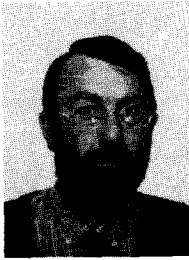


Équations de Chandrasekhar et algorithmes rapides pour le traitement du signal et des images

Chandrasekhar equations and fast algorithms
in signal and image processing



Guy DEMOMENT

Laboratoire des Signaux et Systèmes (CNRS/ESE/UPS), École Supérieure d'Électricité, plateau de Moulon, 91192 GIF-SUR-YVETTE CEDEX, et Gréco SARTA.

Guy Demoment est ingénieur ESE, Docteur-Ingénieur, Docteur ès Sciences. Chercheur au CNRS et professeur de probabilités et théorie du signal à l'École Supérieure d'Électricité. Il est également responsable du groupe « Problèmes inverses en traitement du signal et imagerie » au Laboratoire des Signaux et Systèmes (CNRS/ESE/UPS).

RÉSUMÉ

Les équations de Chandrasekhar ont été introduites en automatique au début des années 1970. Leur apparition a marqué le début d'une activité importante dans le domaine des algorithmes rapides de la famille des moindres carrés. Elles sont pourtant assez peu utilisées. Ceci est compréhensible en automatique où la rapidité n'est pas une exigence essentielle. Mais cette désaffection s'explique moins bien en traitement du signal. Peut-être vient-elle de ce que leur emploi semble lié à une approche interne, ou par modèle d'état, des problèmes d'estimation qui est moins fréquente en traitement du signal.

Le but de cet article est de faire le point sur l'intérêt de ces équations et sur l'utilisation de la famille d'algorithmes qui en découle pour la résolution de problèmes généraux d'estimation linéaire. On tentera de corriger, dans une certaine mesure, l'impression déjà évoquée en montrant les liens que ces équations possèdent avec d'autres familles d'algorithmes plus familières, et en mettant en évidence le caractère quasi canonique qu'elles peuvent présenter dans certaines applications.

MOTS CLÉS

Estimation linéaire, algorithmes rapides, équations de Chandrasekhar.

SUMMARY

Chandrasekhar equations were introduced in automatic control in the early 70s. Their appearance marked the beginning of a considerable amount of research on fast least squares algorithms. They are, however, little used. This is understandable in automatic control where speed is not an essential requirement, but is less obvious in signal processing. It may stem from the fact that their use is linked by an internal approach, or by state space model, to estimation problems which occurs less often in signal processing.

The aim of this article is to take stock of the interest of these equations and the use of the family of algorithms which derive from them for solving general problems of linear estimation. We shall try to a certain extent to correct the impression given above by showing the links that these equations have with other, more familiar, families of algorithms, and the quasi-canonic character that they can display in certain applications.

KEY WORDS

Linear estimation, fast algorithms, Chandrasekhar equations.

Avertissement

Cet article résume des travaux effectués en collaboration avec Du-Xué-Cheng, Yves Goussard, Amrane Houacine, Roger Reynaud et Didier Saint-Félix. Il s'agit donc, contrairement aux autres travaux présentés dans ce numéro spécial consacré au Gréco SARTA, du bilan de l'activité d'une seule équipe. Mais il faut préciser que cette activité a largement bénéficié du climat de discussions et d'échanges créé par ce Gréco. Cette synthèse sur les algorithmes rapides du type Chandrasekhar doit beaucoup en particulier à des discussions approfondies avec Claude Guéguen.

1. Introduction

Les 15 dernières années ont été marquées par un développement très important d'algorithmes rapides pour la résolution de problèmes d'estimation linéaire au sens des moindres carrés. Si l'on excepte l'algorithme de Levinson, beaucoup plus ancien, les premiers algorithmes de cette famille sont en effet apparus au début des années 1970, avec les équations de Chandrasekhar en automatique [18, 22]. Ces équations apportaient une solution rapide au problème du filtrage de Kalman lorsque le modèle d'état utilisé possède des propriétés d'invariance. Elles supposaient donc l'emploi d'une représentation « interne » des signaux observés.

Quelques années plus tard sont apparus les algorithmes des moindres carrés récursifs rapides qui ont connu depuis un développement considérable [15]. Ces algorithmes exploitent aussi des propriétés d'invariance, mais leur principale différence avec les précédents provient de ce qu'ils utilisent une représentation « externe » des signaux observés qui conduit à résoudre des équations normales. Cette famille d'algorithmes est extrêmement variée, en raison essentiellement des multiples choix qui peuvent être effectués : structures transverses ou structures en treillis, fenêtrage ou non des signaux, normalisation ou non des variables, etc. Les techniques d'établissement de ces algorithmes sont elles aussi très variées puisqu'elles vont des méthodes algébriques reposant sur les lemmes d'inversion de Schur jusqu'aux méthodes géométriques liées à l'emploi de critères des moindres carrés. Cette très grande variété est d'ailleurs source de problèmes pour l'utilisateur potentiel qui s'en trouve désemparé.

Si l'on examine la littérature récente du traitement du signal et de l'automatique, on ne peut que constater que ces équations de Chandrasekhar sont assez peu utilisées. Ceci est compréhensible en automatique où la rapidité n'est pas une exigence essentielle. Mais cette désaffection s'explique moins bien en traitement du signal. Peut-être vient-elle de ce que leur emploi semble lié à une approche interne, ou par modèle d'état, des problèmes d'estimation qui est moins fréquente en traitement du signal.

Le but de cet article est de faire le point sur l'intérêt de ces équations et sur la possibilité d'utiliser la famille d'algorithmes qui en découle pour la résolution de problèmes généraux d'estimation linéaire. On tentera de corriger, dans une certaine mesure, l'impression déjà évoquée en montrant les liens que ces équations possèdent avec d'autres familles d'algorithmes plus familières, et en mettant en évidence le caractère quasi canonique qu'elles peuvent présenter dans certaines applications.

Cet article est organisé de la manière suivante : après cette introduction, un second chapitre est consacré à des rappels sur les équations de Chandrasekhar. On montre ensuite comment elles peuvent être utilisées pour la résolution de deux problèmes types du traitement du signal : l'estimation adaptative des coefficients d'une régression linéaire et l'estimation d'un signal ou d'une image observés au travers d'un système linéaire. Un chapitre est consacré à chacune de ces deux applications. Enfin la conclusion tente de montrer, au travers d'un exemple, qu'invariance et stationnarité ne sont pas strictement équivalentes et que les équations de Chandrasekhar offrent, de ce point de vue, un potentiel mal exploité.

2. Rappels sur les équations de Chandrasekhar

Les équations de Chandrasekhar ont été introduites, comme les équations de Kalman, pour résoudre le problème du filtrage linéaire lorsque le point de départ est un modèle d'état décrivant le processus d'observation [10]. Ce processus aléatoire à valeurs réelles y_k , $k = 1, 2, \dots$, de dimension p , représente les signaux observés et est supposé décrit par le modèle d'état suivant :

$$(2.1) \quad \mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k$$

$$(2.2) \quad \mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{b}_k$$

où \mathbf{x}_k désigne le vecteur d'état et où \mathbf{F}_k , \mathbf{G}_k et \mathbf{H}_k sont des matrices réelles de dimensions (n, n) , (n, m) et (p, n) et connues à l'instant k . Les hypothèses de départ sur les processus mis en jeu dans ce modèle sont les suivantes :

$$(2.3) \quad E\{\mathbf{x}_1\} = \hat{\mathbf{x}}_{1/0}, \quad E\{\mathbf{b}_k\} = \mathbf{0}, \quad E\{\mathbf{w}_k\} = \mathbf{0}$$

$$(2.4) \quad E\left\{ \begin{bmatrix} \mathbf{b}_k \\ \mathbf{x}_1 - \hat{\mathbf{x}}_{1/0} \\ \mathbf{w}_k \end{bmatrix} [\mathbf{b}_k^t, (\mathbf{x}_1 - \hat{\mathbf{x}}_{1/0})^t, \mathbf{w}_k^t] \right\} = \begin{bmatrix} \mathbf{R}_k \delta_{kn} & \mathbf{0} & \mathbf{S}_k \delta_{kn} \\ \mathbf{0} & \mathbf{P}_{1/0} & \mathbf{0} \\ \mathbf{S}_k^t \delta_{kn} & \mathbf{0} & \mathbf{Q}_k \delta_{kn} \end{bmatrix}$$

Les matrices de covariance $\mathbf{P}_{1/0}$, \mathbf{Q}_k et \mathbf{R}_k sont, par définition, définies symétriques. Le problème du calcul de l'estimateur optimal $\hat{\mathbf{x}}_{k+1/k}$ (i.e. au sens de la variance d'erreur minimale) de l'état à l'instant $k+1$ étant données les observations jusqu'à l'instant k est classiquement résolu par le filtre de Kalman.

2.1. ÉQUATIONS DE KALMAN

Le filtre de Kalman est un filtre numérique récursif non constant. Il existe différentes façons d'établir les équations de ce filtre qui peuvent elles-mêmes s'exprimer sous des formes variées [1]. Ce filtre est maintenant trop connu pour qu'il soit utile de les détailler ici. Une forme fréquemment rencontrée est la forme compacte suivante :

$$(2.5) \quad \mathbf{e}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k/k-1}$$

$$(2.6) \quad \hat{\mathbf{x}}_{k+1/k} = \mathbf{F}_k \hat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k (\mathbf{R}_k^e)^{-1} \mathbf{e}_k$$

où le gain de Kalman \mathbf{K}_k et la matrice de covariance de l'innovation \mathbf{R}_k^e sont calculés par l'intermédiaire d'une matrice (n, n) , la matrice de covariance de l'erreur de prédiction à un pas du vecteur d'état, $\mathbf{P}_{k/k-1}$:

$$(2.7) \quad \mathbf{R}_k^e = \mathbf{H}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^t + \mathbf{R}_k$$

$$(2.8) \quad \mathbf{K}_k = \mathbf{F}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^t + \mathbf{G}_k \mathbf{S}_k$$

qui est elle-même déterminée à l'aide de la relation :

$$(2.9) \quad \mathbf{P}_{k+1/k} = \mathbf{F}_k \mathbf{P}_{k/k-1} \mathbf{F}_k^t + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^t - \mathbf{K}_k (\mathbf{R}_k^e)^{-1} \mathbf{K}_k^t$$

qui devient une équation aux différences de Riccati lorsque le gain et la covariance de l'innovation sont remplacés par leurs expressions (2.7) et (2.8).

Il est remarquable que l'équation de Riccati du filtre de Kalman soit la même, que les paramètres du modèle (\mathbf{F}_k , \mathbf{G}_k , \mathbf{H}_k , \mathbf{Q}_k , \mathbf{R}_k , \mathbf{S}_k) soient constants, ou qu'ils varient au cours du temps. Des simplifications devraient donc intervenir dans le cas constant en raison de la redondance que cette invariance du modèle entraîne. C'est justement l'objet des équations de Chandrasekhar.

2.2. ÉQUATIONS DE CHANDRASEKHAR

On a pu montrer, au début des années 1970, qu'une forme discrète de certaines équations différentielles non linéaires introduites par Chandrasekhar pour résoudre des équations de Wiener-Hopf à noyaux stationnaires pouvait être utilisée pour gagner jusqu'à un ordre de grandeur dans le volume des calculs d'un filtre de Kalman [18, 22]. Par la suite, ces équations ont été établies par différentes techniques : théorie de la propagation dans les réseaux [6], partitionnement [2], algorithme de Levinson généralisé [11]. Il existe, pour un même problème de filtrage, plusieurs formes d'algorithmes de Chandrasekhar dont les avantages respectifs dépendent des valeurs relatives de m , n et p [10]. Il suffit d'en présenter une pour les besoins de l'exposé.

L'idée suivie pour l'établissement de ces équations est très simple. Lorsque le modèle (2.1) à (2.4) est invariant, le processus observé est asymptotiquement stationnaire. Il est donc pénalisant de recalculer à chaque instant d'observation les quantités nominales du filtre \mathbf{K}_k , \mathbf{R}_k^e et $\mathbf{P}_{k/k-1}$, d'autant que cette dernière n'est qu'un auxiliaire dans le calcul du gain du filtre. Il est préférable de s'intéresser à leurs incréments :

$$(2.10) \quad \delta \mathbf{P}_k = \mathbf{P}_{k/k-1} - \mathbf{P}_{k-1/k-2}$$

$$(2.11) \quad \delta \mathbf{K}_k = \mathbf{K}_k - \mathbf{K}_{k-1}$$

$$(2.12) \quad \delta \mathbf{R}_k^e = \mathbf{R}_k^e - \mathbf{R}_{k-1}^e$$

qui possèdent de « bonnes » propriétés. Tout repose en fait sur la déficience du rang de $\delta \mathbf{P}_k$ qui permet de le factoriser sous la forme suivante :

$$(2.13) \quad \delta \mathbf{P}_k = \mathbf{Y}_{k-1} \mathbf{M}_{k-1} \mathbf{Y}_{k-1}^t$$

où \mathbf{M}_k est une matrice carrée symétrique de dimensions (α, α) avec $\alpha \leq n$ et même souvent $\alpha \ll n$. Mais, contrairement à $\mathbf{P}_{k/k-1}$, \mathbf{M}_k n'a pas de caractère défini établi. On montre seulement que sa signature est constante au cours des récurrences [10, 24]. La technique d'obtention des équations de Chandrasekhar est alors la suivante. On effectue les factorisations :

$$(2.14) \quad \delta \mathbf{K}_k^g = (\mathbf{F} - \mathbf{K}_{k-1}^g \mathbf{H}) \delta \mathbf{P}_k \mathbf{H}^t (\mathbf{R}_k^e)^{-1}$$

$$(2.15) \quad \delta \mathbf{P}_{k+1} = (\mathbf{F} - \mathbf{K}_{k-1}^g \mathbf{H}) \times (\delta \mathbf{P}_k - \delta \mathbf{P}_k \mathbf{H}^t (\mathbf{R}_k^e)^{-1} \mathbf{H} \delta \mathbf{P}_k) (\mathbf{F} - \mathbf{K}_{k-1}^g \mathbf{H})^t$$

où \mathbf{K}_k^g désigne le « gain de Kalman généralisé » :

$$(2.16) \quad \mathbf{K}_k^g = \mathbf{K}_k (\mathbf{R}_k^e)^{-1}$$

Si l'on fait l'hypothèse toujours vérifiée que l'on peut factoriser l'incrément initial sous la forme non unique suivante :

$$(2.17) \quad \delta \mathbf{P}_2 = \mathbf{Y}_1 \mathbf{M}_1 \mathbf{Y}_1^t$$

où \mathbf{Y}_1 et \mathbf{M}_1 sont des matrices de dimensions respectives (n, α) et (α, α) , $\alpha \leq n$, alors on peut établir par récurrence que l'on a :

$$(2.18) \quad \delta \mathbf{P}_{k+1} = \mathbf{Y}_k \mathbf{M}_k \mathbf{Y}_k^t$$

les dimensions des matrices auxiliaires \mathbf{Y}_k et \mathbf{M}_k , $k=1, 2, \dots$ étant constantes. Les relations (2.7) à (2.9) peuvent alors être remplacées par les relations suivantes :

$$(2.19) \quad \mathbf{R}_{k+1}^e = \mathbf{R}_k^e + \mathbf{H} \mathbf{Y}_k \mathbf{M}_k \mathbf{Y}_k^t \mathbf{H}^t$$

$$(2.20) \quad \mathbf{K}_{k+1}^g = (\mathbf{K}_k^g \mathbf{R}_k^e + \mathbf{F} \mathbf{Y}_k \mathbf{M}_k \mathbf{Y}_k^t \mathbf{H}^t) (\mathbf{R}_{k+1}^e)^{-1}$$

$$(2.21) \quad \mathbf{Y}_k = (\mathbf{F} - \mathbf{K}_{k-1}^g \mathbf{H}) \mathbf{Y}_{k-1}$$

$$(2.22) \quad \mathbf{M}_k = \mathbf{M}_{k-1} - \mathbf{M}_{k-1} \mathbf{Y}_{k-1}^t \mathbf{H}^t (\mathbf{R}_k^e)^{-1} \mathbf{H} \mathbf{Y}_{k-1} \mathbf{M}_{k-1}$$

Le rang α de l'incrément initial $\delta \mathbf{P}_2$ étant dans de nombreux cas particuliers très inférieur à la dimension n du vecteur d'état \mathbf{x} , les équations de Chandrasekhar nécessitent bien moins de calcul que les équations habituelles du filtre de Kalman.

2.3. INTERPRÉTATION GÉOMÉTRIQUE DES ÉQUATIONS DE CHANDRASEKHAR

Les équations du filtre de Kalman rapide vues au paragraphe précédent peuvent se mettre sous une forme compacte regroupant dans des matrices-bloc les quantités nominales du filtre disponibles à deux instants consécutifs [10]. Ces formes matricielles sont intéressantes car elles mettent simplement en évidence les liens qui unissent les algorithmes rapides de la famille des moindres carrés. Dans le cas du filtrage

de Kalman, ces matrices sont reliées par une transformation linéaire :

$$(2.22) \quad \begin{bmatrix} \mathbf{R}_{k+1}^e & \mathbf{0} \\ \mathbf{K}_{k+1} & \mathbf{Y}_{k+1} \\ \mathbf{0} & -\mathbf{M}_{k+1}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k^e & \mathbf{H}\mathbf{Y}_k \\ \mathbf{K}_k & \mathbf{F}\mathbf{Y}_k \\ \mathbf{Y}_k^t \mathbf{H}^t & -\mathbf{M}_k^{-1} \end{bmatrix} \times \begin{bmatrix} \mathbf{I}_p & -(\mathbf{R}_k^e)^{-1} \mathbf{H}\mathbf{Y}_k \\ -(\mathbf{R}_k^e)^{-1} \mathbf{Y}_k^t \mathbf{H}^t & \mathbf{I}_\alpha \end{bmatrix}$$

où :

$$(2.23) \quad \mathbf{R}_k^e \triangleq -\mathbf{M}_k^{-1}.$$

La matrice de transformation qui fait passer de l'instant k à l'instant $k+1$:

$$\Theta_k = \begin{bmatrix} \mathbf{I}_p & -(\mathbf{R}_k^e)^{-1} \mathbf{H}\mathbf{Y}_k \\ -(\mathbf{R}_k^e)^{-1} \mathbf{Y}_k^t \mathbf{H}^t & \mathbf{I}_\alpha \end{bmatrix}$$

est telle que :

$$(2.24) \quad \Theta_k^t \mathbf{R}_k \Theta_k = \mathbf{R}_{k+1}$$

avec :

$$(2.25) \quad \mathbf{R}_k \triangleq \begin{bmatrix} \mathbf{R}_k^e & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}_k^e \end{bmatrix}$$

ou bien encore :

$$(2.26) \quad \Theta_k \mathbf{R}_{k+1}^{-1} \Theta_k^t = \mathbf{R}_k^{-1}.$$

La relation (2.22) est une forme d'orthogonalisation pondérée et généralisée puisque la matrice \mathbf{R}_k n'est pas, en général, définie positive et qu'elle ne peut donc pas être associée à une norme euclidienne. On retrouve donc ainsi une caractéristique commune à tous les algorithmes rapides de la famille des moindres carrés. Il est bien connu en effet que l'estimation selon les moindres carrés n'est rien d'autre que le calcul d'un opérateur de projection sur un espace normé approprié. On sait par ailleurs qu'il est beaucoup plus facile de calculer une projection quand on dispose d'une base orthogonale de cet espace. D'où l'idée de remplacer de proche en proche la base initiale par une base orthogonale. Tous ces algorithmes fonctionnent selon le principe de Schmitt.

On peut « normaliser » les équations de Chandrasekhar en absorbant les racines carrées des matrices \mathbf{R}_k^e et \mathbf{R}_k^r :

$$(2.27) \quad \mathbf{R}_k^e = (\mathbf{R}_k^e)^{1/2} (\mathbf{R}_k^e)^{t/2}$$

$$(2.28) \quad \mathbf{R}_k^r = (\mathbf{R}_k^r)^{1/2} \mathbf{S} (\mathbf{R}_k^r)^{t/2}.$$

La matrice \mathbf{S} est la matrice de signature de \mathbf{R}_k^r , ou de $-\mathbf{M}_k$, c'est-à-dire la matrice diagonale formée d'autant de $+1$ et de -1 que la matrice correspondante à de valeurs propres positives et négatives, avec le même ordonnancement. La généralisation par (2.28) de la définition (2.27) d'une racine carrée de matrice est rendue indispensable par le fait que \mathbf{R}_k^r n'est pas nécessairement définie positive. Les factorisations (2.27) et (2.28) ne sont pas uniques, c'est le problème

habituel du choix d'une base orthonormale dans un espace de dimension supérieure à l'unité. \mathbf{S} est sans indice car la signature est la même pour toutes les matrices \mathbf{R}_k^r . Elle ne dépend que de la factorisation initiale.

Les relations (2.27) et (2.28) permettent de réécrire (2.25) sous la forme :

$$(2.29) \quad \mathbf{R}_k \triangleq \begin{bmatrix} (\mathbf{R}_k^e)^{1/2} & \mathbf{0} \\ \mathbf{0} & -(\mathbf{R}_k^e)^{1/2} \end{bmatrix} \times \begin{bmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & -\mathbf{S} \end{bmatrix} \begin{bmatrix} (\mathbf{R}_k^e)^{t/2} & \mathbf{0} \\ \mathbf{0} & -(\mathbf{R}_k^e)^{t/2} \end{bmatrix}$$

ou bien encore, sous forme condensée :

$$(2.30) \quad \mathbf{R}_k = \mathbf{R}_k^{1/2} \mathbb{J} \mathbf{R}_k^{t/2}.$$

La relation d'orthogonalisation pondérée (2.26) devient donc sous forme « normalisée » :

$$(2.31) \quad \tilde{\Theta}_k \mathbb{J} \tilde{\Theta}_k^t = \mathbb{J}$$

avec :

$$(2.32) \quad \tilde{\Theta}_k = \mathbf{R}_k^{t/2} \Theta_k \mathbf{R}_{k+1}^{-t/2}.$$

La matrice \mathbb{J} définit une famille de transformations \mathbb{T} telles que $\mathbb{T} \mathbb{J} \mathbb{T}^t = \mathbb{J}$. Ce sont des transformations J-orthogonales ou hypernormales.

On peut obtenir immédiatement une forme normalisée (ou en « racine carrée ») des équations de Chandrasekhar (2.21) et (2.22) en absorbant les racines carrées de \mathbf{R}_k dans les matrices-bloc :

$$(2.33) \quad \begin{bmatrix} (\mathbf{R}_{k+1}^e)^{1/2} & \mathbf{0} \\ \mathbf{K}_{k+1} (\mathbf{R}_{k+1}^e)^{-t/2} & \mathbf{Y}_{k+1} (\mathbf{R}_{k+1}^e)^{-t/2} \\ \mathbf{0} & (\mathbf{R}_{k+1}^r)^{1/2} \end{bmatrix} = \begin{bmatrix} (\mathbf{R}_k^e)^{1/2} & \mathbf{H}\mathbf{Y}_k (\mathbf{R}_k^e)^{-t/2} \\ \mathbf{K}_k (\mathbf{R}_k^e)^{-t/2} & \mathbf{F}\mathbf{Y}_k (\mathbf{R}_k^e)^{-t/2} \\ (\mathbf{R}_k^r)^{1/2} \mathbf{Y}_k^t \mathbf{H}^t (\mathbf{R}_k^e)^{-t/2} & (\mathbf{R}_k^r)^{1/2} \end{bmatrix} \tilde{\Theta}_k.$$

On a effectué une partition supérieure dans cette relation car on n'a besoin que des seuls blocs supérieurs pour calculer les quantités normalisées $\mathbf{K}_k (\mathbf{R}_k^e)^{-t/2}$ et $(\mathbf{R}_k^e)^{1/2}$ nécessaires à la mise en œuvre de l'algorithme. De plus, comme il n'est pas nécessaire de connaître explicitement la transformation J-orthogonale, on peut, en ne conservant que les blocs supérieurs, obtenir un algorithme rapide « implicite » de la manière suivante :

$$(2.34) \quad \begin{bmatrix} (\mathbf{R}_{k+1}^e)^{1/2} & \mathbf{0} \\ \mathbf{K}_{k+1} (\mathbf{R}_{k+1}^e)^{-t/2} & \tilde{\mathbf{Y}}_{k+1} \end{bmatrix} = \begin{bmatrix} (\mathbf{R}_k^e)^{1/2} & \mathbf{H}\tilde{\mathbf{Y}}_k \\ \mathbf{K}_k (\mathbf{R}_k^e)^{-t/2} & \mathbf{F}\tilde{\mathbf{Y}}_k \end{bmatrix} \mathbb{T}$$

où :

$$(2.35) \quad \tilde{\mathbf{Y}}_k \triangleq \mathbf{Y}_k (\mathbf{R}_k^e)^{-t/2}$$

et où \mathbb{T} est une transformation J-orthogonale quelconque. On peut construire de telles transforma-

tions à partir de la transformation de Householder hyperbolique. Un exemple en sera donné dans le chapitre 3.

2.4. LIENS AVEC LES ÉQUATIONS DE LA PRÉDICTION LINÉAIRE

Il existe de nombreuses situations dans lesquelles il n'est pas possible, ni même souhaitable, d'utiliser de modèle d'état. Supposons par exemple que l'on dispose d'observations y_k , $k=1, 2, \dots$, d'un processus aléatoire de covariance \mathbf{R} connue, et que l'on souhaite estimer un processus aléatoire x_k qui lui est lié, ou bien prédire le processus y_k lui-même. La résolution de ce dernier problème est d'ailleurs la clef de celle du problème général. Le meilleur estimateur linéaire :

$$(2.36) \quad \hat{y}_{n/n-1} = -\mathbf{A}_{n,1} y_{n-1} - \dots - \mathbf{A}_{n,n-1} y_1$$

de y_n étant données les observations y_n , $n=1, 2, \dots$, est obtenu en appliquant un principe d'orthogonalité :

$$(2.37) \quad \mathbf{e}_n \stackrel{\Delta}{=} (y_n - \hat{y}_{n/n-1}) \perp (y_1, \dots, y_{n-1})$$

ce qui conduit à la résolution d'un système d'équations normales :

$$(2.38) \quad \mathbf{A}_n \mathbf{R}_n = [0, \dots, 0, \mathbf{R}_n^e]$$

où \mathbf{R}_n est la matrice de covariance des observations, où \mathbf{R}_n^e est celle de l'innovation \mathbf{e}_n , et où :

$$(2.39) \quad \mathbf{A}_n = [\mathbf{A}_{n,n-1}, \dots, \mathbf{A}_{n,1}, \mathbf{I}]$$

Quand le processus est stationnaire, la matrice \mathbf{R}_n est Toeplitz et l'algorithme de Levinson permet de résoudre le système (2.38) avec une complexité $O(n^2)$ au lieu de $O(n^3)$ dans le cas général. Cet algorithme procède par récurrence sur l'ordre n de la solution et peut s'écrire sous la forme compacte suivante [19] :

$$(2.40) \quad \left[\begin{array}{c|c} \mathbf{R}_{k+1}^e & \mathbf{0} \\ \mathbf{A}_{k+1}^t & \mathbf{B}_{k+1}^t \\ \mathbf{0} & \mathbf{R}_{k+1}^r \end{array} \right] = \left[\begin{array}{c|c} \mathbf{R}_k^e & \Delta_k \\ \mathbf{0} & \mathbf{B}_k^t \\ \mathbf{A}_k^t & \mathbf{0} \\ \Delta_k^t & \mathbf{R}_k^r \end{array} \right] \times \left[\begin{array}{c|c} \mathbf{I}_p & -(\mathbf{R}_k^e)^{-1} \Delta_k \\ \hline -(\mathbf{R}_k^r)^{-1} \Delta_k^t & \mathbf{I}_p \end{array} \right]$$

où les Δ_k sont des covariances résiduelles.

L'analogie avec les équations de Chandrasekhar du paragraphe précédent est flagrante. En effet, la matrice :

$$(2.41) \quad \Psi_k = \left[\begin{array}{c|c} \mathbf{I}_p & -(\mathbf{R}_k^e)^{-1} \Delta_k \\ \hline -(\mathbf{R}_k^r)^{-1} \Delta_k^t & \mathbf{I}_p \end{array} \right]$$

obéit aux mêmes relations :

$$(2.42) \quad \Psi_k^r \mathbf{R}_k \Psi_k = \mathbf{R}_{k+1}, \quad \Psi_k \mathbf{R}_{k+1}^{-1} \Psi_k^t = \mathbf{R}_k^{-1}$$

que la matrice Θ_k du paragraphe 2.3. En fait on peut montrer par des techniques diverses [7, 10, 18, 21, 22] que les équations de Chandrasekhar peuvent être

déduites de celles de Levinson lorsque le processus observé est stationnaire et possède un modèle d'état invariant et connu. Dans ce cas, le calcul de la suite des filtres prédictifs de l'état se simplifie. Le fait que les équations de Chandrasekhar correspondent à une récurrence sur le temps et que celles de Levinson correspondent à une récurrence sur l'ordre n'est pas un obstacle au passage des unes aux autres puisqu'il est possible d'écrire les équations de Levinson dans un format où tous les vecteurs sont de dimension fixe [3] (cf. chap. 4).

Lorsque le processus observé y est strictement stationnaire, c'est-à-dire quand :

$$(2.43) \quad \mathbf{P}_{1/0} = \mathbf{F} \mathbf{P}_{1/0} \mathbf{F}^t + \mathbf{G} \mathbf{Q} \mathbf{G}^t$$

le facteur central de l'incrément initial de covariance d'erreur devient $\mathbf{M}_0 = -\mathbf{I}$ et la matrice de transformation Θ_k est simplement orthogonale. La dissymétrie qui existait entre la prédiction directe et la prédiction rétrograde, et qui se manifestait par une différence des dimensions des quantités auxiliaires ($\alpha \neq p$), a disparu.

La différence principale entre les deux familles d'algorithmes tient plutôt à la nature physique des quantités auxiliaires calculées. Alors que l'algorithme de Levinson propage des coefficients de prédiction directe et rétrograde \mathbf{A}_n et \mathbf{B}_n du processus observé, l'algorithme de Chandrasekhar propage des covariances mutuelles \mathbf{K}_n et \mathbf{Y}_n entre le vecteur d'état et les innovations de ce processus observé. Il s'agit donc, selon la classification usuelle [15], d'un algorithme du type Schur.

2.5. QUELQUES REMARQUES

— Contrairement à une opinion répandue, l'emploi des équations de Chandrasekhar implique que le modèle d'état correspondant soit invariant, mais pas que le processus observé soit stationnaire. La matrice de covariance *a priori* $\mathbf{P}_{1/0}$ n'est en effet soumise à aucune contrainte. Il est clair cependant que l'intérêt de ces équations est directement lié à la faiblesse du rang α de l'incrément initial.

— Nous avons vu que la normalisation d'un algorithme de Chandrasekhar faisait intervenir dans le cas général des transformations hypernormales. Comme ces dernières ne conservent que la norme hyperbolique des vecteurs auxquels elles s'appliquent, on voit que ces algorithmes ne sont pas exempts de problèmes de stabilité numérique.

— Les conditions initiales du filtre jouent un rôle essentiel puisque ce sont elles qui fixent la complexité de l'algorithme. On a même pu dire que les conditions initiales intéressantes étaient de mesure nulle dans l'ensemble des conditions possibles ! Il est malgré tout possible d'y remédier en partie, du moins lorsqu'on ne recherche que la solution asymptotique de l'équation de Riccati. Il suffit pour cela de décomposer la matrice de covariance initiale en la somme d'une partie nominale possédant les bonnes propriétés et d'un terme résiduel dont l'effet sur la solution est propagé par des équations de Chandrasekhar duales [9, 23].

3. Estimation adaptative des coefficients d'une régression linéaire

Un problème fréquent en traitement du signal consiste à estimer les coefficients d'une régression linéaire apparaissant dans un modèle du type :

$$(3.1) \quad y(n) = z(n) + b(n), \quad n = 1, 2, \dots, N$$

$$(3.2) \quad z(n) = \sum_{i=1}^p a_i x(n-i) = \mathbf{a}^t \mathbf{x}_p(n)$$

$$(3.3) \quad \mathbf{a}^t = [a_p, a_{p-1}, \dots, a_1]$$

$$(3.4) \quad \mathbf{x}_p(n)^t = [x(n-p), \dots, x(n-1)]$$

où \mathbf{a} est le vecteur des coefficients du modèle dont l'ordre est p , et où $b(n)$ est un b. b. gaussien centré de variance σ^2 . $x(n)$ est tantôt l'entrée d'un système à identifier dont $y(n)$ est la sortie et \mathbf{a} la réponse impulsionnelle, tantôt $y(n)$ lui-même dans le cas d'un problème de prédiction linéaire. \mathbf{a} est alors le vecteur des coefficients de prédiction.

Les coefficients \mathbf{a} ne sont pas constants, mais ils sont supposés varier lentement par rapport aux fluctuations des signaux observés. On souhaite donc employer des méthodes adaptatives faisant l'hypothèse d'une invariance locale du modèle. Cette adaptativité est obtenue par simple modification du critère usuel des moindres carrés. Si l'on désigne par $e(n)$ l'erreur de prédiction de $y(n)$, on distingue ainsi trois types de méthodes associées à trois types de critères $J(\mathbf{a})$:

– *Traitement par bloc.*

$$(3.5) \quad J(\mathbf{a}) = \sum_{i=P}^Q e(n)^2$$

– *Pondération exponentielle.*

$$(3.6) \quad J(\mathbf{a}) = \sum_{i=1}^N \lambda^{N-i} e(n)^2$$

– *Fenêtre glissante.*

$$(3.7) \quad J(\mathbf{a}) = \sum_{i=N-L}^N e(n)^2$$

Les algorithmes rapides qui résolvent ce problème exploitent les propriétés d'invariance par décalage des vecteurs $\mathbf{x}_p(n)$ successifs. Leur diversité provient en grande partie de la variété des effets de bord (fenêtrage). Nous allons montrer qu'il est possible d'appliquer les techniques de filtrage de Kalman rapide vues au chapitre précédent.

Ces techniques ne semblent pourtant pas pouvoir s'appliquer puisque, si l'on considère la relation :

$$(3.8) \quad y(n) = \mathbf{x}_p(n)^t \mathbf{a} + b(n), \quad n = 1, 2, \dots, N$$

comme étant l'équation d'observation d'un modèle d'état dont \mathbf{a} serait le vecteur d'état, elle ne présente pas les propriétés d'invariance nécessaires, $\mathbf{x}_p(n)$ n'est pas strictement invariant, mais seulement invariant

par décalage. Il faut donc expliciter ces propriétés d'invariance par décalage. On définit pour cela un « vecteur-paramètre étendu » à m composantes :

$$(3.9) \quad \begin{cases} \mathbf{a}_m(i) = [0_{i-1}^t, a_p, a_{p-1}, \dots, a_1, 0, 0, \dots, 0]^t, \\ m \gg p \end{cases}$$

et le vecteur global des données :

$$(3.10) \quad \mathbf{x}_m = [x(-p+1), x(-p+2), \dots, x(0), x(1), \dots, x(m-p)]^t$$

Le modèle linéaire défini par les équations (3.1) à (3.4) peut alors se mettre sous la forme du modèle d'état équivalent suivant :

$$(3.11) \quad \mathbf{a}_m(i+1) = \mathbf{D} \mathbf{a}_m(i), \quad i = 1, 2, \dots$$

$$(3.12) \quad y(i) = \mathbf{x}_m^t \mathbf{a}_m(i) + b(i)$$

où \mathbf{D} est l'opérateur de décalage :

$$(3.13) \quad \mathbf{D} = \begin{bmatrix} 0 & 0 & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & \dots \\ 0 & 1 & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}$$

Cette technique présente une analogie certaine avec celle introduite par Morf pour développer des algorithmes des moindres carrés récursifs rapides. Mais au lieu d'utiliser des quantités auxiliaires dont la dimension est simplement augmentée d'une unité par rapport à la dimension p du modèle, nous introduisons un vecteur paramètre étendu de très grande dimension. Il ne s'agit là que d'un artifice destiné à expliciter la propriété d'invariance que l'on cherche à exploiter. Le principal avantage est qu'il permet ensuite de « dérouler » les mêmes équations dans tous les cas.

La principale différence avec les méthodes des moindres carrés récursifs rapides est dans le vecteur-paramètre qui est maintenant considéré comme une quantité aléatoire caractérisée par sa moyenne et sa matrice de covariance *a priori*. Nous nous retrouvons donc dans un cadre statistique bayésien.

3.2. CAS D'UN TRAITEMENT PAR BLOCS

Ce cas est traité en détail. Le calcul récurent de la solution peut être effectué par le filtre de Kalman suivant [16] :

$$(3.14) \quad \hat{\mathbf{a}}_m(i+1/i) = \mathbf{D} \hat{\mathbf{a}}_m(i/i-1) + \mathbf{k}_m(i) r(i)^{-1} [y(i) - \mathbf{x}_m^t \hat{\mathbf{a}}_m(i/i-1)]$$

$$(3.15) \quad r(i) = \mathbf{x}_m^t \mathbf{P}_m(i/i-1) \mathbf{x}_m + \sigma^2$$

$$(3.16) \quad \mathbf{k}_m(i) = \mathbf{D} \mathbf{P}_m(i/i-1) \mathbf{x}_m$$

$$(3.17) \quad \mathbf{P}_m(i+1/i) = \mathbf{D} \mathbf{P}_m(i/i-1) \times \mathbf{D}^t - \mathbf{k}_m(i) r(i)^{-1} \mathbf{k}_m^t(i)$$

Les conditions initiales sont les suivantes :

$$(3.18) \quad \hat{\mathbf{a}}_m(1/0) = [\mathbf{a}(1/0), 0, \dots, 0]^t$$

$$(3.19) \quad r(1) = \mathbf{x}_m^t \mathbf{P}_m(1/0) \mathbf{x}_m + \sigma^2$$

$$(3.20) \quad \mathbf{k}_m(1) = \mathbf{D} \mathbf{P}_m(1/0) \mathbf{x}_m$$

$$(3.21) \quad \mathbf{P}_m(1/0) = \left[\begin{array}{c|c} \mathbf{P}(1/0) & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array} \right].$$

On peut vérifier qu'à chaque récurrence le nombre de coordonnées non nulles du vecteur-gain $\mathbf{k}_m(i)$ reste égal à p . Ce modèle d'état étant invariant (\mathbf{D} et \mathbf{x}_m constants) et à bruit stationnaire (σ^2 constant sur un bloc), on obtient directement un algorithme rapide en remplaçant les équations (3.15) à (3.17) par des équations de Chandrasekhar.

On part pour cela de la factorisation :

$$(3.22) \quad \delta \mathbf{P}_m(k+1) \triangleq \mathbf{B}(k) \mathbf{M}(k) \mathbf{B}(k)^t.$$

L'application directe des techniques décrites au chapitre 2 conduit aux équations suivantes :

$$(3.23) \quad \mathbf{v}_m(i)^t \triangleq \mathbf{x}_m^t \mathbf{B}(i)$$

$$(3.24) \quad \mathbf{w}_m(i)^t \triangleq \mathbf{v}_m(i)^t \mathbf{M}(i)$$

$$(3.25) \quad \mathbf{B}(i+1) = \mathbf{D} \mathbf{B}(i) - \mathbf{k}_m(i) r(i)^{-1} \mathbf{v}_m(i)^t$$

$$(3.26) \quad \mathbf{k}_m(i+1) = \mathbf{k}_m(i) + \mathbf{D} \mathbf{B}(i) \mathbf{w}_m(i)$$

$$(3.27) \quad r(i+1) = r(i) + \mathbf{v}_m(i)^t \mathbf{w}_m(i)$$

$$(3.28) \quad \mathbf{M}(i+1) = \mathbf{M}(i) - \mathbf{w}_m(i) r(i+1)^{-1} \mathbf{w}_m(i)^t.$$

Une écriture condensée de l'algorithme est obtenue par projection, en ne faisant apparaître que les composantes intervenant effectivement à chaque récurrence dans le calcul. On définit pour cela un vecteur-gain effectif $\mathbf{k}(i)$ de dimension p contenant les coordonnées non nulles de $\mathbf{k}_m(i)$:

$$(3.29) \quad \mathbf{k}_m(i) = [\mathbf{o}_{i+1}^t, \mathbf{k}(i)^t, \mathbf{o}_{m-p-i-1}^t]^t.$$

De la même façon, si α est la dimension de $\mathbf{M}(i)$, on peut aussi écrire :

$$(3.30) \quad \mathbf{B}(i) = [\mathbf{0}_{\alpha, i}, \mathbf{S}(i)^t, \mathbf{0}_{\alpha, m-p-i-1}]^t$$

où $\mathbf{S}(i)$ est une matrice de dimension $[(p+1), \alpha]$. L'algorithme peut alors se mettre sous la forme réduite :

$$(3.31) \quad \mathbf{v}(i)^t = [x(i-p), \mathbf{x}_p(i+1)]^t \mathbf{S}(i)$$

$$(3.32) \quad \mathbf{w}(i)^t = \mathbf{v}(i)^t \mathbf{M}(i)$$

$$(3.33) \quad \mathbf{S}(i+1) = \mathbf{S}(i) - \left[\begin{array}{c} \mathbf{k}(i) \\ 0 \end{array} \right] r(i)^{-1} \mathbf{v}(i)^t$$

$$(3.34) \quad r(i+1) = r(i) + \mathbf{v}(i)^t \mathbf{w}(i)$$

$$(3.35) \quad \left[\begin{array}{c} 0 \\ \mathbf{k}(i+1) \end{array} \right] = \left[\begin{array}{c} \mathbf{k}(i) \\ 0 \end{array} \right] + \mathbf{S}(i) \mathbf{w}(i)$$

$$(3.36) \quad \mathbf{M}(i+1) = \mathbf{M}(i) - \mathbf{w}(i) r(i+1)^{-1} \mathbf{w}(i)^t$$

la remise à jour des paramètres se faisant selon :

$$(3.37) \quad \hat{\mathbf{a}}(i+1/i) = \hat{\mathbf{a}}(i/i-1) + \mathbf{k}(i) r(i)^{-1} [y(i) - \mathbf{x}_p(i)^t \hat{\mathbf{a}}(i/i-1)].$$

On reconnaît sous cette forme un algorithme du type «Schur généralisé». Il présente l'avantage appréciable de s'appliquer aussi bien à des problèmes du type «préfenêtre» qu'à des problèmes du type «covariance». Nous allons montrer, comme on peut s'y attendre, qu'une simplification substantielle de l'algorithme peut être obtenue en cas de «préfenêtrage» des données. Cette simplification se produit à l'initialisation, au moment où sont fixées les dimensions des quantités auxiliaires introduites dans la factorisation de l'incrément de la matrice de covariance $\mathbf{P}_m(k/k-1)$, c'est-à-dire au moment où se précise la complexité calculatoire de la méthode. L'initialisation de l'algorithme est en effet directement conditionnée par la matrice de covariance *a priori* $\mathbf{P}_m(1/0)$ dont se déduisent $\mathbf{B}(1)$ et $\mathbf{M}(1)$ lors de la factorisation de l'incrément initial :

$$(3.38) \quad \delta \mathbf{P}(2) = \mathbf{P}_m(2/1) - \mathbf{P}_m(1/0) = \mathbf{B}(1) \mathbf{M}(1) \mathbf{B}(1)^t.$$

Mais l'initialisation dépend aussi, par l'intermédiaire de $\mathbf{P}(2/1)$, de la forme que prend le vecteur global des données \mathbf{x}_m , selon les valeurs attribuées au signal antérieurement à l'instant initial $t=1$. Il faut donc considérer deux cas.

3.1.1. Méthode dite «de covariance»

Dans cette méthode, les valeurs antérieures à l'instant initial dans le vecteur global \mathbf{x}_m ont été observées. Nous avons donc, en revenant aux équations du filtre sous forme étendue :

$$(3.39) \quad r(1) = \mathbf{x}_m^t \mathbf{P}_m(1/0) \mathbf{x}_m + \sigma^2$$

$$(3.40) \quad \mathbf{k}_m(1) = \mathbf{D} \mathbf{P}_m(1/0) \mathbf{x}_m.$$

L'incrément initial de covariance peut donc s'écrire :

$$(3.41) \quad \delta \mathbf{P}_m(2) = \mathbf{P}_m(2/1) - \mathbf{P}_m(1/0) = \mathbf{D} \mathbf{P}_m(1/0) \mathbf{D}^t - \mathbf{k}_m(1) r(1)^{-1} \mathbf{k}_m(1)^t - \mathbf{P}_m(1/0)$$

avec :

$$(3.42) \quad \mathbf{P}_m(1/0) = \left[\begin{array}{c|c} \mathbf{P}(1/0) & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array} \right].$$

Si l'on suppose que la covariance *a priori* des paramètres est stationnaire :

$$(3.43) \quad \mathbf{P}(1/0) = \gamma_0 \left[\begin{array}{c|c|c|c} 1 & \gamma_1 & \gamma_2 & \dots \\ \gamma_1 & 1 & \gamma_1 & \dots \\ \cdot & \cdot & \cdot & \dots \\ \gamma_{p-1} & \cdot & \cdot & \dots \end{array} \right]$$

alors on peut décomposer :

$$(3.43) \quad \mathbf{D} \mathbf{P}_m(1/0) \mathbf{D}^t - \mathbf{P}_m(1/0) = -\gamma_0 (\mathbf{s} \mathbf{s}^t + \mathbf{s} \mathbf{g}^t + \mathbf{g} \mathbf{s}^t) + \gamma_0 (\mathbf{v} \mathbf{v}^t + \mathbf{v} \mathbf{d}^t + \mathbf{d} \mathbf{v}^t)$$

où :

$$(3.44) \quad \mathbf{s}^t = [1, 0, 0, \dots]$$

$$(3.45) \quad \mathbf{g}^t = [0, \gamma_1, \gamma_2, \dots, \gamma_{p-1}, 0, 0, \dots]$$

$$(3.46) \quad \mathbf{v}^t = [0, 0, \dots, 0, 1, 0, 0, \dots]$$

$$(3.47) \quad \mathbf{d}^t = [0, \gamma_{p-1}, \gamma_{p-2}, \dots, \gamma_1, 0, 0, \dots].$$

Dans le cas particulier où la matrice de covariance *a priori* $\mathbf{P}(1/0)$ est choisie diagonale, les vecteurs \mathbf{g} et \mathbf{d} sont identiquement nuls, et l'incrément initial se réduit alors à :

$$(3.48) \quad \delta\mathbf{P}(2) = -\gamma_0 \mathbf{s}\mathbf{s}^t + \gamma_0 \mathbf{v}\mathbf{v}^t - \mathbf{k}_m(1) r(1)^{-1} \mathbf{k}_m(1)^t.$$

Son rang est donc au plus égal à 3 puisqu'il se décompose en une somme de 3 matrices dyadiques ou anti-scalaires, chacune de rang égal à 1. Nous choisissons donc comme facteur central de :

$$(3.49) \quad \delta\mathbf{P}(2) = \mathbf{B}(1) \mathbf{M}(1) \mathbf{B}(1)^t$$

$$(3.50) \quad \mathbf{M}(1) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ce qui conduit, une fois la projection effectuée pour obtenir l'algorithme en dimension réduite, à :

$$(3.51) \quad \mathbf{S}(1) = [\tilde{\mathbf{k}}(1), \mathbf{b}, \mathbf{c}]$$

$$(3.52) \quad \tilde{\mathbf{k}}(1) = [0, \mathbf{k}(1)^t r(1)^{-1/2}]^t$$

$$(3.53) \quad \mathbf{k}(1) = \gamma_0 \mathbf{x}_p(1)$$

$$(3.54) \quad \mathbf{b} = [\gamma_0^{1/2}, 0, 0, \dots, 0]^t$$

$$(3.55) \quad \mathbf{c} = [0, 0, 0, \dots, \gamma_0^{1/2}]^t$$

$$(3.56) \quad r(1) = \mathbf{x}_m^t \mathbf{P}_m(1/0) \mathbf{x}_m + \sigma^2 \\ = \gamma_0 \mathbf{x}_p(1)^t \mathbf{x}_p(1) + \sigma^2.$$

3.1.2. Méthode dite « préfenêtrée »

Dans ce cas des simplifications vont survenir à cause de la structure particulière du vecteur global des données \mathbf{x}_m . On a en effet :

$$(3.57) \quad \mathbf{x}_p(1)^t = \mathbf{0}$$

avec les conséquences suivantes :

$$(3.58) \quad \mathbf{k}_m(1) = \mathbf{D}\mathbf{P}_m(1/0) \mathbf{x}_m = \mathbf{0}$$

$$(3.58) \quad r(1) = \gamma_0 \mathbf{x}_p(1)^t \mathbf{x}_p(1) + \sigma^2 = \sigma^2$$

$$(3.59) \quad \delta\mathbf{P}(2) = -\gamma_0 \mathbf{s}\mathbf{s}^t + \gamma_0 \mathbf{v}\mathbf{v}^t.$$

Le rang de l'incrément initial n'est plus que $\alpha=2$ au lieu de 3 ! Nous pouvons donc choisir comme facteur initial :

$$(3.60) \quad \mathbf{M}(1) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$(3.61) \quad \mathbf{S}(1) = [\mathbf{b}, \mathbf{c}].$$

La complexité algorithmique de la méthode, qui est de l'ordre de $3\alpha p + 3\alpha^2 + 5\alpha$ multiplications par instant d'échantillonnage (mults), passe ainsi de $9p + 42$ mults à $6p + 22$ mults. Le gain est donc d'environ 30%, ce

qui est appréciable dans une application en temps-réel.

On voit donc que cette complexité passe de $6p$ à $9p$, $12p$ et $15p$ selon que le signal est préfenêtré ou non et que les paramètres sont corrélés *a priori* ou non. Mais ce qui est remarquable, c'est qu'on utilise le même algorithme dans tous les cas.

3.1.3. Hypernormalisation de l'algorithme

La diminution de la complexité numérique des algorithmes de la famille des moindres carrés récurrents, tel le filtre de Kalman, s'obtient en expurgeant toute la redondance que peuvent présenter les équations standards de ces algorithmes lorsque les modèles des signaux considérés présentent certaines propriétés d'invariance. Mais cette rapidité est souvent obtenue au détriment de la robustesse numérique, et il est bien connu que ces algorithmes rapides peuvent poser des problèmes de stabilité numérique pouvant aller jusqu'à la divergence [30]. Un moyen classique de réduire la sensibilité d'un algorithme à la précision limitée des calculs en machine consiste à propager les « racines carrées » des matrices de covariance plutôt que les matrices elles-mêmes [25]. Outre que cela permet de conserver obligatoirement le caractère défini de ces matrices dont la perte annonce bien souvent la divergence, cela permet de gagner aussi un facteur 2 sur la dynamique des composantes de ces matrices. Nous avons donc développé une version « en racine carrée » de l'algorithme précédent afin d'évaluer l'amélioration qui pourrait en résulter, et de disposer, le cas échéant, d'un algorithme ayant de meilleures qualités numériques.

Cette écriture en racine carrée repose sur l'interprétation géométrique des filtres de Kalman rapides et sur l'emploi de transformations matricielles particulières, les transformations de Householder hyperboliques.

La transformation de Householder ordinaire est couramment utilisée dans les problèmes de triangularisation de matrice. On sait qu'une matrice de Householder est orthonormale. Mais les matrices de Householder hyperboliques ne le sont pas. On les désigne souvent par matrices « hypernormales » parce qu'elles conservent la norme hyperbolique des vecteurs auxquels elles s'appliquent, alors que les matrices de Householder ordinaires conservent la norme euclidienne usuelle [28].

Soit \mathbf{S} une matrice de signature, c'est-à-dire une matrice diagonale dont les éléments diagonaux sont +1 ou -1. Soit \mathbf{x} un vecteur réel. On appelle la quantité :

$$(3.62) \quad \mathbf{x}^t \mathbf{S} \mathbf{x} = \sum_{i=1}^N |x_i|^2 s_{ii} = \sum_{i=1}^N \pm |x_i|^2$$

la « norme hyperbolique » de \mathbf{x} . Toute matrice \mathbf{H} qui satisfait la relation :

$$(3.63) \quad \mathbf{H}\mathbf{S}\mathbf{H}^t = \mathbf{S}$$

est appelée matrice hypernormale puisqu'elle préserve la norme hyperbolique de tout vecteur auquel elle est appliquée. Il est important de remarquer qu'une matrice qui est hypernormale pour une matrice de

signature S_1 n'est pas en général hypernormale pour une autre matrice de signature S_2 . On doit donc spécifier la matrice de signature lorsqu'on discute d'hypernormalité.

Une matrice Q construite de la manière suivante :

$$(3.64) \quad Q = S - 2bb^t/b^t S b$$

où b est un vecteur de dimension appropriée, est une matrice de Householder hyperbolique qui satisfait la relation (3.63). On vérifie que si $S=I$, alors Q est une matrice de Householder ordinaire, c'est-à-dire simplement orthonormale.

La transformation de Householder correspond à la notion géométrique de réflexion d'un vecteur à travers un plan [28]. Elle permet notamment de triangulariser une matrice en procédant ligne par ligne et en calculant à chaque fois la transformation Q telle que :

$$(3.65) \quad Qu_j = \sigma e_j$$

où u_j est un vecteur représentant la j -ième ligne de la matrice à triangulariser, et où e_j est un vecteur dont la j -ième coordonnée vaut l'unité et dont toutes les autres sont nulles. Il suffit de choisir dans (3.64) :

$$(3.66) \quad b = u_j + \sigma e_j.$$

À l'initialisation de l'algorithme, la factorisation que nous avons retenue (3.50) ou (3.60) nous met bien l'algorithme sous forme normalisée puisque nous avons :

$$(3.65) \quad M(1) = S.$$

Mais cette forme n'est pas conservée par la suite si l'on utilise les récurrences de Chandrasekhar car $M(k)$ évolue et n'est plus une matrice de signature lorsque $k > 1$. L'utilisation d'une transformation de Householder hyperbolique entre deux récurrences successives permettra de propager les quantités définies dans (2.34) au lieu de (2.22), en choisissant comme matrice T la matrice Q hyperbolique au sens de \mathbb{J} :

$$(3.66) \quad \left[\begin{array}{c|c} r(i+1)^{1/2} & \mathbf{0}^t \\ \hline \tilde{\mathbf{k}}_m(i+1) & \tilde{\mathbf{B}}(i+1) \end{array} \right] = \left[\begin{array}{c|c} r(i)^{1/2} & \mathbf{x}_m^t \tilde{\mathbf{B}}(i) \\ \hline \tilde{\mathbf{k}}_m(i) & D\tilde{\mathbf{B}}(i) \end{array} \right] Q$$

où l'on a posé :

$$(3.67) \quad \tilde{\mathbf{B}}(i) = \mathbf{B}(i) (-\mathbf{M}(i)^{1/2})$$

et :

$$(3.68) \quad \tilde{\mathbf{k}}_m(i) = \mathbf{k}_m(i) r(i)^{-1/2}.$$

Cette matrice peut se décomposer de la manière suivante :

$$(3.67) \quad Q \left[\begin{array}{c|c} q_1 & \mathbf{q}_2^t \\ \hline \mathbf{q}_2 & Q_3 \end{array} \right]$$

où q_1 est un scalaire puisque l'observation $y(n)$, et donc son innovation $e(n)$, sont scalaires; où \mathbf{q}_2 est un vecteur de dimension α ; et où Q_3 est une matrice de dimension (α, α) , α étant le rang de l'incrément initial. En développant la relation (3.66) on obtient les équations normalisées du filtre mettant en jeu les quantités

étendues :

$$(3.68) \quad r(i+1)^{1/2} = r(i)^{1/2} q_1 + \mathbf{x}_m^t \tilde{\mathbf{B}}(i) \mathbf{q}_2$$

$$(3.69) \quad \tilde{\mathbf{k}}_m(i+1) = \tilde{\mathbf{k}}_m(i) q_1 + D\tilde{\mathbf{B}}(i) \mathbf{q}_2$$

$$(3.70) \quad \tilde{\mathbf{B}}(i+1) = \tilde{\mathbf{k}}_m(i) \mathbf{q}_2^t + D\tilde{\mathbf{B}}(i) Q_3$$

$$(3.71) \quad \hat{\mathbf{a}}_m(i+1/i) = D\hat{\mathbf{a}}_m(i/i-1) + \tilde{\mathbf{k}}_m(i) r(i)^{-1/2} [y(i) - \mathbf{x}_m^t(i) \hat{\mathbf{a}}_m(i/i-1)].$$

Ces quantités étendues normalisées ont la même structure que leurs homologues non normalisées, et on obtient par une projection analogue les équations de l'algorithme en racine carrée. On définit pour cela un vecteur-gain effectif $\tilde{\mathbf{k}}(i)$ de dimension p contenant les coordonnées non nulles de $\tilde{\mathbf{k}}_m(i)$:

$$(3.72) \quad \tilde{\mathbf{k}}_m(i) = [0_{i+1}^t, \tilde{\mathbf{k}}(i)^t, 0_{m-p-i-1}^t]^t.$$

De la même façon, si α est la dimension de $M(i)$, on peut aussi écrire :

$$(3.73) \quad \tilde{\mathbf{B}}(i) = [0_{\alpha, i}, L(i)^t, 0_{\alpha, m-p-i-1}^t]^t$$

où $L(i)$ est une matrice de dimensions $[(p+1), \alpha]$. L'algorithme peut alors se mettre sous la forme réduite :

$$(3.74) \quad r(i+1)^{1/2} = r(i)^{1/2} q_1 + [x(i-p), \mathbf{x}_p(i+1)^t] L(i) \mathbf{q}_2$$

$$(3.75) \quad \left[\begin{array}{c} 0 \\ \tilde{\mathbf{k}}(i+1) \end{array} \right] = r(i)^{-1/2} r(i+1)^{1/2} \times q_1 \left[\begin{array}{c} \tilde{\mathbf{k}}(i) \\ 0 \end{array} \right] + r(i+1)^{1/2} L(i) \mathbf{q}_2$$

$$(3.76) \quad L(i+1) = L(i) Q_3 + r(i)^{-1/2} \left[\begin{array}{c} \tilde{\mathbf{k}}(i) \\ 0 \end{array} \right] \mathbf{q}_2^t.$$

La remise à jour des paramètres se faisant selon :

$$(3.77) \quad \hat{\mathbf{a}}(i+1/i) = \hat{\mathbf{a}}(i/i-1) + \tilde{\mathbf{k}}(i) r(i)^{-1/2} [y(i) - \mathbf{x}_p(i)^t \hat{\mathbf{a}}(i/i-1)].$$

Le décompte des opérations arithmétiques élémentaires montre que le volume de calcul par récurrence est environ doublé lorsque l'on passe de la version normale de l'algorithme à sa version en racine carrée. À l'inverse, les expérimentations numériques que nous avons effectuées montrent que la stabilité numérique est effectivement améliorée par l'hypernormalisation qui repousse la limite de divergence de l'algorithme.

3.2. PONDÉRATION EXPONENTIELLE DES DONNÉES

Ce cas, ainsi que le suivant, ne sera pas détaillé. Le lecteur intéressé pourra se reporter à [16] par exemple. Le passage d'un traitement par bloc, ou à mémoire croissante, à un traitement par pondération exponentielle se fait de façon très simple puisque le seul effet (ce qui d'ailleurs le but recherché) de cette pondération est de modifier légèrement l'équation d'évolution de la matrice de covariance $P_{k/k-1}$. La relation (2.22)

devient ainsi :

$$(3.78) \quad \left[\begin{array}{c|c} \mathbf{R}_{k+1}^e & \mathbf{0} \\ \mathbf{K}_{k+1} & \mathbf{Y}_{k+1} \\ \mathbf{0} & -(\lambda \mathbf{M}_{k+1})^{-1} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{R}_k^e & \mathbf{HY}_k \\ \mathbf{K}_k & \mathbf{FY}_k \\ \mathbf{Y}_k^t \mathbf{H}^t & -\mathbf{M}_k^{-1} \end{array} \right] \Theta_k$$

La modification est donc mineure et la technique présentée au paragraphe précédent s'applique immédiatement.

3.3. FENÊTRE GLISSANTE

Lorsque l'on fait une estimation récursive avec une fenêtre glissante sur les données, la forme du filtre change notablement puisqu'il faut calculer deux gains à chaque récurrence, l'un pour propager l'innovation directe correspondant à la donnée qui entre dans la fenêtre, l'autre pour oublier l'innovation rétrograde correspondant à celle qui sort de la fenêtre [1]. La remise à jour de l'estimée se fait selon la relation :

$$(3.79) \quad \hat{\mathbf{x}}_{k+1} = \mathbf{F} \hat{\mathbf{x}}_k + \mathbf{K}_k^e (\mathbf{R}_k^e)^{-1} \mathbf{e}_k - \mathbf{K}_k^r (\mathbf{R}_k^r)^{-1} \mathbf{r}_k$$

Les équations de Chandrasekhar correspondantes sont plus longues à établir [16], mais elles peuvent toujours se mettre sous une forme compacte faisant appel à deux transformations analogues à celle de l'équation (2.22) :

$$(3.80) \quad \left[\begin{array}{c|c} \mathbf{R}_{k+1}^e & \mathbf{0} \\ \mathbf{K}_{k+1}^e & \mathbf{Z}_{k+1} \\ \mathbf{0} & -\mathbf{N}_{k+1}^{-1} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{R}_k^e & \mathbf{HFY}_k \\ \mathbf{K}_k^e & \mathbf{FY}_k \\ \mathbf{Y}_k^t \mathbf{F}^t \mathbf{H}^t & -\mathbf{M}_k^{-1} \end{array} \right] \times \left[\begin{array}{c|c} \mathbf{I}_1 & -(\mathbf{R}_k^e)^{-1} \mathbf{HFY}_k \\ -\mathbf{M}_k \mathbf{Y}_k^t \mathbf{F}^t \mathbf{H}^t & \mathbf{I}_2 \end{array} \right]$$

$$(3.81) \quad \left[\begin{array}{c|c} \mathbf{R}_{k+1}^r & \mathbf{0} \\ \mathbf{K}_{k+1}^r & \mathbf{Y}_{k+1} \\ \mathbf{0} & -\mathbf{M}_{k+1}^{-1} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{R}_k^r & \mathbf{HF}^{-L} \mathbf{Z}_k \\ \mathbf{K}_k^r & \mathbf{Z}_k \\ \mathbf{Z}_k^t \mathbf{F}^{-L,t} \mathbf{H}^t & -\mathbf{N}_k^{-1} \end{array} \right] \times \left[\begin{array}{c|c} \mathbf{I}_3 & -(\mathbf{R}_k^r)^{-1} \mathbf{HF}^{-L} \mathbf{Y}_k \\ -\mathbf{N}_k \mathbf{Z}_k^t \mathbf{F}^{-L,t} \mathbf{H}^t & \mathbf{I}_4 \end{array} \right]$$

3.5. EXEMPLE D'APPLICATION

Un exemple d'application de ces algorithmes rapides à l'analyse spectrale adaptative peut être trouvé dans [17]. Le principe de la méthode consiste à utiliser de manière combinée :

- un modèle autorégressif (AR) pour conserver la bilinéarité du modèle et une solution algorithmique simple;
- un ordre élevé pour améliorer la résolution potentielle de la méthode;
- une approche statistique bayésienne pour estimer les paramètres du modèle.

La réduction d'information indispensable à toute modélisation d'un signal ne s'effectue plus alors au niveau du modèle AR, avec la contrainte d'un ordre

relativement faible, mais au niveau de la loi *a priori* sur ces paramètres qui est caractérisée par un petit nombre d'hyperparamètres. Ceux-ci sont estimés par maximisation de leur vraisemblance marginale dont le calcul est considérablement simplifié par l'emploi de l'algorithme précédent. L'ensemble est décrit dans [17] et la méthode est effectivement utilisée pour l'analyse des signaux Doppler.

4. Estimation d'un signal ou d'une image

Nous considérons dans ce chapitre le problème dual du précédent, c'est-à-dire celui de l'estimation d'un processus aléatoire \mathbf{x}_k de dimension n à partir d'observations d'un processus lié \mathbf{y}_k de dimension p . Quand l'information *a priori* porte sur les moments d'ordre deux de leur loi conjointe, nous avons vu au chapitre 2 que les différents problèmes de prédiction ou de lissage ont en commun de nécessiter, pour leur résolution, l'inversion d'une matrice de covariance. Quand le processus observé \mathbf{y}_k est stationnaire, cette matrice est Toeplitz, et il est bien connu que l'algorithme de Levinson permet de l'inverser rapidement. Quand le processus \mathbf{y}_k n'est pas stationnaire, l'utilisation du concept de rang de déplacement permet de passer progressivement d'une complexité $O(n^2)$ dans le cas Toeplitz à une complexité $O(n^3)$ dans le cas le plus général [11, 12]. L'algorithme rapide de calcul récursif du filtre optimal fait appel à des quantités auxiliaires qui sont les matrices de prédiction directe et rétrograde du processus observé \mathbf{y}_k . Mais le calcul de ces prédicteurs se fait à l'aide d'un algorithme de Levinson généralisé dont l'établissement utilise des techniques algébriques assez lourdes [11, 12].

On peut montrer que le choix de prédicteurs direct et rétrograde, aussi naturel soit-il, n'est pas nécessairement le mieux adapté à la résolution du problème initial, et qu'un choix différent, conduisant à un algorithme de Schur généralisé, présente les avantages d'obtenir plus simplement la solution sans nécessiter de faire l'hypothèse d'un modèle d'état invariant ou d'un processus observé stationnaire [5].

4.1. ÉQUATIONS DE CHANDRASEKHAR GÉNÉRALISÉES

Nous nous limitons par la suite aux problèmes linéaires de la forme :

$$(4.1) \quad \mathbf{y}_k = \mathbf{H} \mathbf{x}_k + \mathbf{b}_k$$

où \mathbf{b}_k est un bruit blanc centré stationnaire et indépendant de \mathbf{x}_k . La technique utilisée pour développer les équations de Chandrasekhar correspondant à ce problème est tout à fait analogue à celle du chapitre précédent. Elle consiste à introduire un vecteur auxiliaire étendu \mathbf{X} , construit par concaténation de tous les \mathbf{x}_k , afin de pouvoir exploiter simplement les propriétés d'invariance du modèle (4.1) lors du calcul du lisseur ou du prédicteur à un pas optimal [5]. La différence essentielle avec le cas de la régression linéaire est que cette fois la dimension du problème

(c'est-à-dire celle de \mathbf{X}) est supérieure à celle de la matrice d'observation \mathbf{H} . Le contraste s'est inversé.

Le vecteur \mathbf{X} est de dimension nN . Soit \mathbf{H} , la matrice-bloc contenant N blocs de dimension np et construite par concaténation de $i-1$ matrices nulles, de la matrice \mathbf{H} , et de $N-i$ matrices nulles. L'équation (4.1) peut alors s'écrire indifféremment :

$$(4.2) \quad \mathbf{X} = \text{Cte}$$

$$(4.3) \quad \mathbf{y}_k = \mathbf{H}_k \mathbf{X} + \mathbf{b}_k$$

$$(4.4) \quad \mathbf{H}'_{k+1} = \mathbf{D}\mathbf{H}'_k, \quad \mathbf{H}'_k = \mathbf{D}'\mathbf{H}'_{k+1}$$

où \mathbf{D} est la matrice de décalage par bloc dans le cas d'une observation scalaire, et la matrice-bloc de décalage par bloc dans le cas général. Cette relation est utilisée dans la résolution du problème qui est faite par un filtrage de Kalman appliqué à ce modèle qui est dégénéré, car sans dynamique dans l'état.

Le calcul du gain de ce filtre peut être fait de manière rapide en utilisant les propriétés d'invariance par décalage du modèle (4.1) explicitées par (4.4), et la stationnarité du bruit \mathbf{b}_k . L'idée générale est la même que celle utilisée dans le chapitre 2. On ne manipule que les incréments des quantités nominales du filtre de Kalman correspondant. La difficulté provient du fait que le modèle (4.2) à (4.4) n'est pas strictement invariant et qu'on ne peut utiliser les équations de Chandrasekhar usuelles. Il suffit de généraliser la définition des incréments :

$$(4.5) \quad \delta \mathbf{P}_k = \mathbf{P}_{k/k-1} - \mathbf{D}\mathbf{P}_{k-1/k-2}\mathbf{D}'$$

$$(4.6) \quad \delta \mathbf{K}_k = \mathbf{K}_k - \mathbf{D}\mathbf{K}_{k-1}$$

$$(4.7) \quad \delta \mathbf{R}_k^e = \mathbf{R}_k^e - \mathbf{R}_{k-1}^e.$$

Les deux factorisations possibles :

$$(4.8) \quad \delta \mathbf{P}_k = \mathbf{Y}_{k-1} \mathbf{M}_{k-1} \mathbf{Y}'_{k-1}$$

conduisent alors aux algorithmes de Chandrasekhar généralisés [5].

La complexité de ces algorithmes est directement liée à la dimension de la matrice \mathbf{M}_k , c'est-à-dire au rang de l'incrément initial qui s'écrit dans ce cas :

$$(4.9) \quad \delta \mathbf{P}_1 = \mathbf{P}_{1/10} - \mathbf{D}\mathbf{P}_{1/0}\mathbf{D}'.$$

Il est remarquable que ce rang soit justement le rang de déplacement de la matrice de covariance *a priori* du processus à estimer. On retrouve donc bien, sous-jacente, la même mesure de complexité que dans [11].

Ces équations de Chandrasekhar généralisées peuvent recevoir le même genre d'interprétation géométrique que les équations de Chandrasekhar usuelles. Il suffit, pour s'en rendre compte, de les regrouper sous une forme matricielle compacte :

$$(4.10) \quad \begin{bmatrix} \mathbf{R}'_{k+1} & \mathbf{0} \\ \mathbf{K}_{k+1} & \mathbf{Y}_{k+1} \\ \mathbf{0} & -\mathbf{M}'_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}'_k & \mathbf{H}_{k+1} \mathbf{Y}_k \\ \mathbf{D}\mathbf{K}_k & \mathbf{Y}_k \\ \mathbf{Y}'_k \mathbf{H}'_{k+1} & -\mathbf{M}'_{k+1} \end{bmatrix} \times \left[\begin{array}{c|c} \mathbf{I}_1 & -(\mathbf{R}_k^e)^{-1} \mathbf{H}_{k+1} \mathbf{Y}_k \\ \hline -\mathbf{M}_k \mathbf{Y}'_k \mathbf{H}'_{k+1} & \mathbf{I}_2 \end{array} \right].$$

La matrice :

$$(4.11) \quad \Theta_k = \left[\begin{array}{c|c} \mathbf{I}_1 & -(\mathbf{R}_k^e)^{-1} \mathbf{H}_{k+1} \mathbf{Y}_k \\ \hline -\mathbf{M}_k \mathbf{Y}'_k \mathbf{H}'_{k+1} & \mathbf{I}_2 \end{array} \right]$$

est aussi une matrice J-orthogonale [5]. L'analogie avec la forme matricielle compacte de l'algorithme de Levinson généralisé peut être mise en évidence simplement. Le plus simple pour cela est de partir de la forme de ce dernier dans laquelle les dimensions des différents vecteurs ou matrices sont fixes [3]. Soient, en reprenant les notations de [3], \mathbf{R} la matrice $(n+1, n+1)$ normale à inverser, \mathbf{A}_i le vecteur $(n+1)$ obtenu en complétant le prédicteur rétrograde \mathbf{a}_i d'ordre i par des zéros, \mathbf{B}_i la matrice $(n+1, \alpha+1)$ construite à partir du prédicteur rétrograde \mathbf{b}_i , σ_i^2 et Δ_i respectivement l'erreur de prédiction et la matrice d'erreur, \mathbf{K}_i le vecteur $(\alpha+1)$ qui généralise les coefficients de réflexion. L'algorithme de Levinson généralisé [3, 11, 12] peut alors se mettre sous la forme matricielle compacte suivante :

$$(4.12) \quad \left[\begin{array}{c|c} \sigma_{i+1}^2 & \mathbf{0} \\ \mathbf{A}_{i+1} & \mathbf{B}_{i+1} \\ \mathbf{0} & \sigma_{i+1}^2 \Delta_{i+1} \end{array} \right] = \left[\begin{array}{c|c} \sigma_i^2 & -\sigma_i^2 \mathbf{K}_i' \\ \mathbf{D}\mathbf{A}_i & \mathbf{B}_i \\ -\sigma_i^2 \mathbf{K}_i & \sigma_i^2 \Delta_i \end{array} \right] \left[\begin{array}{c|c} \mathbf{I}_1 & \mathbf{K}_i' \\ \hline \Delta_i^{-1} \mathbf{K}_i & \mathbf{I}_{\alpha+1} \end{array} \right],$$

Dans cette expression, la matrice :

$$(4.13) \quad \Psi_i = \left[\begin{array}{c|c} \mathbf{I}_1 & \mathbf{K}_i' \\ \hline \Delta_i^{-1} \mathbf{K}_i & \mathbf{I}_{\alpha+1} \end{array} \right]$$

est telle que :

$$(4.14) \quad \Psi_k' \mathbb{R}_k \Psi_k = \mathbb{R}_{k+1}, \quad \Psi_k \mathbb{R}_{k+1}^{-1} \Psi_k' = \mathbb{R}_k^{-1}$$

avec :

$$(4.15) \quad \mathbb{R}_k \triangleq \left[\begin{array}{c|c} \sigma_i^2 & \mathbf{0} \\ \hline \mathbf{0} & -\sigma_i^2 \Delta_i \end{array} \right].$$

Il s'agit donc, là encore, d'une matrice J-orthogonale. Cette présentation compacte et en dimensions fixes montre bien que la différence principale entre ces deux algorithmes (4.10) et (4.12) n'est pas dans le type de récurrence utilisé, mais plutôt dans la nature des quantités calculées à chaque récurrence. L'algorithme de Chandrasekhar généralisé ne calcule plus les vecteurs (ou les matrices) des prédicteurs direct et rétrograde du processus observé, mais des covariances mutuelles entre le vecteur d'état et les innovations directes et rétrogrades de ce processus. Il s'agit donc d'un algorithme de Schur généralisé.

La résolution du problème initial est obtenue par projection, à partir de la solution du problème auxiliaire étendu [5]. On retrouve ainsi les équations de Friedlander *et al.* [11] d'une manière beaucoup plus simple car on résout en fait directement le problème initial qui est celui de l'estimation du processus \mathbf{x} , au lieu de le résoudre indirectement à partir du problème auxiliaire de la prédiction du processus observé \mathbf{y} .

4.2. EXEMPLES D'APPLICATION

La méthode précédente est bien adaptée aux situations où l'on ne dispose pas de modèle interne ou paramétrique du processus à estimer. Elle a été utilisée dans des problèmes de restauration ou de déconvolution de signaux et d'images [8, 9, 13, 29]. Dans ces applications, le processus à restaurer est simplement modélisé par ses moyenne et covariance *a priori*, et l'ensemble de ses échantillons est concaténé dans un vecteur d'état sans dynamique. La matrice D est la matrice-bloc de décalage par bloc dans le cas de la restauration d'une image observée ligne par ligne. C'est la matrice de décalage ordinaire dans les autres cas. La dimension très importante de ces modèles est compensée par l'emploi des algorithmes rapides décrits plus haut avec, en plus, l'avantage d'obtenir une estimée lissée en un seul balayage des données, alors que la plupart des méthodes de déconvolution d'images à variance minimale ne fournissent qu'une estimée filtrée.

5. Conclusion

Nous avons montré dans cet article que les algorithmes rapides du type Chandrasekhar n'étaient pas fondamentalement différents des autres algorithmes rapides de la famille des moindres carrés. Ils ont des liens étroits avec ceux des types Levinson et Schur, ils sont soumis aux mêmes risques d'instabilité numérique; ils présentent les mêmes difficultés dans le cas d'observations vectorielles. Leur trait caractéristique principal est que ce sont en fait les algorithmes rapides naturels des problèmes d'estimation linéaires et bayésiens. Ceci explique sans doute pourquoi ils ont été assez peu utilisés. Ils offrent cependant plusieurs avantages, parmi lesquels le caractère systématique de leur obtention n'est pas le moindre. Il suffit en effet d'explicitier les propriétés d'invariance que l'on veut utiliser sous la forme d'un modèle d'état pour bénéficier ensuite du caractère canonique des équations de Chandrasekhar. Mais il est important de préciser que ces algorithmes n'exploitent que des propriétés d'invariance et que leur emploi n'a aucune raison d'être limité au cas de signaux stationnaires ou proches de la stationnarité. Il suffit pour s'en convaincre d'examiner l'exemple de l'inversion des matrices browniennes. On sait qu'il existe des algorithmes rapides, et même très rapides, pour les inverser [26, 27]. Or ce problème n'est qu'un aspect de celui de l'estimation d'un processus décrit par le modèle suivant:

$$(5.1) \quad \mathbf{x}_{k+1} = \mathbf{F} \mathbf{x}_k + \mathbf{g} w_k$$

$$(5.2) \quad y_k = \mathbf{h}' \mathbf{x}_k + b_k$$

où w_k et b_k sont deux bruits blancs gaussiens, et où les matrices \mathbf{F} , \mathbf{g} et \mathbf{h} sont ainsi constituées:

$$(5.3) \quad \mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Ce modèle étant invariant, le problème peut être résolu à l'aide d'équations de Chandrasekhar usuelles. La factorisation initiale de l'incrément de covariance est immédiate:

$$(5.4) \quad \delta \mathbf{P}_1 = \mathbf{g} \sigma^2 \mathbf{g}' = \mathbf{Y}_0 \mathbf{M}_0 \mathbf{Y}_0'$$

où σ^2 est la variance de w_k . \mathbf{M}_0 est un scalaire et, comme les matrices \mathbf{F} , \mathbf{g} et \mathbf{h} ne sont formées que de zéros et de uns, on obtient finalement un algorithme très rapide dont la complexité globale est $O(N)$. Ce résultat est d'autant plus remarquable qu'une matrice brownienne a un rang de déplacement maximum!

Manuscrit reçu le 5 mai 1988.

BIBLIOGRAPHIE

- [1] B. D. O. ANDERSON et J. B. MOORE, *Optimal filtering*, Prentice-Hall, 1979.
- [2] L. CHIN, Advances in computational efficiencies of linear filtering, in *Control and Dynamic Systems*, Academic Press, 1983, p. 125-192.
- [3] C. DEMEURE et L. SCHARF, Vector algorithms for computing QR and Cholesky factors of close-to-Toeplitz matrices, *IEEE ICASSP 87*, Dallas, 1987, p. 1851-1854.
- [4] G. DEMOMENT et R. REYNAUD, Fast minimum variance deconvolution, *IEEE Trans.*, ASSP-33, 1985, p. 1324-1326.
- [5] G. DEMOMENT et R. REYNAUD, Équations de Schur et de Chandrasekhar généralisées pour des problèmes généraux d'estimation linéaire, 11^e Colloque GRETSI, Nice, 1987, p. 9-12.
- [6] U. B. DESAI et G. S. SIDHU, A method of orthogonal directions III. Estimation algorithms of Chandrasekhar and Cholesky types for discrete-time nonconstant models, *J. Math. Anal. Appl.*, 61, 1977, p. 797-812.
- [7] J. P. DUGRÉ, L. L. SCHARF et C. GUÉGUEN, Exact likelihood for stationary vector autoregressive moving average processes, *Signal processing*, 11, 1986, p. 105-118.
- [8] DU-XUE-CHENG, Algorithmes rapides pour la restauration d'images dégradées par un système linéaire et par du bruit, *Thèse de Doctorat*, Université de Paris-Sud, Centre d'Orsay, 1986.
- [9] DU-XUE-CHENG, D. SAINT-FÉLIX et G. DEMOMENT, Comparison between a factorization method and a partitioning method to derive invariant Kalman filters for fast image restoration, in *Mathematics in Signal Processing*, T. S. DURRANI et al. éd., Clarendon Press, Oxford, 1987, p. 349-362.
- [10] G. FAVIER, *Filtrage, modélisation et identification des systèmes linéaires stochastiques à temps discret*, CNRS, Paris, 1982.
- [11] B. FRIEDLANDER, T. KAILATH, M. MORF et L. LJUNG, Extended Levinson and Chandrasekhar equations for general discrete-time linear estimation problems, *IEEE Trans.*, AC-23, 1978, p. 653-659.
- [12] B. FRIEDLANDER, M. MORF, T. KAILATH et L. LJUNG, New Inversion Formulas for Matrices Classified in Terms of Their Distance from Toeplitz Matrices, *Lin. Algebra. Applic.*, 27, 1979, p. 31-60.

- [13] Y. GOUSSARD et G. DEMOMENT, Détection-estimation récursive rapide de séquences Bernoulli-gaussiennes. — *Traitement du Signal*, 4, 1987, p. 377-388.
- [14] C. GUÉGUEN, Sur une mesure du caractère non-stationnaire, 9^e Colloque GRETSI, Nice, 1983, p. 27-33.
- [15] C. GUÉGUEN, An introduction to displacement ranks and related fast algorithms, *Lecture notes*, NATO Summer School on Signal Processing, Les Houches, 1985.
- [16] A. HOUACINE, Algorithmes des moindres-carrés récursifs rapides régularisés pour le filtrage et l'analyse spectrale adaptatifs, *Thèse de Doctorat*, Université de Paris-Sud, Centre d'Orsay, 1987.
- [17] A. HOUACINE et G. DEMOMENT, Approche bayésienne de l'analyse spectrale adaptative: modèles AR longs et filtrage de Kalman rapide, *Traitement du Signal*, 4, 1987, pp. 389-397.
- [18] T. KAILATH, Some new algorithms for recursive estimation in constant linear systems, *IEEE Trans.*, IT-19, 1973, p. 750-760.
- [19] T. KAILATH, A. VIEIRA et M. MORF, Orthogonal transformation (square-root) implementations of the generalized Chandrasekhar and generalized Levinson algorithms, *Proc. IRIA 1978 Conf. on Syst. Optimization and Analysis*, BENSOUSSAN et LIONS éd., 1978, p. 81-91.
- [20] T. KAILATH, Some alternatives in recursive estimation, *Int. J. Contr.*, 32, 1980, p. 311-328.
- [21] J. LE ROUX, Sur les algorithmes de factorisation spectrale dans le cas des signaux stationnaires échantillonnés, *Thèse de Doctorat d'État*, Université de Nice, 1985.
- [22] A. LINDQUIST, A new algorithm for optimal filtering of discrete-time stationary processes, *SIAM J. Contr.*, 12, 1974, p. 736-746.
- [23] L. LJUNG et T. KAILATH, Efficient change of initial conditions, dual Chandrasekhar equations, and some applications, *IEEE Trans.*, AC-22, 1977, p. 443-447.
- [24] M. MORF, G. S. SIDHU et T. KAILATH, Some new algorithms for recursive estimation in constant, linear systems, *IEEE Trans.*, AC-19, 1974, p. 315-323.
- [25] M. MORF et T. KAILATH, Square-Root Algorithms for Least-Squares Estimation, *IEEE Trans.*, AC-20, 1975, p. 487-497.
- [26] B. PICINBONO, Fast Algorithms for Brownian Matrices, *IEEE Trans.*, ASSP-31, 1983, p. 512-514.
- [27] B. PICINBONO et M. BÉNIDIR, On a class of signals and matrices giving very fast algorithms for linear problems, *Signal Processing*, 9, 1985, p. 215-224.
- [28] C. M. RADER et A. O. STEINHARDT, Hyperbolic Householder transformation, *IEEE Trans.*, ASSP-34, 1986, p. 1589-1602.
- [29] D. SAINT-FÉLIX, DU-XUE-CHENG et G. DEMOMENT, Filtrés de Kalman 2-D à modèle d'état non-causal pour la restauration d'image, *Traitement du Signal*, 4, 1987, p. 399-410.
- [30] M. VERHAEGEN et P. VAN DOOREN, Numerical aspects of different Kalman filter implementation, *IEEE Trans.*, AC-31, 1986, p. 907-917.
- [31] A. S. WILLSKY, *Digital signal processing and control and estimation theory*, MIT Press, Cambridge, Mass., 1979.