1}{2} \frac{de(n)}{d\mathbf{w}}.$$

Le terme qui pilote la dérivée devient alors

$$\begin{aligned} f(\mathbf{w}) &= E[e(n)\psi(n)|\mathbf{w}] \\ &= -\frac{1}{2} E\left(e(n) \frac{de(n)}{d\mathbf{w}}\right) \\ &= -\frac{dE[e^2(n)]}{d\mathbf{w}} \end{aligned}$$

et l'équation différentielle associée prend la forme

$$\frac{d\mathbf{w}}{dt} = -\frac{dE[e^2(n)]}{d\mathbf{w}}. \quad (5.18)$$

Le comportement de cette équation différentielle se déduit aisément en considérant une fonction dite de Lyapunov, ici sous la forme :

$$L(\mathbf{w}) = E[e^2(n)|\mathbf{w}].$$

En tenant compte de l'équation différentielle (5.18), la fonction $L(\mathbf{w})$ est contrainte de décroître avec le temps :

$$\begin{aligned} \frac{dL}{dt} &= \left[\frac{dL}{d\mathbf{w}} \right]^t \frac{d\mathbf{w}}{dt} \\ &= -\left\| \frac{dE[e^2(n)]}{d\mathbf{w}} \right\|^2 \begin{cases} < 0, & dE[e^2]/d\mathbf{w} \neq 0; \\ = 0, & dE[e^2]/d\mathbf{w} = 0; \end{cases} \end{aligned}$$

où nous notons que $dL/d\mathbf{w} = -d\mathbf{w}/dt$ dans la dernière ligne. Ceci montre que la fonction $L(\mathbf{w})$, qui n'est autre que la variance de l'erreur de sortie en fonction des coefficients du filtre, diminue jusqu'à ce qu'un point minimum soit atteint. Ce comportement est, bien entendu, conforme à l'objectif de l'algorithme de gradient.

5.8.2. Algorithme de Steiglitz-McBride

En raison de la réactualisation en permanence des préfiltres dans la méthode de Steiglitz-McBride, l'algorithme ne suit pas le gradient d'une fonction de coût. Ainsi, l'analogie de la recherche d'un point minimum ne s'applique pas à cet algorithme. Toutefois, l'équation différentielle associée est convergente, au moins dans le cas où l'ordre M choisi pour le filtre adaptatif coïncide avec le degré du système à identifier [STO 81], [FAN 88].

Le vecteur regressor dans l'algorithme Steiglitz-McBride paraît comme

$$\psi(n) = \begin{bmatrix} -d'(n-1) \\ \vdots \\ -d'(n-M) \\ u'(n) \\ \vdots \\ u'(n-M) \end{bmatrix}$$

où les séquences $d'(n)$ et $u'(n)$ sont les sorties des filtres tout-pôle appliqués à $d(n)$ et à $u(n)$:

$$d'(n) = \frac{1}{\widehat{A}(z)} d(n) \quad u'(n) = \frac{1}{\widehat{A}(z)} u(n).$$

L'équation différentielle associée prend alors la forme

$$\frac{d\mathbf{w}}{dt} = E \left\{ e(n) \begin{bmatrix} -d'(n-1) \\ \vdots \\ -d'(n-M) \\ u'(n) \\ \vdots \\ u'(n-M) \end{bmatrix} \right\},$$

qui, en soi, s'avère difficile à analyser.

Pour rendre cette équation plus maniable, séparons les composantes signal $y(n)$ et bruit $v(n)$ dans la sortie observé $d(n)$, et examinons l'influence de chacune après préfiltrage :

$$d'(n) = y'(n) + v'(n), \quad \text{avec} \quad y'(n) = \frac{1}{\widehat{A}(z)} y(n) \quad \text{et} \quad v'(n) = \frac{1}{\widehat{A}(z)} v(n)$$

On montre [STO 81], [REG 95] que lorsque le bruit $\{v(n)\}$ est blanc, il ne fait aucune contribution à l'équation différentielle associée. Ceci permet d'étudier l'équation différentielle comme si le bruit de sortie était absent ; dans ce cas, elle vient

$$\frac{d\mathbf{w}}{dt} = E \left\{ e(n) \begin{bmatrix} -y'(n-1) \\ \vdots \\ -y'(n-M) \\ u'(n) \\ \vdots \\ u'(n-M) \end{bmatrix} \right\}. \quad (5.19)$$

Or, l'erreur de sortie $e(n)$ coïncide avec une erreur d'équation avec préfiltrage :

$$\begin{aligned} e(n) &= \widehat{A}(z) y'(n) - \widehat{B}(z) u'(n) \\ &= y'(n) + \sum_{k=1}^M \widehat{a}_k y'(n-k) + \sum_{k=0}^M \widehat{b}_k u'(n-k) \end{aligned} \quad (5.20)$$

Si le système à identifier a une fonction de transfert rationnelle de degré M , on a

$$y(n) = H(z) u(n) = \frac{B(z)}{A(z)} u(n)$$

ce qui donne $A(z) y'(n) - B(z) u'(n) = 0$, car

$$\begin{aligned} A(z) y'(n) - B(z) u'(n) &= A(z) \frac{1}{\widehat{A}(z)} y(n) - B(z) \frac{1}{\widehat{A}(z)} u(n) \\ &= \frac{1}{\widehat{A}(z)} \left(A(z) \frac{B(z)}{A(z)} - B(z) \right) u(n) = 0. \end{aligned}$$

En développant $A(z) y'(n) - B(z) u'(n) = 0$, nous avons

$$0 = y'(n) + \sum_{k=1}^M a_k y'(n-k) - \sum_{k=0}^M b_k u'(n-k),$$

et en soustrayant cette équation de (5.20), nous obtenons

$$e(n) = \sum_{k=1}^M (\widehat{a}_k - a_k) y'(n-k) - \sum_{k=0}^M (\widehat{b}_k - b_k) u'(n-k).$$

Cette dernière forme pour $e(n)$ permet de réécrire l'équation différentielle [5.19] sous la forme

$$\frac{d\mathbf{w}}{dt} = \mathbf{E} \left\{ \underbrace{\begin{bmatrix} -y'(n-1) \\ \vdots \\ -y'(n-M) \\ u'(n) \\ \vdots \\ u'(n-M) \end{bmatrix} \begin{bmatrix} -y'(n-1) & \cdots & -y'(n-M) & u'(n) & \cdots & u'(n-M) \end{bmatrix}}_{\mathbf{R}(\mathbf{w})} \right\}$$

$$\times \underbrace{\begin{bmatrix} a_1 - \hat{a}_1 \\ \vdots \\ a_M - \hat{a}_M \\ b_0 - \hat{b}_0 \\ \vdots \\ b_M - \hat{b}_M \end{bmatrix}}_{\mathbf{w}_* - \mathbf{w}}$$

soit

$$\frac{d\mathbf{w}}{dt} = \mathbf{R}(\mathbf{w}) (\mathbf{w}_* - \mathbf{w}).$$

Notons que $\mathbf{R}(\mathbf{w})$ est une matrice de covariance qui dépend de \mathbf{w} , mais elle est symétrique et définie positive pour tout \mathbf{w} pour lequel le polynôme $\hat{A}(z)$ est à minimum de phase. Si nous choisissons maintenant comme fonction de Lyapunov

$$L(\mathbf{w}) = \frac{1}{2} \|\mathbf{w} - \mathbf{w}_*\|^2$$

sa dérivée temporelle devient

$$\begin{aligned} \frac{dL}{dt} &= \left[\frac{dL}{d\mathbf{w}} \right]^t \frac{d\mathbf{w}}{dt} \\ &= -(\mathbf{w}_* - \mathbf{w})^t \mathbf{R}(\mathbf{w}) (\mathbf{w}_* - \mathbf{w}) \begin{cases} < 0, & \mathbf{w} \neq \mathbf{w}_* ; \\ = 0, & \mathbf{w} = \mathbf{w}_* ; \end{cases} \end{aligned}$$

car $\mathbf{R}(\mathbf{w})$ est définie positive. Ceci montre que $\mathbf{w}(t)$ tend globalement vers \mathbf{w}_* lorsque $t \rightarrow \infty$, et donc que l'équation différentielle associée est globalement convergente.

Notons qu'à la différence de la méthode de gradient, ce résultat de convergence ne s'applique que si l'ordre M du filtre adaptatif coïncide avec l'ordre du système à

identifier. Dans le cas sous-modélisé [$M < d^\circ H(z)$], la convergence de l'équation différentielle s'avère difficile à établir. L'obstacle principal provient du fait que le terme qui pilote l'équation différentielle, en l'occurrence $E[\psi(n) e(n)]$, ne s'interprète pas comme le vecteur gradient d'une fonction de coût. L'analogie de la recherche d'un point minimum ne s'applique pas donc à la méthode de Steiglitz-McBride dans le cas général.

Les expériences de simulation menées dans différentes études (e.g., [FAN 86], [FAN 90], [FAN 93], [REG 95]) attestent toutefois d'un comportement favorable dans le cas sous-modélisé : l'algorithme observé converge souvent vers un voisinage du minimum global de la variance de l'erreur de sortie, même dans les cas où ce critère présente des minima locaux.

Une question préalable à la convergence dans le cas sous-modélisé concerne l'existence même des points stationnaires de l'algorithme ; ceux-ci sont obtenus en résolvant l'équation

$$E[e(n) \psi(n) | \hat{\mathbf{a}}, \hat{\mathbf{b}}] = \mathbf{0},$$

qui consiste à trouver les coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}$ qui annulent la valeur moyenne du terme de réactualisation des paramètres, si elles existent. Dans la mesure où ce système est non linéaire vis-à-vis des coefficients $\hat{\mathbf{a}}$, des solutions sous forme analytique sont difficiles à obtenir. L'article [REG 97] démontre que, au moins dans le cas d'une entrée $\{u(n)\}$ blanche, ce système admet toujours une solution pour laquelle le polynôme $\hat{A}(z) = 1 + \sum_{k=1}^M \hat{a}_k z^{-k}$ est à minimum de phase, correspondant à une fonction de transfert $\hat{H}(z) = \hat{B}(z)/\hat{A}(z)$ stable.

5.8.3. SHARF

Pour un pas d'adaptation μ choisi suffisamment petit, correspondant à l'adaptation lente, l'algorithme HARP se simplifie en l'algorithme SHARF (voir paragraphe 5.7), que nous examinons ici dans le cas où l'ordre M choisi pour le filtre adaptatif coïncide avec l'ordre du système à identifier.

Le vecteur régresseur paraît maintenant comme

$$\psi(n) = \begin{bmatrix} -\hat{y}(n-1) \\ \vdots \\ -\hat{y}(n-M) \\ u(n) \\ \vdots \\ u(n-M) \end{bmatrix}.$$

et donc l'équation différentielle associée prend la forme

$$\frac{d\mathbf{w}}{dt} = \mathbf{E}[\psi(n) \epsilon(n)]$$

Or, lors du paragraphe 5.6 nous avons écrit $\epsilon(n)$ sous la forme

$$\epsilon(n) = \frac{C(z)}{A(z)} s(n)$$

où $s(n)$ s'écrit

$$s(n) = - \sum_{k=1}^M [a_k - \hat{a}_k] \hat{y}(n-k) + \sum_{k=0}^M [b_k - \hat{b}_k] u(n-k)$$

si les coefficients sont fixes. Ceci permet d'écrire l'équation différentielle sous la forme

$$\frac{d\mathbf{w}}{dt} = \mathbf{E} \left\{ \underbrace{\begin{bmatrix} -\hat{y}(n-1) \\ \vdots \\ -\hat{y}(n-M) \\ u(n) \\ \vdots \\ u(n-M) \end{bmatrix} \cdot \left(\frac{C(z)}{A(z)} [-\hat{y}(n) \quad \cdots \quad -\hat{y}(n-M) \quad u(n) \quad \cdots \quad u(n-M)] \right)}_{\mathbf{R}(\mathbf{w})} \right\}$$

$$\times \underbrace{\begin{bmatrix} a_1 - \hat{a}_1 \\ \vdots \\ a_M - \hat{a}_M \\ b_0 - \hat{b}_0 \\ \vdots \\ b_M - \hat{b}_M \end{bmatrix}}_{\mathbf{w}_* - \mathbf{w}}$$

ou bien

$$\frac{d\mathbf{w}}{dt} = \mathbf{R}(\mathbf{w}) (\mathbf{w}_* - \mathbf{w}).$$

Notons que la matrice $\mathbf{R}(\mathbf{w})$ associée à SHARF n'est pas symétrique, en raison de la présence de la fonction de transfert $C(z)/A(z)$ appliquée à chaque argument à droite dans l'opérateur d'espérance. Le lemme suivant montre l'intérêt d'avoir $C(z)/A(z)$ comme fonction strictement réelle positive :

Lemme 5.1 *Supposons que $d^\circ \hat{H}(z) = M$ (aucune compensation pôle-zéro). Si la fonction $C(z)/A(z)$ est strictement réelle positive, la matrice $\mathbf{R}(\mathbf{w}) + \mathbf{R}^t(\mathbf{w})$ est définie positive.*

Pour une démonstration, voir [LJU 77], [REG 95, p. 512]. Grâce à ce lemme, nous pouvons maintenant choisir comme fonction de Lyapunov

$$L(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}_* - \mathbf{w}\|^2.$$

Sa dérivée temporelle devient

$$\begin{aligned} \frac{dL}{dt} &= \left[\frac{dL}{d\mathbf{w}} \right]^t \frac{d\mathbf{w}}{dt} \\ &= -(\mathbf{w}_* - \mathbf{w})^t \mathbf{R}(\mathbf{w}) (\mathbf{w}_* - \mathbf{w}) \\ &= -\frac{1}{2} (\mathbf{w}_* - \mathbf{w})^t [\mathbf{R}(\mathbf{w}) + \mathbf{R}^t(\mathbf{w})] (\mathbf{w}_* - \mathbf{w}) \begin{cases} < 0, & \mathbf{w} \neq \mathbf{w}_* ; \\ = 0, & \mathbf{w} = \mathbf{w}_* . \end{cases} \end{aligned}$$

On en déduit que $\mathbf{w}(t)$ converge globalement vers \mathbf{w}_* lorsque $t \rightarrow \infty$. Ce résultat de convergence ne s'applique que si l'ordre du filtre adaptatif est choisi correctement; dans le cas sous-modélisé, l'algorithme ne suit pas le gradient d'une fonction de coût, et donc l'analogie de la recherche d'un point minimum ne s'applique pas en général à cet algorithme.

L'existence des points stationnaires dans le cas sous-modélisé amène vers l'étude des solutions de l'équation

$$E[e(n) \psi(n) | \hat{\mathbf{a}}, \hat{\mathbf{b}}] = \mathbf{0},$$

qui consiste à chercher des coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}$ qui annulent la valeur moyenne du terme de réactualisation des coefficients. Dans l'article [REG 99], nous avons démontré que ce système admet toujours une solution pour laquelle le polynôme $\hat{A}(z) = 1 + \sum_{k=1}^M \hat{a}_k z^{-k}$ est à minimum de phase, moyennant une contrainte raisonnable que la densité spectrale de puissance du signal d'entrée soit bornée et non nulle pour toute fréquence.

Soit $\hat{A}_*(z)$ le polynôme obtenu en un point stationnaire. Selon les études menées dans [JOH 81], [AND 82], si le filtre de compensation $C(z)$ est choisi pour assurer que la fonction $C(z)/\hat{A}_*(z)$ soit strictement réelle positive, l'algorithme devrait converger vers un voisinage du point stationnaire en question, au moins pour l'adaptation lente.

5.9. Approximation dans le Cas Sous-Modélisé

Dans le cas sous-modélisé, l'erreur de sortie résiduelle sera non nulle quel que soit le réglage des coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}$ du filtre $\hat{H}(z)$; la notion d'identification au sens

littéral est donc inapplicable. Il est légitime, toutefois, d'examiner si, à un point de convergence, le modèle $\hat{H}(z)$ fournit une approximation adéquate du système inconnu $H(z)$. Nous présentons dans ce paragraphe quelques résultats dans ce sens, applicables lorsque l'entrée est un bruit blanc.

Revenons à la fonction de transfert

$$H(z) = \sum_{k=0}^{\infty} h_k z^{-k}.$$

A partir de sa réponse impulsionnelle $\{h_k\}$, on introduit la forme de Hankel Γ_H , qui est une matrice infinie et constante le long de chaque anti-diagonale :

$$\Gamma_H = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots \\ h_2 & h_3 & h_4 & \cdots \\ h_3 & h_4 & h_5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Notons que cette matrice exclut le terme h_0 . La norme euclidienne induite

$$\|\Gamma_H\| \triangleq \sup_{\|\mathbf{x}\|=1} \|\Gamma_H \mathbf{x}\|$$

donne, par définition, la *norme de Hankel* de $H(z)$. On montre [GLO 84], [GEN 81] que cette norme se situe entre les normes L_2 et L_∞ :

$$\left(\sum_{k=1}^{\infty} h_k^2 \right)^{1/2} \leq \|\Gamma_H\| \leq \sup_{\omega} \left| \sum_{k=1}^{\infty} h_k e^{j\omega k} \right| \quad (5.21)$$

où nous notons que le terme h_0 est toujours exclu. Selon le théorème de Kronecker [ADA 71], la matrice infinie de Hankel Γ_H sera de rang fini, disons N , si et seulement si $H(z)$ est une fraction rationnelle de degré N . Pour les systèmes physiques rencontrés dans les applications, on a souvent de mal à déduire un rang fini pour la matrice de Hankel ; voir, pour exemple, [LIA 98] pour une étude des degrés des chemins d'écho acoustique.

Le modèle ajustable $\hat{H}(z)$ est contraint d'avoir un degré M , et donc sa forme de Hankel aura un rang égal à M :

$$\text{rang} \underbrace{\begin{bmatrix} \hat{h}_1 & \hat{h}_2 & \hat{h}_3 & \cdots \\ \hat{h}_2 & \hat{h}_3 & \hat{h}_4 & \cdots \\ \hat{h}_3 & \hat{h}_4 & \hat{h}_5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{\Gamma_{\hat{H}}} = M$$

Le problème d'approximation par norme de Hankel consiste, à partir d'une matrice de Hankel Γ_H de rang arbitraire (mais supposé "grand"), à trouver une matrice de Hankel $\Gamma_{\hat{H}}$ de rang inférieur qui est la plus proche par norme matricielle induite :

$$\min_{\text{rang } \Gamma_{\hat{H}} = M} \|\Gamma_H - \Gamma_{\hat{H}}\|$$

Si $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq 0$ désignent les valeurs singulières de Γ_H , un résultat classique en analyse matricielle [HOR 88] montre que σ_{M+1} est la distance entre Γ_H et l'ensemble de matrices de rang M . En plus, une approximation optimale de rang M peut être construite en tronquant la décomposition en valeurs singulières de Γ_H à ses M premiers termes. Une telle approximation, toutefois, ne donne pas une matrice de Hankel en général, et donc ne donne pas une solution à notre problème car $\Gamma_{\hat{H}}$ est contrainte d'être elle aussi une matrice de Hankel.

Adamjan, Arov et Kreĭn [ADA 71] ont toutefois démontré un résultat remarquable : une matrice de Hankel infinie admet toujours une approximation optimale qui garde sa structure de Hankel. On aura donc :

$$\min_{\text{rang } \Gamma_{\hat{H}} \leq M} \|\Gamma_H - \Gamma_{\hat{H}}\| = \sigma_{M+1}.$$

En plus, il existe une matrice de Hankel unique $\Gamma_{\hat{H}}$, de rang ne dépassant pas M , qui atteint la borne.

Par l'inégalité entre la norme L_2 et la norme de Hankel (5.21), on a

$$\left(\sum_{k=1}^{\infty} (h_k - \hat{h}_k)^2 \right)^{1/2} \leq \|\Gamma_H - \Gamma_{\hat{H}}\|$$

Puisque le membre à droite admet σ_{M+1} comme valeur minimum, on en déduit une borne applicable à l'approximation par norme L_2 :

$$\min_{d^\circ \hat{H}(z) = M} \|H(z) - \hat{H}(z)\|_2 = \min_{d^\circ \hat{H}(z) = M} \left(\sum_{k=1}^{\infty} (h_k - \hat{h}_k)^2 \right)^{1/2} \leq \sigma_{M+1} \quad (5.22)$$

(en veillant à choisir $\hat{h}_0 = h_0$, bien entendu). On peut aussi montrer une borne applicable à la norme L_∞ [GLO 84] :

$$\min_{d^\circ \hat{H}(z) = M} \|H(z) - \hat{H}(z)\|_\infty \leq \sigma_{M+1} + \sigma_{M+2} + \sigma_{M+3} + \dots$$

Ces bornes prouvent que si σ_{M+1} est "petite" alors $H(z)$ est "proche" d'une fonction rationnelle de degré M , par norme de Hankel, norme L_2 , ou norme L_∞ .

Considérons maintenant l'algorithme de gradient en erreur de sortie. Si les coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}$ sont fixes et l'entrée $\{u(n)\}$ est blanche, la fonction de coût s'exprime comme

$$\begin{aligned}
 E[e^2(n)] &= E\{(d(n) - \hat{y}(n))^2\} \\
 &= E\left[\left(v(n) + \sum_{k=0}^{\infty} (h_k - \hat{h}_k) u(n-k)\right)^2\right] \\
 &= E[v^2(n)] + E[u^2(n)] \underbrace{\sum_{k=0}^{\infty} (h_k - \hat{h}_k)^2}_{\|H(z) - \hat{H}(z)\|_2^2} \\
 &\quad + \sum_{k=0}^{\infty} (h_k - \hat{h}_k) \underbrace{E[v(n) u(n-k)]}_0 \\
 &= E[v^2(n)] + E[u^2(n)] \cdot \|H(z) - \hat{H}(z)\|_2^2
 \end{aligned}$$

car le bruit de sortie $\{v(n)\}$ est supposé indépendant de l'entrée $\{u(n)\}$. En tenant compte de la borne (5.22) applicable au minimum global du critère à norme L_2 , nous déduisons qu'au point minimum global,

$$\frac{E[e^2(n)]}{E[u^2(n)]} \leq \frac{E[v^2(n)]}{E[u^2(n)]} + \sigma_{M+1}^2.$$

Ainsi, si σ_{m+1} est petit, le modèle $\hat{H}(z)$ fourni par la méthode de gradient au minimum global est garanti d'être une bonne approximation du système inconnu. Notons toutefois que d'éventuels minima locaux risquent de fournir des modèles donnant une erreur résiduelle importante, ce qui est un inconvénient majeur de la méthode de gradient.

Pour la méthode de Steiglitz-McBride, si l'entrée $\{u(n)\}$ est blanche, en chaque point stationnaire on a la borne suivante [REG 96] :

$$\|H(z) - \hat{H}(z)\|_2^2 \leq \sigma_{M+1}^2 + \frac{\max_{\omega} S_v(e^{j\omega}) - E[v^2(n)]}{E[u^2(n)]}$$

où $S_v(e^{j\omega})$ est la densité spectrale de puissance du bruit de sortie. Si le bruit de sortie est blanc, on a $S_v(e^{j\omega}) = E[v^2(n)]$ pour tout ω , et la borne se simplifie en

$$\|H(z) - \hat{H}(z)\|_2 \leq \sigma_{M+1}.$$

Ainsi, si σ_{M+1} est petit, un point stationnaire de l'algorithme est garanti de fournir une bonne approximation du système inconnu. Notons que la méthode de Steiglitz-McBride peut, dans certains cas, admettre des points stationnaires multiples [FAN 90]. La non unicité, pourtant, n'est pas gênante, en ce sens que la borne ci-dessus s'applique à chaque point stationnaire.

Soit maintenant $\hat{H}_{\min}(z)$ la fonction qui minimise globalement la variance de l'erreur de sortie, et soit $\hat{H}_{SM}(z)$ la fonction de transfert obtenue à un point stationnaire de l'algorithme de Steiglitz-McBride. Puisque $\|H(z) - \hat{H}_{\min}(z)\|_2 \leq \sigma_{M+1}$ et $\|H(z) - \hat{H}_{SM}(z)\|_2 \leq \sigma_{M+1}$, nous obtenons, par l'inégalité triangulaire,

$$\begin{aligned} \|\hat{H}_{\min}(z) - \hat{H}_{SM}(z)\|_2 &= \|[H(z) - \hat{H}_{SM}(z)] + [H(z) - \hat{H}_{\min}(z)]\|_2 \\ &\leq \|H(z) - \hat{H}_{SM}(z)\|_2 + \|H(z) - \hat{H}_{\min}(z)\|_2 \\ &\leq 2\sigma_{M+1}. \end{aligned}$$

Cette inégalité prouve que, lorsque σ_{M+1} est petit, un point stationnaire de l'algorithme de Steiglitz-McBride sera proche du minimum global de la variance de l'erreur de sortie.

5.10. Conclusions

Nous avons présenté les algorithmes principaux pour le filtrage RII adaptatif, ainsi que divers résultats concernant les propriétés de convergence et de modélisation. Par souci de simplicité, nous avons examiné seulement la réalisation du filtre adaptatif en forme directe. Cette forme de réalisation présente un inconvénient majeur : sa stabilité dans les environnements non stationnaires est difficile à assurer. Cette considération favorise le développement d'algorithmes pour d'autres formes de réalisation, notamment la forme en treillis qui a le mérite de rester stable pendant l'adaptation [REG 95]. Le lecteur qui souhaite étudier en plus de profondeur ces autres formes d'algorithmes pourrait consulter les références [JOH 84], [SHY 89], [REG 92], [MIA 94], [REG 95], [LOP 01].

5.11. Bibliographie

- [ADA 71] ADAMJAN V. M., AROV D. Z., KREĬN M. G., « Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem », *Math. USSR Sbornik*, vol. 15, p. 31–73, 1971.
- [AND 68] ANDERSON B. D. O., « A simplified viewpoint of hyperstability », *IEEE Trans. Automatic Control*, vol. 13, p. 292–294, juin 1968.

- [AND 82] ANDERSON B. D. O., JOHNSON JR. C. R., « On reduced-order adaptive output error identification and adaptive IIR filtering », *IEEE Trans. Automatic Control*, vol. 27, p. 927–933, août 1982.
- [BEN 87] BENVENISTE A., METIVIER M., PRIOURET P., *Algorithmes Adaptatifs et Approximations Stochastiques*, Masson, Paris, 1987.
- [FAN 86] FAN H., JENKINS W. K., « A new adaptive IIR filter », *IEEE Trans. Circuits and Systems*, vol. 33, p. 939–947, 1986.
- [FAN 88] FAN H., « Application of Benveniste's convergence results in the study of adaptive IIR filtering algorithms », *IEEE Trans. Information Theory*, vol. 34, p. 692–709, juillet 1988.
- [FAN 90] FAN H., NAYERI M., « On reduced order identification : Revisiting "On some system identification techniques for adaptive filtering" », *IEEE Trans. Circuits and Systems*, vol. 37, p. 1144–1151, septembre 1990.
- [FAN 93] FAN H., DOROSLOVAČKI D., « On "global convergence" of Steiglitz-McBride adaptive algorithm », *IEEE Trans. Circuits and Systems II*, vol. 40, p. 73–87, février 1993.
- [GEN 81] GENIN Y., KUNG S. Y., « A two-variable approach to the model reduction problem with Hankel norm criterion », *IEEE Trans. Circuits and Systems*, vol. 28, p. 912–924, septembre 1981.
- [GLO 84] GLOVER K., « All optimal Hankel-norm approximations to linear multivariable systems and their L^∞ error bounds », *Int. J. Control*, vol. 39, p. 1115–1193, 1984.
- [HOR 88] HORN R. A., JOHNSON C. R., *Matrix Analysis*, Cambridge University Press, Cambridge, 1988.
- [JOH 79] JOHNSON JR. C. R., « A convergence proof for a hyperstable adaptive recursive filter », *IEEE Trans. Information Theory*, vol. 25, p. 745–749, novembre 1979.
- [JOH 81] JOHNSON JR. C. R., ANDERSON B. D. O., « Sufficient excitation and stable reduced-order adaptive IIR filtering », *IEEE Trans. Acoustic, Speech, and Signal Processing*, vol. 29, p. 1212–1215, décembre 1981.
- [JOH 84] JOHNSON JR. C. R., « Adaptive IIR filtering : Current results and open issues », *IEEE Trans. Information Theory*, vol. 30, p. 237–250, mars 1984.
- [KUS 84] KUSHNER H. J., SHWARTZ A., « Weak convergence and asymptotic properties of adaptive filters with constant gains », *IEEE Trans. Information Theory*, vol. 30, n°2, mars 1984.
- [LAN 76] LANDAU I. D., « Unbiased recursive identification using model reference adaptive techniques », *IEEE Trans. Automatic Control*, vol. 21, p. 194–202, avril 1976.
- [LAR 80] LARIMORE M. G., TREICHLER J. R., JOHNSON JR. C. R., « SHARF : An algorithm for adapting IIR digital filters », *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 28, p. 428–440, août 1980.
- [LIA 98] LIAVAS A., REGALIA P. A., « Acoustic echo cancellation : Do IIR models offer better modeling capabilities than their FIR counterparts ? », *IEEE Trans. Signal Processing*, vol. 46, p. 2499–2504, septembre 1998.

- [LJU 77] LJUNG L., « On positive real transfer functions and the convergence of some recursive schemes », *IEEE Trans. Automatic Control*, vol. 22, p. 539–551, août 1977.
- [LOP 99] LOPEZ-VALCARCE R., DASGUPTA S., « A new proof for the stability of equation-error models », *IEEE Signal Processing Letters*, vol. 6, p. 148–150, juin 1999.
- [LOP 01] LOPEZ-VALCARCE R., PEREZ-GONZALEZ F., « Adaptive lattice IIR filtering revisited : Convergence issues and new algorithms with improved stability properties », *IEEE Trans. Signal Processing*, vol. 49, p. 811–821, avril 2001.
- [MIA 94] MIAO K. X., FAN H., DOROSLOVAČKI M., « Cascade normalized adaptive IIR filters », *IEEE Trans. Signal Processing*, vol. 42, p. 721–742, avril 1994.
- [MUL 76] MULLIS C. T., ROBERTS R. A., « The use of second-order information in the approximation of discrete-time linear systems », *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 24, p. 226–238, juin 1976.
- [POP 63] POPOV V. M., « The solution of a new stability problem for controlled systems », *Automation and Remote Control*, vol. 24, p. 1–23, janvier 1963.
- [REG 92] REGALIA P. A., « Stable and efficient lattice algorithms for adaptive IIR filtering », *IEEE Trans. Signal Processing*, vol. 40, p. 375–388, février 1992.
- [REG 94] REGALIA P. A., « An unbiased equation error identifier and reduced-order approximations », *IEEE Trans. Signal Processing*, vol. 42, p. 1397–1412, juin 1994.
- [REG 95] REGALIA P. A., *Adaptive IIR Filtering in Signal Processing and Control*, Marcel Dekker, New York, 1995.
- [REG 96] REGALIA P. A., MBOUP M., « Undermodeled adaptive filtering : An a priori error bound for the Steiglitz-McBride method », *IEEE Trans. Circuits and Systems*, vol. 43, p. 105–116, février 1996.
- [REG 97] REGALIA P. A., MBOUP M., ASHARI M., « On the existence of stationary points for the Steiglitz-McBride algorithm », *IEEE Trans. Automatic Control*, vol. 42, p. 1592–1596, novembre 1997.
- [REG 99] REGALIA P. A., MBOUP M., ASHARI M., « Existence of stationary points for pseudo-linear regression algorithms », *IEEE Trans. Automatic Control*, vol. 44, p. 994–998, mai 1999.
- [SHY 89] SHYNK J. J., « Adaptive IIR filtering », *IEEE ASSP Magazine*, avril 1989.
- [SÖD 75] SÖDERSTRÖM T., « On the uniqueness of maximum likelihood identification », *Automatica*, vol. 11, p. 193–197, 1975.
- [SÖD 81] SÖDERSTRÖM T., STOICA P., « On the stability of dynamic models obtained by least-squares identification », *IEEE Trans. Automatic Control*, vol. 26, p. 575–577, avril 1981.
- [SÖD 82] SÖDERSTRÖM T., STOICA P., « Some properties of the output error method », *Automatica*, vol. 18, n° 1, p. 93–99, 1982.
- [STE 65] STEIGLITZ K., MCBRIDE L. E., « A technique for the identification of linear systems », *IEEE Trans. Automatic Control*, vol. 10, p. 461–464, octobre 1965.

- [STO 81] STOICA P., SÖDERSTRÖM T., « The Steiglitz-McBride algorithm revisited—Convergence analysis and accuracy aspects », *IEEE Trans. Automatic Control*, vol. 26, p. 712–717, juin 1981.

Index

- ALE (Adaptive Line Enhancer), 45
- Algorithme adaptatif, 33, 107
- Algorithme déterministe moyen, 109
- Algorithme projection affine (APA), 73
- Algorithme rapide
 - en treillis (MCRT), 179
 - en treillis normalisé, 183
 - MCR 2D-1D, 170
 - MCR-QR, 196
 - MCR1 (transverse), 158
 - MCR2 (transverse), 162
- Annulation d'écho, 65
- Approximation
 - de sous-espaces, 103
 - gaussienne (diffusion), 121
 - rationnelle (Hankel), 233
- Approximation stochastique, 32, 36, 45, 106
- ARMA (modélisation), 101
- Attracteur, 112
 - cycle limite, 117
- Birkhoff (Théorème ergodique), 39
- Chaîne de Markov, 108
- Champ de vecteurs (d'un algorithme), 37, 108
- Champ moyen, 109
- Coercivité (condition de), 115
- Constante de temps
 - du Gradient déterministe, 31
 - du RLS, 66
- Contrainte
 - (Algorithme avec), 112
 - monique, 204
 - Norme unité, 204
- Contrainte (Algorithme avec), 35
- Convergence
 - de l'APA, 79
 - en loi des processus aléatoires, 119
 - en m.q. du NLMS, 56
 - p.s. (Algos étendus), 113
 - p.s. de ELS, 117
 - p.s. du LMS, 47
 - p.s. du RLS, 64
 - superlinéaire du RLS, 66
- Courbe d'apprentissage, 31
- Cramer-Rao (borne de), 21, 37, 69
- Diffusion gaussienne, 120
- Echelles de temps (en poursuite), 129
- Efficacité (d'un estimateur), 37
- Egalisation, 18, 35
 - aveugle, 101
- Egaliseur
 - Algorithme de Sato, 102
- ELS, 107, 116
- Equation différentielle
 - stochastique (EDS), 121
- Equations normales, 27, 29
 - empiriques, 29
 - pondérées, 63
- Ergodique (processus, signal), 39
- Ergodisme, 32, 38, 48, 64
 - de la moyenne, 38
 - du second ordre, 38
- Erreur

- d'ajustement du LMS, 50
- d'ajustement du NLMS, 56
- d'équation, 200, 214
- d'équation (stabilité), 207
- de sortie (méthode), 209
- quadratique du RLS, 67
- quadratique moyenne résiduelle (EQMR), 51
- Erreur quadratique moyenne (EQM), 20, 43
- Espace
 - d'observation, 25, 28
- Estimateur, 63
 - de moyenne, 110
- Estimateur (propriétés statistiques), 37
- Estimation
 - bayésienne, 21
 - d'une moyenne, 34
 - en m.q., 23
 - espérance conditionnelle, 23
 - linéaire en m.q. (ELMQ), 23, 24
 - optimale, 20
 - paramétrique, 21
 - statistique, 36
- Estimation de retards, 103
- Etat
 - à représentation markovienne, 130
- Excitation persistante, 48, 202, 206
- Facteur d'oubli, 62
- Filtrage
 - de Kalman, 24
 - de Wiener, 25
 - de Wiener RIF, 26
 - de Wiener RIF (transverse), 43
 - en treillis, 81
 - RII, 100
- Fletcher-Reeves (Algorithme), 33
- Fluctuation
 - approximation gaussienne, 120
 - asymptotique, 118
 - du LMS, 47
 - résiduelle du LMS, 49
 - résiduelle du RLS, 66
- Forme de Hankel, 232
- Gain
 - matriciel, 132
- Gain matriciel, 32, 129
- Gradient, 105
- Gradient (Algorithme de)
 - conjugué, 33
 - stochastique (LMS), 36
 - déterministe, 29
 - Erreur de sortie, 212
 - forme bloc-exacte du LMS, 58
 - stochastique, 205
 - stochastique (LMS), 43
 - stochastique (LMS-RII), 106
 - stochastique normalisé (NLMS), 54
- HARF (Algorithme), 220
- Hilbert (espace de), 22
- Hypermodèle, 129
 - Dérive déterministe, 130
 - Marche aléatoire, 130
- Hyperstabilité, 217
 - Théorème de Popov, 217
- Identification, 19, 200
- Indépendance asymptotique, 40
- Information de Fisher (matrice), 69
- Kushner
 - Théorème de convergence p.s., 113
- Lemme d'inversion matricielle, 61, 149
- Liapounoff
 - Equation de, 121
 - fonction de, 112, 114, 225, 228, 231
- LMS, 45
- LMS-RII, 106, 116
- Loi des grands nombres, 38
- M-indépendance, 40
- Mélange (propriété de, ergodisme), 40
- Maximum
 - de vraisemblance, 21
 - de probabilité *a posteriori* (MAP), 21
 - de vraisemblance, 68, 143, 209
 - de vraisemblance récursif (RML), 106
- Minima locaux, 36, 100, 210
- Modélisation RII, 105
- Modélisation
 - ARMA, 100
- Moindres carrés, 27
 - étendus (ELS), 107
 - lien avec l'APA, 76
 - récursifs, 60
- Mouvement brownien, 121
- Newton (Algorithme de), 31, 32

- NLMS, 54
 Non linéaire (estimation), 20
 Non-stationnarité
 Degré de, 125
 Norme
 de Hankel, 232
 ODE, 107
 Méthode de l', 111, 224
 Poursuite
 d'une régression variable, 124
 Préfiltrage, 213
 Projection orthogonale, 22
 Puissance itérée (méthode de), 134
 QR (décomposition), 186
 Quasi-Newton
 (Algorithme de), 32, 106
 stochastique, 61
 Quotient de Rayleigh, 137
 Régression linéaire, 64, 68
 Régularisation, 77
 Regression
 linéaire variable, 124
 Représentation Markovienne, 108
 RLS, 61
 à oubli λ , 63
 déterministe, 65
 RML, 106
 Schur
 complément de, 202
 SHARF (Algorithme), 222
 Sous-gaussienne (loi), 102
 Sous-modélisation, 207, 229, 231
 Soustraction de bruit, 18
 SPR (Condition), 115, 116, 217, 230
 Stabilité
 en m.q. du LMS, 51
 exponentielle du LMS, 49
 système non linéaire, 217
 Stabilité, 108
 au sens de Liapounoff, 114
 de l'ODE, 113
 exponentielle des transitoires, 112
 globale d'un attracteur, 115
 Steiglitz-McBride
 méthode, 213
 Algorithme, 216
 Convergence, 226
 Théorème
 de Gauss-Markov (RLS), 68
 de Limite Centrale, 37, 69, 119, 121
 de projection, 22
 Toeplitz (matrice de), 147
 Transitoire
 du LMS, 47
 du RLS, 65
 Treillis, 169
 adaptatif d'un filtre de Wiener RIF, 84
 gradient et variantes, 83
 LMS, 84
 Moindres Carrés, 85
 Moindres Carrés (Burg), 86
 Prédicteur en, 172
 prédiction linéaire, 81
 Unimodalité, 101
 Vecteur d'état, 37, 107
 Vraisemblance (variable de), 151
 Wiener-Lévy (processus de), 121

Fiche pour le service de fabrication

5.12. *

Auteurs :

sous la direction de
François MICHAUT
Maurice BELLANGER

5.13. *

Titre du livre :

Filtrage adaptatif : théorie et applications
Volume 1
théorie et algorithmes

5.14. *

Titre abrégé :

Filtrage adaptatif

5.15. *

Date de cette version :

28 juin 2005

5.16. *

Contact :

- téléphone : 04 92 94 27 48
- télécopie : 04 92 94 28 96
- Mél : rr@unice.fr

5.17. *

Logiciel pour la composition :

- L^AT_EX, avec la classe ouvrage-hermes.cls,
- version 1.3, 17/09/2001.
- traité (option treatise) : Oui (*chapitres avec différents auteurs*)
- livre en anglais (option english) : Non (*par défaut en français*)
- tracé des limites de page (option cropmarks) : Non (*par défaut*)
- suppression des en-têtes de page (option empty) : Non (*par défaut*)
- impression des pages blanches (option allpages) : Non (*par défaut*)
- césures actives : voir la coupure du mot signal dans le fichier .log