



HAL
open science

Algorithmes adaptatifs d'identification et de reconstruction de processus AR à échantillons manquants

Rawad Zgheib

► **To cite this version:**

Rawad Zgheib. Algorithmes adaptatifs d'identification et de reconstruction de processus AR à échantillons manquants. Mathématiques [math]. Université Paris Sud - Paris XI, 2007. Français. NNT : . tel-00273585

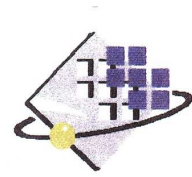
HAL Id: tel-00273585

<https://theses.hal.science/tel-00273585>

Submitted on 15 Apr 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° D'ORDRE :

UNIVERSITÉ PARIS-SUD XI
Faculté des Sciences d'Orsay

THÈSE DE DOCTORAT

Spécialité : physique

École Doctorale « Sciences et Technologies de l'Information,
des Télécommunications et des Systèmes »

Présentée par : Rawad ZGHEIB

**ALGORITHMES ADAPTATIFS D'IDENTIFICATION ET DE
RECONSTRUCTION DE PROCESSUS AR À ÉCHANTILLONS
MANQUANTS**

Soutenue le 19 déc. 2007 devant les membres du jury :

M. ALBERTOS PEREZ	Pedro	Rapporteur
M. FUCHS	Jean-Jacques	Rapporteur
M. BARBARESCO	Frédéric	Examineur
M. BONDON	Pascal	Président
M. FLEURY	Gilles	Directeur de thèse

Remerciements

Je tiens à exprimer, en début de cette thèse, ma reconnaissance à des personnes, qui de près ou de loin, ont contribué à mener à bien ce travail de thèse.

Gilles Fleury, mon directeur de thèse, tant pour ses compétences scientifiques que pour ces qualités humaines. Sa disponibilité, ses conseils avisés et sa patience ont largement contribué à l'aboutissement de ce travail.

Tous les membres du Jury, qui ont consacré une partie de leur temps pour s'intéresser à mon travail de recherche. Je remercie en particulier Pedro Albertos et Jean-Jacques Fuchs pour avoir accepté de rapporter ma thèse et pour leur lecture attentive du manuscrit et leurs remarques précieuses. Je tiens également à remercier Pascal Bondon qui m'a fait l'honneur de présider mon jury de thèse.

Fabienne Suraud, Sophie David et Karine El Rassi, qui ont été successivement secrétaires du département Signaux et Systèmes Électroniques, pour leur sympathie. Elles ont été toujours prêtes pour nous aider dans les démarches administratives.

Toute l'équipe du département Signaux et Systèmes Électroniques pour leur amitié et leur soutien.

Toute l'équipe du CESFO, pour la bonne nourriture qu'ils nous offraient.

Miguel, l'amoureux du CESFO, qui a été mon collègue de bureau pendant les deux premières années, ainsi que Carine et Hassan, mes collègues du bureau actuels, pour leur amitié et la bonne ambiance de travail.

Sylvain, un collègue du département, grâce à qui j'ai amélioré mon français, pour sa sympathie et sa gentillesse.

Tous mes amis et mes proches qui ont été pour moi ma famille en France, pour leur encouragement.

Finalement, un grand merci à mes parents, vous m'avez toujours inculqué la soif du savoir et la recherche de l'excellence. Je vous dédie ce travail.

Résumé

On souhaite reconstruire en ligne des signaux à échantillons manquants en utilisant une approche paramétrique. On propose alors des algorithmes adaptatifs d'identification et de reconstruction de processus AR à échantillons manquants. On s'intéresse premièrement à l'extension des algorithmes de gradient au cas des signaux à échantillons manquants. On propose alors deux alternatives à un algorithme existant fondées sur deux autres prédicteurs. Les algorithmes proposés convergent vers une estimation non biaisée des paramètres. Or les algorithmes de gradient souffrent d'une faible vitesse de convergence. Pour cela, on s'intéresse à l'extension de l'algorithme MCR au cas des signaux à échantillons manquants. On utilise alors l'algorithme MCR pseudo-linéaire pour l'identification conjointement avec un filtre de Kalman pour une prédiction optimale du signal au sens des moindres carrés. L'algorithme résultant permet une identification non biaisée des paramètres. De plus, il est rapide et bien adapté à l'identification de processus non stationnaires. Néanmoins, souhaitant contrôler la stabilité du filtre identifié, on s'intéresse ensuite à une identification fondée sur une structure en treillis du filtre. Ainsi, on propose une extension de l'algorithme de Burg adaptatif au cas des signaux à échantillons manquants, en utilisant pour la prédiction un filtre de Kalman. La stabilité du modèle ainsi identifié est garantie. De plus, l'algorithme s'adapte rapidement aux variations des paramètres. Finalement, on propose d'utiliser les algorithmes proposés dans un système à transmission non uniforme. On obtient ainsi l'amélioration simultanée du RSB et du débit de transmission moyen.

Abstract

We are concerned in online reconstruction of signals subject to missing samples using a parametric approach. We propose adaptive algorithms for identification and reconstruction of AR processes with missing samples. Firstly, we consider the extension of gradient algorithms to the case of signals with missing samples. We propose two alternatives to an existent algorithm based on two other predictors. The proposed algorithms converge toward unbiased estimation of the parameters. However, gradient algorithms suffer from slow convergence. Therefore, we consider the RLS algorithm extension to the case of signals subject to missing samples. We use jointly, the pseudo-linear RLS algorithm for the identification and a Kalman filter for optimal reconstruction of the signal in the least mean square sense. The estimated parameters, using the proposed algorithm, are unbiased. In addition, it is fast and well adapted to the identification of non stationary signals. Nevertheless, looking for the control of the identified filter stability, we propose to identify the signal using the lattice structure of the filter. We propose an extension of the Burg adaptive algorithm to the case of signals subject to missing observations, using a Kalman filter for the prediction. The estimated parameters guarantee the stability of the corresponding filter. In addition, it offers a faster tracking parameter. Finally, we use the proposed algorithms in non uniform transmission systems. We then get the improvement of both the SNR and the average transmission rate.

Table des matières

Remerciements	i
Résumé	ii
Abstract	iv
Table des matières	vi
Table des figures	viii
Liste des tableaux	x
Introduction	1
1 Etat de l'art	5
1.1 Modélisation et estimation paramétrique du signal	5
1.2 Modélisation par un processus AR du signal	6
1.2.1 Définitions	7
1.2.2 Identification du modèle	8
1.3 Traitement paramétrique de signaux à échantillons manquants	12
1.3.1 Méthodes de traitement off-line	12
1.3.2 Méthodes de traitement on-line	19
2 Algorithmes de gradient	27
2.1 Extension des algorithmes de gradient au cas d'échantillons manquants	27
2.1.1 Analyse du critère minimisé	28
2.1.2 Erreur quadratique instantanée	29
2.1.3 Erreur quadratique moyenne (fenêtre croissante)	29
2.1.4 Erreur quadratique sur une fenêtre glissante	30
2.1.5 Résumé global des algorithmes	31
2.1.6 Observations et analyse des performances des algorithmes	32
2.2 Alternatives proposées	32
2.2.1 Approches de prédiction utilisées	32
2.2.2 Algorithmes de gradient utilisant le prédicteur à k -pas	37
2.2.3 Algorithmes de type gradient utilisant le prédicteur à passé incomplet	38
2.3 Simulations et Comparaison des performances	40
2.4 Conclusions	45

3	Algorithme MCR pseudo linéaire	47
3.1	Extension de l'algorithme MCR au cas de signaux à échantillons manquants	47
3.1.1	Régression pseudo-linéaire	47
3.1.2	Algorithme des MCR pseudo-linéaire	49
3.1.3	Observations et analyse des performances	51
3.2	Prédiction en utilisant un filtre de Kalman	51
3.2.1	Représentation d'état d'un processus AR à échantillons manquants	52
3.2.2	Filtre de Kalman	53
3.2.3	Analyse du filtre et conclusions	54
3.3	Algorithme Kalman MCR pseudo-linéaire	56
3.3.1	Description de l'algorithme	56
3.3.2	Analyse de la convergence de l'algorithme	58
3.4	Simulations	60
3.5	Conclusions	67
4	Algorithmes en treillis	69
4.1	Introduction	69
4.2	Identification adaptative de filtres en treillis	70
4.2.1	La structure en treillis du filtre	70
4.2.2	Calcul des coefficients de réflexion : estimation par bloc	71
4.2.3	Identification adaptative du filtre en treillis	73
4.3	Prediction en utilisant un filtre de Kalman	76
4.3.1	Représentation d'état	76
4.3.2	Filtre de Kalman	77
4.4	Algorithme de Burg adaptatif combiné à un filtre de Kalman	77
4.4.1	Estimation optimale des erreurs de prédiction directe et rétrograde	78
4.4.2	Algorithme proposé	79
4.5	Simulations	81
4.6	Conclusion	86
5	Echantillonnage adaptatif, systèmes à transmission non uniforme	89
5.1	Echantillonnage adaptatif	89
5.1.1	Critère fondé sur l'erreur de reconstruction	90
5.1.2	Critère fondé sur l'erreur de quantification	91
5.2	Système à transmission non uniforme	94
5.3	Simulations	96
5.4	Le codage MICDA	103
5.5	Codage MICDA à transmission non uniforme	105
5.5.1	Transmission non uniforme dans un contexte de codage différentiel	105
5.5.2	Description du système à transmission non uniforme	106
5.6	Simulations	107
	Conclusions et perspectives	109
A	Représentation autorégressive du prédicteur à k-pas pour un AR(2)	115
B	Représentation autorégressive du prédicteur à passé incomplet pour un AR(L)	117
	Bibliographie	118

Table des figures

1.1	Modélisation source-filtre d'un signal AR - filtre tout-pôle, RII.	7
1.2	Principe de l'identification - Filtrage adaptatif.	10
2.1	Adaptation du prédicteur à k -pas à la prédiction de signaux à échantillons manquants.	34
2.2	Prédiction d'un processus AR(L) à échantillons manquants utilisant le prédicteur à passé fini incomplet.	35
2.3	Schémas de perte particulier permettant la comparaison des trois prédicteurs utilisés dans ce chapitre.	36
2.4	Principe de l'algorithme LMS-like basé sur la prédiction à k -pas.	37
2.5	Comparaison de l'estimation du paramètre a_1 en utilisant les trois algorithmes.	40
2.6	Estimation du paramètre a_1 après la convergence des trois algorithmes.	41
2.7	Densité de probabilité empirique de l'estimation du paramètre a_1 utilisant les trois algorithmes	42
2.8	EQMR des trois algorithmes en fonction de la probabilité de perte q	43
2.9	Estimation instantanée moyenne du paramètre a_1 pour différentes valeurs de μ grâce à l'algorithme de gradient utilisant le prédicteur à k -pas.	44
2.10	Estimation instantanée moyenne du paramètre a_1 pour différentes valeurs de μ grâce à l'algorithme de gradient utilisant le prédicteur à passé infini incomplet.	45
3.1	Modélisation du système global.	52
3.2	Biais de l'estimation de a_1 en fonction de la probabilité de perte q en utilisant l'algorithme Kalman MCR pseudo-linéaire pour différentes valeurs de λ	61
3.3	Biais de l'estimation de a_1 en utilisant l'algorithme Kalman MCR pseudo-linéaire en fonction du facteur d'oubli λ pour différentes probabilités de perte q	62
3.4	Variance de l'estimée de a_1 par l'algorithme Kalman MCR pseudo-linéaire en fonction du facteur d'oubli λ pour différentes probabilités de perte q	63
3.5	Estimation instantanée moyenne du paramètre $a_1 = 1.5$ en utilisant l'algorithme Kalman MCR pseudo-linéaire pour différentes valeurs du facteur d'oubli pour une probabilité de perte $q = 0.2$	63
3.6	Estimation instantanée moyenne du paramètre $a_1 = 1.5$ pour différentes probabilités de perte en utilisant l'algorithme Kalman MCR pseudo-linéaire avec un oubli constant $\lambda = 0.99$	64
3.7	RSB fourni par l'algorithme Kalman MCR pseudo-linéaire, fonction de la probabilité de perte q	65
3.8	Estimation instantanée moyenne du paramètre a_1 en utilisant les algorithmes MCR pseudo-linéaire, Kalman MCR pseudo-linéaire et l'algorithme du gradient stochastique qui utilise la prédiction à passé incomplet.	66

4.1	Schéma de base d'un filtre tout pôle en treillis.	70
4.2	Estimation instantanée du paramètre a_1 en utilisant les deux algorithmes comparés.	82
4.3	RSB moyen en fonction de la probabilité de perte d'échantillons pour l'expérience 2.	83
4.4	Ecart-type du RSB en fonction de la probabilité de perte d'échantillons pour l'expérience 2.	84
4.5	Reconstruction du signal au début d'une nouvelle note en utilisant les deux algorithmes comparés pour l'expérience 3.	85
4.6	Organigramme de l'algorithme en treillis pour l'identification et la reconstruction de signaux à échantillons manquants.	87
4.7	Organigramme de l'algorithme en treillis pour l'identification et la reconstruction de signaux à échantillons manquants.	88
5.1	Schéma simplifié d'un système de transmission numérique.	92
5.2	Schéma simplifié d'un système de transmission numérique.	97
5.3	Evolution du RSB(dB) en fonction du débit de transmission pour un codage MIC du signal test.	98
5.4	Débit de transmission moyen obtenu en utilisant la méthode A en fonction du nombre de bits B_2 pour RSB = 19.33 dB.	99
5.5	Comparaison des méthodes MIC, A-KMCR et A-KMCRT.	100
5.6	Schéma simplifié d'un codeur décodeur MICDA.	104
5.7	Schéma simplifié d'un codeur décodeur MICDA à transmission non uniforme.	108

Liste des tableaux

2.1	Performances des 3 algorithmes comparés pour la simulation 1.	42
3.1	Comparaison des algorithmes Kalman MCR pseudo-linéaire, MCR pseudo-linéaire et l'algorithme du gradient stochastique qui utilise la prédiction à passé incomplet.	66
4.1	Répartition des RSB fournis par chacun des deux algorithmes comparés pour 1000 générations différentes du schéma de perte.	82
4.2	Comparaison des deux algorithmes pour l'expérience 3	85
5.1	Comparaison des méthodes MIC, B-KMCR et B-KMCRT pour $B_1 = 5$ et $B_2 = 6$	101
5.2	Comparaison des méthodes MIC, B-KMCR et B-KMCRT pour $B_1 = 3$ et $B_2 = 4$	101
5.3	Comparaison des méthodes A-KMCR, A-KMCRT, B-KMCR et B-KMCRT pour le RSB obtenu en 5.1.	102
5.4	Comparaison des méthodes A-KMCR, A-KMCRT, B-KMCR et B-KMCRT pour le RSB obtenu en 5.2.	102
5.5	Comparaison des méthodes MICDA-TNU et MICDA pour un nombre de bits $B = 2$ par échantillon.	107
5.6	Comparaison des méthodes MICDA-TNU et MICDA pour un nombre de bits $B = 3$ par échantillon.	108
5.7	Comparaison des méthodes MICDA-TNU et MICDA pour un nombre de bits $B = 4$ par échantillon.	108
5.8	Comparaison des méthodes MICDA-TNU et MICDA pour un nombre de bits $B = 5$ par échantillon.	109

Introduction

Actuellement, les outils théoriques dans le domaine du traitement numérique du signal sont largement centrés sur les processus à échantillonnage uniforme. Pourtant, il existe de nombreuses applications où les instants de mesure ne sont pas répartis de façon régulière. C'est par exemple le cas des signaux à échantillonnage aléatoire. A titre d'exemple, certains instruments de mesures comme le compteur de particules, la vélocimétrie laser ou l'anémométrie à effet Doppler, réalisent des mesures de grandeurs physiques à des instants aléatoires. D'autre part, les signaux échantillonnés périodiquement mais qui ont subi une perte d'échantillons sont aussi des signaux à échantillonnage non uniforme.

Dans bien des applications telles que les télécommunications numériques, on est confronté au traitement de signaux ayant subi une perte d'échantillons. La perte des échantillons peut être due, par exemple, à des erreurs de transmissions, ou à une indisponibilité de l'information ou un mauvais fonctionnement du système de mesure. La perte d'échantillons peut être aléatoire ou déterministe et en particulier périodique conduisant à un système à sous échantillonnage. En économétrie, par exemple, l'indisponibilité de la bourse les weekends est une cause de manque d'échantillons périodique.

Par ailleurs, dans certaines applications, la perte des échantillons est volontaire : l'échantillonnage est alors adaptatif. Ainsi, les instants d'échantillonnage sont choisis de sorte qu'il soit possible de reconstruire le signal avec une erreur tolérée selon un critère donné. Dans ce cas aussi, on est ramené à traiter des signaux à échantillons manquants. Ceci est le cas, par exemple, en codage de la parole et des images dans le but de minimiser le débit sur un support quelconque (compression de données). Une technique de compression de données peut être fondée sur le principe de ne transmettre un échantillon que lorsque ceci est nécessaire selon un critère donné. Ainsi, on évite de transmettre un échantillon s'il peut être prédit par le récepteur avec une erreur de reconstruction tolérée selon un critère donné. Le pourcentage d'échantillons transmis varie donc en fonction du signal, de l'algorithme de reconstruction utilisé, ainsi que de l'erreur de reconstruction tolérée. D'autre part, le prélèvement d'échantillons peut être minimisé afin d'économiser de l'énergie. Ainsi, par exemple, les systèmes CMOS travaillant sur pile ne consomment de l'énergie que lorsqu'ils sont actifs.

En pratique, le traitement en ligne du signal est requis dans bien des applications. De plus, les signaux traités ne sont pas toujours stationnaires. On souhaite alors reconstruire en ligne un signal pouvant être non stationnaire à échantillons manquants en utilisant une approche paramétrique. Il s'agit donc d'estimer en ligne les paramètres d'un modèle qui caractérise les échantillons observés et de prédire les échantillons manquants en utilisant les paramètres ainsi estimés et les échantillons déjà observés. Nous considérons dans le cadre de ce mémoire le processus AR pour la modélisation des signaux. Ce modèle est classiquement utilisé pour caractériser une gamme très large de signaux rencontrés en pratique (signaux de parole, de vibrations, radar, de texture dans une image, ...). En outre, la prédiction des signaux modélisés par un processus AR est linéaire en les paramètres, ce qui simplifie leur traitement. Par contre, la prédiction de processus AR à

échantillons manquants n'est pas linéaire par rapport paramètres. Ainsi, nous nous proposons dans ce mémoire de développer des algorithmes adaptatifs pour l'identification et la reconstruction de processus AR à échantillons manquants. Les algorithmes développés dans ce mémoire sont indépendants de la nature de la perte subie par le signal. Pour les simulations réalisées dans ce mémoire, la perte utilisée est aléatoire suivant une loi de Bernoulli indépendante du signal.

Le chapitre 1 présente un cadre général de l'étude, suivi d'un état de l'art des différents algorithmes déjà développés pour le traitement des signaux à échantillons manquants. Bien qu'on se soit intéressé au traitement en ligne des signaux, les méthodes à traitement en temps différé du signal seront brièvement présentées. En effet, certaines de ces méthodes seront utiles pour le travail effectué dans le cadre de cette thèse.

L'extension des algorithmes fondés sur le principe du gradient au cas de signaux à échantillons manquants fait l'objet du chapitre 2. L'algorithme *LMS-like* proposé par Mirsaidi *et al.* [53] dans cette optique est décrit et analysé notamment en terme de biais d'identification. Cet algorithme converge vers une estimée biaisée en moyenne des paramètres à cause de l'approche de prédiction utilisée. En effet, la prédiction par régression linéaire modifiée, utilisée dans cet algorithme, n'est pas optimale au sens des moindres carrés. Pour résoudre ce problème de biais d'identification, nous proposons deux alternatives fondées sur deux autres approches de prédictions qui sont le prédicteur à k -pas et le prédicteur optimal de processus à passé infini incomplet proposé par P. Bondon [13]. Les algorithmes résultants permettent une estimation non biaisée des paramètres. De plus, l'algorithme du gradient fondé sur le prédicteur proposé par P. Bondon offre une reconstruction optimale au sens des moindres carrés au dépens d'une plus grande complexité calculatoire.

Cependant, les algorithmes de gradient souffrent d'une faible vitesse de convergence. Ces algorithmes ne sont pas capables de s'adapter rapidement aux variations des paramètres dans le cas de signaux non stationnaires. Pour cela, nous nous intéressons au chapitre 3 à l'extension de l'algorithme MCR au problème d'identification et de reconstruction des signaux à échantillons manquants. Nous commençons alors par décrire et analyser l'algorithme MCR pseudo-linéaire proposé par Sanchis *et al.* [62, 63]. Lorsqu'une prédiction par régression linéaire modifiée est utilisée, cet algorithme converge typiquement vers une estimation biaisée des paramètres pour les mêmes raisons que dans le cas de l'algorithme *LMS-like*. Nous utilisons alors, conjointement avec l'algorithme MCR pseudo-linéaire, un filtre de Kalman pour une prédiction récursive optimale au sens des moindres carrés de processus AR(L) à échantillons manquants. Une analyse de convergence en moyenne et en moyenne quadratique des paramètres estimés en utilisant l'algorithme résultant est réalisée, des expressions asymptotiques du biais et de la variance des paramètres estimés sont établies. Cet algorithme permet une identification non biaisée des paramètres et une reconstruction optimale au sens des moindres carrés du signal. De plus, il est rapide et présente une meilleure vitesse de convergence que les algorithmes de type gradient.

Les algorithmes discutés dans les chapitres 2 et 3 utilisent une représentation directe du filtre transverse. Par conséquent, ils peuvent conduire à l'estimation de paramètres qui correspondent à des filtres tout-pôle instables. Dans la théorie du traitement numérique du signal, plusieurs représentations du filtre sont possibles, particulièrement, la représentation en treillis du filtre. Celle-ci est caractérisée par les coefficients de réflexion. La structure en treillis du filtre est intéressante particulièrement lorsqu'on souhaite contrôler la stabilité du filtre. Pour cela, nous proposons dans le chapitre 4 d'identifier le signal à échantillons manquants en utilisant la structure en treillis du filtre. Les différentes techniques déjà existantes pour l'estimation des coefficients de réflexion à partir des données sont présentées en s'intéressant particulièrement aux algorithmes des moindres carrés récursifs en treillis (MCRT). Dans le but de l'analyse par prédiction linéaire de signaux

non stationnaires, Makhoul *et al.* [49] proposent une méthode générale pour la mise à jour adaptative des coefficients de réflexion. Cette méthode est une version adaptative de l'algorithme de Burg. Les coefficients de réflexion ainsi estimés garantissent la stabilité du filtre correspondant. On propose alors une extension de cet algorithme à l'identification de signaux non stationnaires à échantillons manquants en utilisant un filtre de Kalman pour la prédiction des échantillons manquants. Nous proposons, à notre connaissance, le premier algorithme qui permet le traitement en ligne de signaux à échantillons manquants en utilisant la structure en treillis du filtre. La robustesse à un fort taux de perte et la stabilité du modèle ainsi identifié sont garanties. De plus, l'algorithme s'adapte rapidement aux variations de paramètres. Les performances de ce nouvel algorithme dépassent celles des algorithmes existants et ce d'autant plus que la probabilité de perte est élevée.

Les algorithmes développés dans ce mémoire sont appliqués à la reconstruction de signaux audio à échantillons manquants. Ils montrent de bonnes performances en termes d'erreur quadratique de reconstruction. On propose alors de les utiliser dans un système à transmission non uniforme dans le but de minimiser le nombre d'échantillons transmis. Ceci fait l'objet d'études au chapitre 5. Ainsi, plusieurs méthodes d'échantillonnage adaptatif sont décrites. En combinant les méthodes de transmission non uniforme et les algorithmes adaptatifs de reconstruction, nous proposons des systèmes de codage/décodage à transmission non uniforme. Nous considérons dans un premier temps la transmission non uniforme de signaux codés en MIC. Deux systèmes fondés sur deux méthodes de transmission non uniforme différentes sont alors conçus. Ils permettent une amélioration simultanée du débit moyen de transmission et du rapport signal sur bruit de reconstruction (RSB). Nous nous intéressons par la suite à la transmission non uniforme dans un contexte de codage différentiel. Une méthode de transmission non uniforme des erreurs de prédiction est alors proposée. En combinant cette méthode et les méthodes paramétriques de reconstruction des signaux, un système de codage MICDA à transmission non uniforme (MICDA-TNU) est conçu.

Finalement, nous récapitulons, dans la conclusion, les différentes méthodes développées dans ce mémoire. Différentes perspectives d'amélioration possible ainsi que des nouvelles pistes d'exploration sont mises en relief.

Publications relatives à ce travail

Le travail présenté dans ce mémoire a fait l'objet, pendant le déroulement de la thèse, de deux communications orales dans des conférences internationales avec comité de lecture, d'une communication orale dans un colloque francophone avec comité de lecture et d'un article de revue. La première communication orale, présentée à EUSIPCO 2006, portait sur l'extension des algorithmes de gradient au cas de signaux à échantillons manquants. Le travail présenté dans cette publication est repris, de manière beaucoup plus détaillée, dans le chapitre 2 de ce mémoire. La deuxième, présentée à DSP 2006, concernait l'extension de l'algorithme des moindres carrés récursifs à l'identification de processus AR à échantillons manquants. Une étude théorique concernant cette publication se retrouve dans le chapitre 3 de ce mémoire. Dans la troisième communication orale, présentée à GRETSI 2007, nous proposons une extension des algorithmes d'identification en treillis au cas de signaux à échantillons manquants. L'algorithme proposé dans cette communication est décrit plus en détail dans un article de revue soumis à IEEE transactions on signal processing. Cet article a été accepté pour publication dans la revue IEEE transactions on signal processing avec des révisions mineures obligatoires (*ACCEPTED FOR PUBLICATION WITH MANDATORY MINOR REVISIONS (AQ)*). Le problème d'identification et de

reconstruction adaptatives des signaux en utilisant la structure en treillis du filtre fait l'objet du chapitre 4 de ce mémoire.

R. Zgheib, G. Fleury and E. Lahalle, "New fast recursive algorithms for simultaneous reconstruction and identification of AR processes with missing observations," in the proceedings of the fourteenth European signal processing conference, (EUSIPCO), Florence, Italy, September 2006.

R. Zgheib, G. Fleury and E. Lahalle, "New fast algorithm for simultaneous identification and optimal reconstruction of non stationary AR processes with missing observations," in the proceedings of the IEEE twelveth digital signal processing workshop, (DSP), Wyoming, USA, September 2006.

R. Zgheib, G. Fleury et E. Lahalle, "Identification stable et reconstruction robuste de signaux non stationnaires à échantillons manquants," XXIème colloque GRETSI, Troyes, France, September 2007.

R. Zgheib, G. Fleury et E. Lahalle, "Lattice algorithm for adaptive stable identification and robust reconstruction of non stationary AR processes with missing observations," submitted to IEEE transactions on signal processing, accepted for publication with mandatory minor revisions.

Chapitre 1

Etat de l'art

Le traitement de signaux à échantillonnage non uniforme est un domaine vaste du traitement numérique des signaux. Nous nous intéressons plus particulièrement dans le cadre de cette thèse au traitement paramétrique de signaux pouvant être non stationnaires à échantillons manquants. Les signaux observés sont caractérisés par un modèle paramétrique qu'on se propose d'identifier en temps réel et d'utiliser pour prédire les échantillons manquants. Nous considérons dans le cadre de ce mémoire le processus AR pour la modélisation des signaux, ce choix est justifié au paragraphe 1.2 de ce chapitre.

Ce premier chapitre propose un état de l'art des principales méthodes d'identification et de prédiction de signaux à échantillons manquants. Le principe général des méthodes d'estimation paramétrique est tout d'abord présenté. On s'intéresse par la suite à la modélisation des signaux par un processus AR. Nous présentons brièvement les principales méthodes d'identification de processus AR. Enfin, une bonne partie de ce chapitre est destinée aux méthodes de traitement paramétrique de signaux à échantillons manquants. Bien qu'on se soit intéressé au traitement en ligne des signaux, les méthodes à traitement en temps différé du signal seront brièvement présentées. En effet, certaines de ces méthodes seront utiles pour le travail effectué dans le cadre de cette thèse.

1.1 Modélisation et estimation paramétrique du signal

La modélisation est une description mathématique d'un processus réel construite dans un but précis [70]. Dans le cas du traitement des signaux, la modélisation permet, par exemple, la suppression du bruit, la compression des données, le filtrage, l'interpolation ou la prédiction à court terme de la valeur du signal, etc. Les méthodes d'estimation paramétrique se décomposent essentiellement en quatre étapes :

1. Le choix d'une classe de modèles

C'est une étape fondamentale dans tout problème d'estimation. Le choix d'un modèle paramétrique s'effectue dans une classe de modèles fixée *a priori*. Plusieurs classes de modèles peuvent être discernées. On peut distinguer les modèles linéaires et non linéaires, à temps continu et à temps discret ainsi que déterministes et stochastiques. Le choix entre les différentes classes devra tenir compte du but de la modélisation, de la nature des données à modéliser ainsi que du coût d'investigation en termes de complexité. Nous nous intéressons dans le cadre de cette thèse à la modélisation de processus stochastiques. D'autre part, le problème du traitement de signaux à échantillons manquants est par nature à temps discret. Il existe plusieurs modèles qui permettent de caractériser des processus stochastiques à

temps discret, par exemple, les modèles AR, MA, ARMA, ARX, etc. On s'intéresse au traitement de signaux qui peuvent être caractérisés par un processus AR. Ce choix est justifié dans la section suivante 1.2, où sont aussi présentées quelques généralités sur la modélisation AR.

2. Le choix d'un critère d'optimisation

Une fois la caractérisation effectuée, il s'agit de déterminer le meilleur modèle possible dans la classe choisie, compte tenu du but fixé à la modélisation. Ceci se fait par optimisation d'un critère que nous définissons comme une fonction scalaire des paramètres. Ainsi la valeur optimale des paramètres (qui sera nommée estimée des paramètres) dépendra bien sûr du critère choisi. Les critères quadratiques sont de loin les plus utilisés [70]. Ils consistent en la minimisation par rapport aux paramètres de l'erreur quadratique de prédiction du signal, celle-ci est parfois pondérée. La meilleure estimée des paramètres au sens d'un critère quadratique s'obtient de façon analytique pour des modèles linéaires en les paramètres. Il existe aussi d'autres critères tels que les critères en valeur absolue, le maximum de vraisemblance [70].

3. Le choix d'un algorithme d'optimisation

Il s'agit de l'algorithme utilisé pour estimer les paramètres qui optimisent le critère défini précédemment. Les algorithmes d'optimisation peuvent être de type "on-line" ou "off-line". Dans le cas des algorithmes "off-line", le traitement du signal est réalisé en temps différé pour traiter l'ensemble des données observées en bloc. Ces méthodes ne prennent pas en considération l'évolution au cours du temps des caractéristiques du signal, elles ne sont donc pas appropriées au traitement de signaux non stationnaires. Dans un traitement "on-line", les échantillons du signal sont traités au fur et à mesure dans le temps. Ainsi, les paramètres du modèle sont mis à jour de façon adaptative dans le temps à chaque échantillon traité en ne tenant compte que des échantillons passés. Dans un contexte temps réel, le temps nécessaire au traitement de chaque échantillon devrait être inférieur à la période d'échantillonnage. Ainsi l'algorithme d'optimisation est choisi en fonction des contraintes de l'application et des signaux traités.

4. Le choix de critère d'évaluation des performances

Une fois l'estimation paramétrique effectuée, il s'agit d'évaluer l'incertitude sur les paramètres. Ceci implique l'utilisation de méthodes de Monte-Carlo ou d'estimations statistiques afin d'évaluer typiquement la moyenne et la variance de l'estimation des paramètres. Dans le cas du traitement de signaux à échantillons manquants, l'évaluation des performances de l'algorithme peut porter aussi sur l'erreur quadratique de reconstruction du signal.

Dans le paragraphe suivant, la modélisation de signaux par un processus AR est introduite ainsi que les méthodes classiques d'identification de modèles AR.

1.2 Modélisation par un processus AR du signal

Une grande part de la littérature du traitement numérique du signal est fondée sur une modélisation autorégressive (AR) du signal. Ceci s'explique par le fait que c'est le modèle le plus simple permettant un traitement paramétrique du signal. De plus, la classe des signaux autorégressifs permet de bien modéliser une gamme très large de signaux rencontrés en pratique (signaux de parole, de vibrations, radar, de texture dans une image, ...). Même s'ils ne peuvent que mal représenter des signaux dont le spectre possède des minimums d'énergie très marqués, il suffit en

théorie d'augmenter l'ordre L du modèle pour obtenir un résultat concluant. En outre, d'un point de vue pratique, l'estimation des paramètres AR conduit à la résolution de systèmes linéaires, pour lesquels existent de très nombreuses méthodes de résolution, adaptées aux particularités des systèmes rencontrés. Enfin, le modèle AR peut se justifier par une considération théorique qui consiste à faire appel au principe du Maximum d'Entropie. En effet, c'est le modèle, à minimum d'information, qui respecte la connaissance des (L) premiers coefficients d'autocorrélation d'un signal [19].

1.2.1 Définitions

Un processus x_n est dit autorégressif (AR) d'ordre L , si un échantillon à l'instant n dépend linéairement des L échantillons précédents et d'un bruit blanc additif. Il vérifie l'équation aux différences suivante :

$$x_n = \sum_{i=1}^L a_i x_{n-i} + \varepsilon_n \quad (1.1)$$

où les coefficients $(a_i)_{1 \leq i \leq L}$ sont les paramètres du modèle AR et ε_n est un bruit blanc centré de variance σ_ε^2 . Ce dernier est généralement représenté par un bruit blanc u de variance unité multiplié par l'écart type σ_ε .

La relation précédente peut être présentée sous la forme d'un produit scalaire de deux vecteurs :

$$x_n = \mathbf{a}^\top \mathbf{x}_{n-1} + \varepsilon_n \quad \text{avec} \quad \begin{cases} \mathbf{x}_{n-1} &= [x_{n-1} \dots x_{n-L}]^\top \\ \mathbf{a} &= [a_1 \dots a_L]^\top \end{cases} \quad (1.2)$$

Le vecteur \mathbf{x}_{n-1} ainsi défini s'appelle vecteur de régression.

Un tel processus résulte du filtrage d'un bruit blanc par le filtre de fonction de transfert $H(z) = \sigma_\varepsilon/A(z)$ où $A(z) = 1 - a_1 z^{-1} - \dots - a_L z^{-L}$ représenté par la figure 1.1.

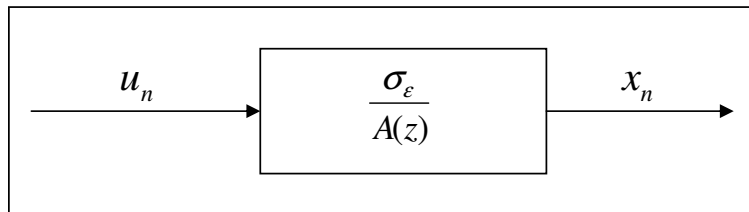


FIG. 1.1 – Modélisation source-filtre d'un signal AR - filtre tout-pôle, RII.

Un tel processus est stationnaire puisqu'il est défini comme un filtré linéaire d'un bruit blanc gaussien, lui-même stationnaire. Il est de moyenne nulle, le bruit générateur l'étant également. Le bruit d'entrée étant décorrélé, la variance du processus AR est donnée par :

$$E\{x_n^2\} = r_x(0) = \sum_{i=1}^{i=L} a_i r_x(i) + \sigma_\varepsilon^2 \quad (1.3)$$

où $r_x(i) = E\{x_n x_{n-i}\}$ est le coefficient d'auto-corrélation d'ordre i du processus $\{x_k\}$.

La densité spectrale de puissance $S_x(f)$ du processus AR $\{x_n\}$ découle directement de sa représentation *source – filtre*. Elle est donnée par :

$$S_x(f) = \frac{\sigma_\varepsilon^2}{|1 - \sum_{i=1}^L a_i e^{-j2\pi i f}|^2} = \frac{\sigma_\varepsilon^2}{A(z) \cdot A(z^{-1})} \Big|_{z=\exp(j2\pi f)} \quad (1.4)$$

Ainsi l'identification des paramètres du modèle AR qui caractérise un signal permet, outre la prédiction du signal, une analyse spectrale du signal.

Le modèle ainsi défini permet de caractériser des signaux stationnaires. Dans le cas de processus non stationnaires, les paramètres du modèle AR évoluent dans le temps. Grenier [30] montre qu'il est possible d'obtenir une représentation autorégressive d'ordre fini du processus non stationnaire donnée par l'équation suivante :

$$x_n = \sum_{i=1}^L a_{i,n-i} x_{n-i} + \varepsilon_n \quad (1.5)$$

où $a_{i,n}$ est la valeur du paramètre a_i à l'instant n .

Dans ce qui suit, nous introduisons le problème d'identification de modèles AR en s'intéressant particulièrement au problème d'identification de signaux non stationnaires.

1.2.2 Identification du modèle

Pour un modèle AR d'ordre L , le prédicteur optimal au sens des moindres carrés d'un échantillon par rapport à son passé complet (sans échantillons manquants) est donné par :

$$\hat{x}_n = \sum_{i=1}^L a_i x_{n-i} = \mathbf{a}^\top \mathbf{x}_{n-1} \quad (1.6)$$

L'erreur de prédiction à l'instant n s'exprime donc par :

$$e_n = x_n - \hat{x}_n = x_n - \sum_{i=1}^L a_i x_{n-i} \quad (1.7)$$

Afin d'identifier les paramètres du modèle AR, le critère usuel à minimiser est l'erreur quadratique moyenne de prédiction. Il s'exprime par :

$$J = E \{e_n^2\} = r_x(0) - 2\mathbf{a}^\top \mathbf{r}_x + \mathbf{a}^\top R_x \mathbf{a} \quad (1.8)$$

$$\text{où } R_x = \begin{bmatrix} r_x(0) & r_x(1) & \dots & r_x(L-1) \\ r_x(1) & r_x(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_x(1) \\ r_x(L-1) & \dots & r_x(1) & r_x(0) \end{bmatrix} \quad \text{et } \mathbf{r}_x = \begin{bmatrix} r_x(1) \\ \vdots \\ \vdots \\ r_x(L) \end{bmatrix}$$

Cette approche est équivalente au maximum de vraisemblance compte tenu de la modélisation du bruit [14].

Comme il a été évoqué ci-dessus, l'estimation des paramètres du modèle peut se faire de deux façons différentes, soit de façon différée, soit de façon récursive sur le temps.

Méthodes d'identification "off-line"

Les techniques d'identification "off-line" permettent l'estimation des paramètres AR en temps différé à partir d'un bloc de données. Dans ce cas, l'ensemble des paramètres optimaux qui minimisent l'erreur quadratique moyenne de prédiction sont solutions des équations de Yule-Walker, dites équations normales :

$$R_x \mathbf{a} = \mathbf{r}_x \quad (1.9)$$

où la matrice R_x et le vecteur \mathbf{r}_x sont définis dans l'équation (1.8).

En pratique, les espérances mathématiques sont remplacées par des moyennes temporelles. Le critère réellement minimisé est alors la somme des erreurs quadratiques de prédiction. Sa minimisation conduit à résoudre les équations normales où les coefficients d'autocorrélation sont remplacés par leurs estimées données par :

$$\hat{r}_x(i) = \frac{1}{N-i} \sum_k x_k x_{k-i} \quad (1.10)$$

Cette méthode s'appelle la méthode d'autocorrélation. L'algorithme de Durbin-Levinson [23, 40] permet de résoudre les équations normales de façon récursive sur l'ordre. Cet algorithme utilise la structure Toeplitz symétrique de la matrice d'autocorrélation R_x , il nécessite $O(N^2)$ opérations où N est la longueur du signal à identifier.

Dans certains cas les signaux non stationnaires ont un comportement quasi-stationnaire, ils peuvent être considérés comme localement stationnaires. Dans le cas de la parole par exemple, les plages de stationnarité sont de l'ordre de 25 ms [59]. Dans ce cas, les techniques d'identification "off-line" de processus AR s'appliquent sur les tranches stationnaires du signal. Cette approche, dite LPC, est utilisée par exemple dans les techniques d'analyse et de synthèse de la parole. Cette longueur de tranche de stationnarité permet de modéliser correctement la plupart des phonèmes excepté les phonèmes à variations brusques tels que les plosives [21, 32]. La réduction de la largeur de la zone de stationnarité cause une mauvaise estimation spectrale et la mise à jour plus fréquente du modèle cause une augmentation du nombre de valeurs qui caractérisent le signal. Pour cela, d'autres approches utilisent une décomposition linéaire des paramètres dépendant du temps sur une base de fonctions temporelles déterministes [21, 31, 32]. Ainsi, les paramètres AR s'expriment sous la forme suivante :

$$a_{i,n} = \sum_{j=0}^m a_{ij} f_{j,n} \quad (1.11)$$

où a_{ij} sont des coefficients constants, $f_{0,n}, \dots, f_{m,n}$ forment une base de fonctions temporelles déterministes. Cet ensemble de fonctions est souvent choisi comme une base des polynômes de Legendre. Les simulations montrent que 5 fonctions orthogonales sont suffisantes pour représenter les variations temporelles des paramètres AR qui modélisent un signal de parole [21]. L'identification du modèle consiste alors à estimer les coefficients constants a_{ij} . Ceci se fait par application de l'algorithme de Durbin-Levinson sur le vecteur résultant de la projection du processus scalaire non stationnaire sur la base de fonctions temporelles. Ainsi une tranche du signal est caractérisée par un modèle AR variable dans le temps caractérisé par les coefficients a_{ij} constants sur cette tranche. Cette technique est avantageuse en codage et synthèse de la parole utilisant une approche de segmentation du signal. Ainsi, les coefficients a_{ij} permettent de caractériser tout une syllabe d'une durée de 200 ms. Par conséquent, 250 valeurs sont nécessaires pour coder une seconde d'un signal de parole au lieu de 400 valeurs dans l'approche LPC classique (10 paramètres mis à jour tous les 25 ms) [31, 32].

Méthodes d'identification "on-line"

On s'intéresse dans le cadre de travail au traitement en ligne des signaux non stationnaires. Ainsi l'estimation des paramètres est adaptative, chaque échantillon étant traité dès qu'il est disponible. Ceci permet de suivre l'évolution des paramètres du modèle au cours du temps. L'estimation du vecteur des paramètres autoregressifs à un instant n sera notée par $\hat{\mathbf{a}}_n = [\hat{a}_{1,n} \dots \hat{a}_{L,n}]^\top$. Le principe général d'un problème d'identification adaptatif est représenté dans la figure 1.2.

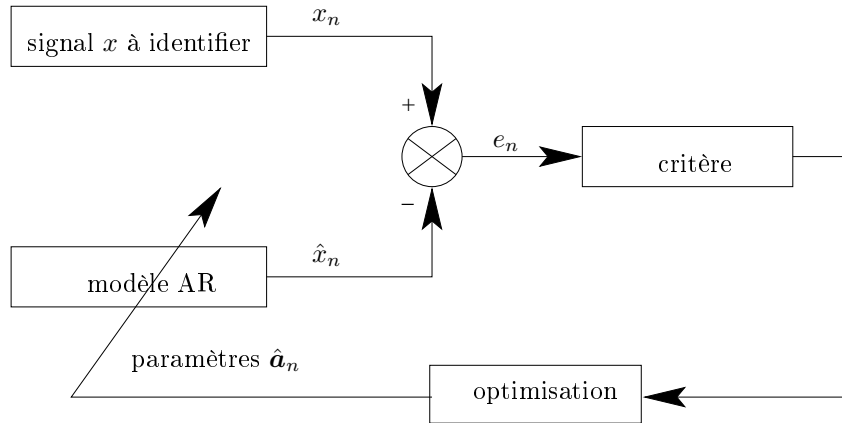


FIG. 1.2 – Principe de l'identification - Filtrage adaptatif.

La fonction de coût est optimisée à l'aide d'algorithmes implémentant des méthodes telles que celles fondées sur le gradient ou celle des moindres carrés récurrents [9, 44].

1. **La méthode du gradient** Elle consiste à minimiser récursivement un critère en se dirigeant dans la direction opposée au gradient de celui-ci, et ce avec un facteur homothétique μ fixé au préalable. Les paramètres sont alors mis à jour à l'instant $n+1$ par une expression de la forme suivante :

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n - \mu \left. \frac{\partial J}{\partial \mathbf{a}} \right|_{\mathbf{a}=\hat{\mathbf{a}}_n} . \quad (1.12)$$

Ceci nécessite donc le calcul analytique du gradient de la fonction de coût. En particulier, dans le cas de l'identification d'un modèle AR en minimisant le critère donné par (1.8), le gradient du critère s'écrit :

$$\frac{\partial J}{\partial \mathbf{a}} = -2\mathbf{r}_x + 2R_x \mathbf{a} \quad (1.13)$$

L'annulation de ce gradient nous ramène à la résolution des équations de Yule Walker (1.9). Ici, la minimisation du critère par la méthode du gradient s'exprime par l'itération suivante :

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n + 2\mu (\mathbf{r}_x - R_x \hat{\mathbf{a}}_n) \quad (1.14)$$

Ceci nécessite donc la connaissance des fonctions d'autocorrélation qui ne sont en pratique pas connues. L'algorithme du gradient stochastique est une solution à ce problème [45]. Le critère minimisé par cet algorithme est l'erreur quadratique de prédiction instantanée correspondant à la réalisation aléatoire courante (d'où son nom de stochastique). Cet algorithme est très répandu car le plus simple à mettre en œuvre [45]. Il existe plusieurs variantes à cet algorithme, elles se distinguent par l'approximation du critère. Ainsi, la moyenne quadratique de l'erreur de prédiction peut être estimée à chaque instant en considérant toutes les erreurs précédentes (fenêtre croissante). Dans le cas de signaux non

stationnaires, la moyenne de l'erreur quadratique de prédiction est estimée sur un horizon fini (fenêtre glissante) d'une durée fonction du degré de non stationnarité du signal. L'algorithme du gradient stochastique correspond à une fenêtre glissante de longueur 1. La méthode du gradient présente une faiblesse en termes de vitesse de convergence. En effet, le temps de convergence est fonction du pas μ ainsi que de l'ordre du modèle et des valeurs propres de la matrice d'autocorrélation R_x (et donc des paramètres du modèle à identifier) [45]. Ainsi, soit, $\tilde{\mathbf{a}}_k = \hat{\mathbf{a}}_k - \mathbf{a}$, le vecteur de déviation des paramètres estimés par rapport à l'optimalité à l'instant k . Si le temps de convergence de l'algorithme est défini comme étant le nombre d'itérations nécessaires pour que $\|E\{\tilde{\mathbf{a}}_k\}\| \leq \delta$, le temps de convergence moyen de l'algorithme du gradient stochastique s'exprime par [25] :

$$\tau_{\text{LMS}} = \frac{\ln(1/\delta)}{\mu L} \sum_{i=1}^L \frac{1}{\lambda_i} \quad (1.15)$$

où λ_i sont les valeurs propres de la matrice de covariance du signal et δ est l'amplitude tolérée du vecteur de déviation moyen.

2. **L'algorithme des moindres carrés récurrents (MCR)** Il permet de résoudre de façon adaptative un problème de moindres carrés. Ainsi, à l'instant n , le critère minimisé est l'erreur quadratique de prédiction accumulée depuis l'indice de départ (fenêtre croissante). Une pondération des erreurs de prédictions est introduite dans le critère à minimiser pour donner à l'algorithme une capacité d'oubli donc de poursuite dans un contexte non stationnaire. La fonction de coût minimisée à chaque instant n est donc de la forme :

$$J_n = \sum_{k=L+1}^n \lambda^{n-k} \left[x_k - \mathbf{a}_n^\top \mathbf{x}_{k-1} \right]^2 \quad (1.16)$$

où $0 \leq \lambda \leq 1$ est un facteur d'oubli. Lorsque $\lambda = 1$ on retrouve le critère minimisé par les moindres carrés classiques. Le principe est de calculer à chaque instant n les paramètres optimaux $\hat{\mathbf{a}}_n$ récursivement par mise à jour de $\hat{\mathbf{a}}_{n-1}$ tout en calculant de manière récursive la matrice G_n qui est la matrice inverse de $\hat{R}_{x,n} = \sum_{k=L+1}^n \lambda^{n-k} \mathbf{x}_{k-1}^\top \mathbf{x}_{k-1}$. Ceci se fait en exploitant l'aspect récursif sur le temps de la matrice $\hat{R}_{x,n}$ et en utilisant le lemme d'inversion matricielle. L'algorithme MCR se résume à un instant $n + 1$ aux équations récursives suivantes [9, 44, 70] :

$$\Psi_{n+1} = \mathbf{x}_n = [x_n \dots x_{n-L+1}]^\top, \quad (1.17a)$$

$$\hat{x}_{n+1} = \Psi_{n+1}^\top \hat{\mathbf{a}}_n, \quad (1.17b)$$

$$\gamma_{n+1} = \frac{G_n \Psi_{n+1}}{\lambda + \Psi_{n+1}^\top G_n \Psi_{n+1}}, \quad (1.17c)$$

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n + \gamma_{n+1} (\mathbf{x}_{n+1} - \hat{x}_{n+1}), \quad (1.17d)$$

$$G_{n+1} = \frac{1}{\lambda} (I_d - \gamma_{n+1} \Psi_{n+1}^\top) G_n \quad (1.17e)$$

où I_d est la matrice identité et le vecteur γ_{n+1} est le gain d'adaptation de l'algorithme MCR à l'instant n .

Bien que simple, cet algorithme est plus complexe que l'algorithme du gradient stochastique. Par contre, cet algorithme offre une convergence rapide et une adaptation rapide aux variations de paramètres dans le cas non stationnaire. Le temps de convergence de l'algorithme dépend bien sûr du choix du facteur d'oubli λ . Ainsi la vitesse de convergence croît

lorsque λ décroît, ceci se fait au détriment d'une plus forte variance des paramètres estimés. Le choix de λ est donc un bon compromis entre la vitesse de convergence et la variance des paramètres estimés. De plus, contrairement à l'algorithme du gradient stochastique, les valeurs propres de la matrice d'autocorrélation R_x n'interviennent pas dans la constante de temps de l'algorithme MCR, τ_{MCR} [45]. Cette dernière est donnée par [9, 45] :

$$\tau_{\text{MCR}} = \frac{1}{1 - \lambda}. \quad (1.18)$$

D'où l'importance de cet algorithme pour l'identification de signaux non stationnaires.

Les approches présentées jusque là concernent le problème d'identification de processus AR échantillonnés périodiquement sans échantillons manquants. Elles ne sont pas appropriées au traitement de signaux à échantillons manquants. Ainsi, par exemple, le prédicteur linéaire optimal (1.6) n'est pas approprié au cas de signaux à échantillons manquants étant donné que certains échantillons du vecteur régresseur \mathbf{x}_{n-1} peuvent être manquants. Plusieurs méthodes ont déjà été développées pour le traitement de signaux à échantillons manquants [37, 53, 63, 69]. Toutefois, ce problème reste encore imparfaitement résolu. On présente dans ce qui suit un résumé des principales méthodes de traitement paramétrique des signaux à échantillons manquants.

1.3 Traitement paramétrique de signaux à échantillons manquants

Plusieurs méthodes ont déjà été développées pour le traitement numérique des signaux à échantillons manquants. On propose de les classer suivant le type de traitement : "on-line" ou "off-line". Dans chacune de ces deux classes, les différentes méthodes sont réparties suivant les approches qu'elles utilisent et les buts pour lesquels elles ont été développées. Bien que l'on se soit intéressé, dans le cadre de cette thèse, au traitement "on-line" des signaux non stationnaires, on commence par présenter dans ce qui suit un résumé des principales méthodes "off-line" de traitement de signaux à échantillons manquants. En effet, certaines de ces méthodes, ainsi que des résultats obtenus pour leur développement, peuvent servir au développement de méthodes "on-line" en les rendant récursives sur le temps. De plus, les performances des méthodes développées dans ce mémoire seront comparées à celles de certaines méthodes "off-line". Le résumé des méthodes "off-line" est suivi par un résumé des méthodes existantes "on-line".

1.3.1 Méthodes de traitement off-line

Les méthodes de traitement off-line sont conçues dans le but de l'identification ou de la reconstruction des signaux à échantillons manquants à partir d'un ensemble d'échantillons disponibles. Se situant dans un contexte de traitement paramétrique des signaux, les méthodes de reconstruction des signaux impliquent une estimation des paramètres du modèle qui caractérisent le signal.

Méthode du maximum de vraisemblance

La méthode du maximum de vraisemblance repose sur un modèle probabiliste du signal. Cette approche permet d'estimer les paramètres en tenant compte des informations disponibles sur la nature du bruit d'observation ou de l'erreur de prédiction (dans le cas des processus AR ou ARMA). Dans le cas de processus ARMA (et donc en particulier de processus AR), l'erreur de modélisation qui est l'innovation du processus est un bruit blanc gaussien. Ainsi chaque

échantillon x_n de la série temporelle est supposé distribué suivant une loi $\mathcal{N}(\hat{x}_n, \sigma_i^2)$. On en déduit que la vraisemblance d'un processus gaussien x_1, \dots, x_N de moyenne nulle s'écrit :

$$V = \prod_{i=1}^N (2\pi\sigma_i^2)^{(-1/2)} \exp\left(-\frac{(x_i - \hat{x}_i)^2}{2\sigma_i^2}\right) \quad (1.19)$$

En pratique, le logarithme de la fonction de vraisemblance est maximisé. Dans le cas d'un processus gaussien, cette approche est équivalente à l'estimation par les moindres carrés.

La fonction de vraisemblance d'un processus ARMA peut être calculée exactement et de façon récursive en se basant sur une représentation d'état markovienne et en utilisant un filtre de Kalman. Cette approche implique la manipulation de matrices de dimension l'ordre du modèle. Il est à noter que pour un calcul exact de la fonction de vraisemblance, une initialisation convenable de la matrice de covariance de l'erreur d'estimation est nécessaire. Le calcul de l'état initial de cette matrice est présenté dans [37]. Des algorithmes d'optimisation non linéaire sont par la suite utilisés pour l'estimation des paramètres au sens du maximum de vraisemblance. Un rappel de cette méthode est présenté dans [37]. Dans cet article, Jones propose une extension de cet algorithme au problème d'identification de modèles ARMA à partir de séries temporelles à échantillons manquants. Il propose de ne pas filtrer l'estimation de l'état par le filtre de Kalman lorsqu'un échantillon est manquant et par suite la matrice de covariance est conservée. Cet algorithme permet une identification non biaisée de modèles ARMA, par contre la maximisation de la fonction de vraisemblance nécessite des algorithmes numériques d'optimisation non linéaire qui sont dans ce cas très coûteux en calcul.

Dans [33], Isaksson propose une alternative à l'algorithme précédent permettant la maximisation de la fonction de vraisemblance dans le cas de signaux à échantillons manquants. Cette méthode est basée sur le principe de l'algorithme EM (proposé par Dempster *et al.* [22]). Il permet l'estimation au sens du maximum de vraisemblance des paramètres d'un modèle probabiliste dépendant de variables sous jacentes non observables. Il consiste en deux étapes principales : l'étape E (Expectation) suivie d'une étape M (Maximization). Ces deux étapes sont alternées jusqu'à la convergence. Durant l'étape E, l'espérance par rapport aux données sous jacentes de la fonction de vraisemblance est calculée en tenant compte de l'estimation courante des paramètres. L'espérance est conditionnée aux données observées. L'étape M permet d'obtenir une nouvelle estimée des paramètres en maximisant la vraisemblance trouvée à l'étape E. On utilise ensuite les paramètres trouvés en M comme point de départ d'une nouvelle phase d'évaluation de l'espérance. L'article [56] est une synthèse de l'algorithme EM appliqué aux problèmes d'estimation en traitement du signal. Isaksson [33] propose donc d'appliquer cet algorithme au problème d'identification de modèles ARX à partir de signaux à échantillons manquants. Une itération n de l'algorithme dans le cas d'identification d'un modèle AR se résume aux deux étapes suivantes :

L'étape E Elle consiste à calculer :

$$Q\left(\mathbf{a}, \hat{\mathbf{a}}^{(n)}\right) = \mathbb{E} \left\{ \frac{1}{N} \sum_{k=1}^N (x_k - \mathbf{x}_{k-1}^\top \mathbf{a})^2 \mid \mathbf{x}^d, \hat{\mathbf{a}}^{(n)} \right\}. \quad (1.20)$$

où $\hat{\mathbf{a}}^{(n)}$ désigne l'estimée des paramètres obtenue à l'itération n et \mathbf{x}^d est le vecteur constitué des échantillons observés du processus $\{x_k\}$.

L'étape M Elle consiste à maximiser la log vraisemblance, donc à minimiser $Q(\mathbf{a}, \hat{\mathbf{a}}^{(n)})$ par rapport aux paramètres :

$$\hat{\mathbf{a}}^{(n+1)} = \left(\sum_{k=1}^N \mathbb{E}\{\mathbf{x}_{k-1} \mathbf{x}_{k-1}^\top\} \right)^{-1} \sum_{k=1}^N \mathbb{E}\{\mathbf{x}_{k-1} \mathbf{x}_k\}. \quad (1.21)$$

Pour le calcul des espérances mathématiques conditionnelles figurant dans les équations précédentes, Isaksson propose d'utiliser un filtre de Kalman [33]. Cet algorithme fournit une estimation au sens du maximum de vraisemblance des paramètres du modèle et offre des performances similaires à celles obtenues grâce à l'algorithme proposé par Jones [37]. En revanche, cet algorithme est plus rapide.

Identification basée sur l'estimation des fonctions de covariance

La méthode proposée par Jones [37] utilise directement les données disponibles pour l'identification du modèle ARMA. Au contraire, Porat *et al.* [58] proposent d'utiliser les échantillons disponibles pour estimer les fonctions d'autocorrélation qui seront utilisées pour estimer les paramètres du modèle ARMA. Ainsi la complexité de calcul est considérablement réduite [58].

Pour l'estimation des fonctions d'autocorrélation, ils se basent sur une définition introduite par Parzen [57]. En effet, Parzen s'intéresse à l'estimation des fonctions d'autocorrélation de signaux ayant subi une modulation d'amplitude. Il introduit une fonction de modulation d'amplitude c_n pour laquelle la fonction $r_c(l)$ définie par l'équation (1.22) existe pour tout l .

$$r_c(l) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=l+1}^N c_n c_{n-l}, \quad l \geq 0 \quad (1.22)$$

Soit $\{y_n\}$ le processus résultant de la modulation en amplitude de $\{x_n\}$ par la fonction c_n , il est défini par :

$$y_n = c_n x_n, \quad \text{pour tout instant } n \quad (1.23)$$

Parzen démontre que $\hat{r}_x(l)$ donné par :

$$\hat{r}_x(l) = \frac{\frac{1}{N} \sum_{n=l+1}^N y_n y_{n-l}}{r_c(l)} \quad (1.24)$$

est une estimée consistante de la fonction d'autocorrélation d'ordre l de $\{x_n\}$, à condition que $r_c(l) \neq 0$.

En particulier, lorsque c_n est binaire, définie comme suit :

$$c_n = \begin{cases} 1 & \text{si l'échantillon } x_n \text{ est disponible} \\ 0 & \text{si l'échantillon } x_n \text{ est manquant,} \end{cases} \quad (1.25)$$

le processus y_n peut être considéré comme l'observation du processus x_n à échantillons manquants. Dans ce cas, lorsque l'on connaît *a posteriori* les instants d'échantillons manquants, l'estimée de la fonction d'autocorrélation d'ordre l du processus $\{x_n\}$ s'exprime par :

$$\hat{r}_x(l) = \frac{\frac{1}{N} \sum_{n=l+1}^N y_n y_{n-l}}{\frac{1}{N} \sum_{n=l+1}^N c_n c_{n-l}} \quad (1.26)$$

Si pour un retard l donné, le dénominateur est nul, $\hat{r}_x(l)$ est considérée nulle. Ceci revient tout simplement à utiliser pour l'estimation de $\hat{r}_x(l)$ les couples (x_n, x_{n-l}) disponibles.

Pour une série temporelle stationnaire, la densité spectrale de puissance est donné par :

$$S_x(f) = \sum_{l=-\infty}^{\infty} r_x(|l|) e^{-j2\pi lf}. \quad (1.27)$$

On sait de plus que pour un processus AR, celle-ci s'exprime par l'équation (1.4). Ceci permet d'obtenir une relation entre les fonctions d'autocorrélation du signal et les paramètres du modèle à identifier de la forme :

$$\frac{D(z)}{A(z)} = \frac{d_0 + d_1 z^{-1} + \dots + d_L z^{-L}}{1 - a_1 z^{-1} - \dots - a_L z^{-L}} = \frac{1}{2} r_x(0) + \sum_{l=1}^{\infty} r_x(l) z^{-l} \quad (1.28)$$

où le polynôme $D(z)$ vérifie $D(z)A(z^{-1}) + D(z^{-1})A(z) = \sigma_\epsilon^2$.

Porat *et al.* [58] recherchent des polynômes $A(z)$ et $D(z)$ de telle façon que les fonctions d'autocorrélation obtenues grâce à la relation (1.28) se rapprochent au mieux de leurs estimées obtenues par (1.26) et ceci selon le critère suivant :

$$V(a_1, \dots, a_L, d_0, d_1, \dots, d_L) = \frac{1}{2} \sum_{l=0}^{N_r} \frac{N_l}{N} [r_x(l) - \hat{r}_x(l)]^2. \quad (1.29)$$

où N_l désigne le nombre de paires d'échantillons disponibles séparés d'un retard l , i.e. $N_l = \sum_{n=l+1}^N c_n c_{n-l}$. N_r est le nombre de fonctions d'autocorrélation nécessaire à l'obtention d'une bonne estimation des paramètres sans avoir à calculer toutes les fonctions d'autocorrélation.

La minimisation de la fonction de coût scalaire (1.29) nécessite le calcul de ses dérivées partielles par rapport aux paramètres ainsi que celles des fonctions d'autocorrélation. Le principe de l'optimisation de (1.29) ainsi que les détails du calcul des différentes dérivées partielles sont explicitées dans [58].

Dans le même but d'identification d'un modèle ARMA grâce à l'estimation des fonctions d'autocorrélation, Rosen *et al.* [60] proposent de minimiser la fonction de coût suivante :

$$V(\hat{\mathbf{a}}, \hat{\mathbf{r}}_x) = [\mathbf{r}_x(\hat{\mathbf{a}}) - \hat{\mathbf{r}}_x]^\top W(\hat{\mathbf{a}}) [\mathbf{r}_x(\hat{\mathbf{a}}) - \hat{\mathbf{r}}_x]. \quad (1.30)$$

où $\mathbf{r}_x(\hat{\mathbf{a}})$ est le vecteur constitué des N_r premières fonctions d'autocorrélation calculées à partir de l'estimée du vecteur des paramètres $\hat{\mathbf{a}}$. $\hat{\mathbf{r}}_x$ est le vecteur constitué des N_r premières fonctions d'autocorrélation estimées grâce à l'équation (1.26). Afin de minimiser asymptotiquement la variance des paramètres estimés, la matrice de pondération $W(\hat{\mathbf{a}})$ est choisie comme suit [60] :

$$W(\hat{\mathbf{a}}) = U^{-1}(\hat{\mathbf{a}}) \quad \text{où} \quad U(\hat{\mathbf{a}}) = \lim_{T \rightarrow \infty} T \mathbf{cov}\{\hat{\mathbf{r}}_x\} \quad (1.31)$$

où $\mathbf{cov}\{\hat{\mathbf{r}}_x\}$ est la fonction de covariance de $\hat{\mathbf{r}}_x$. L'algorithme ainsi obtenu est asymptotiquement optimal au sens de la variance minimale de l'erreur d'estimation des paramètres.

Les méthodes présentées dans cette partie permettent une identification non biaisée d'un processus stationnaire. Elles n'impliquent pas de prédiction des échantillons manquants, seules les données disponibles sont utilisées. Dans ce qui suit, des méthodes d'identification et de reconstruction de signaux à échantillons manquants par la méthode des moindres carrés sont présentées. La fonction de coût à minimiser par rapport aux paramètres du modèle est fonction de l'erreur quadratique de prédiction du signal.

Reconstruction et Identification par la méthode des moindres carrés

Comme il l'a été évoqué précédemment, les paramètres optimaux au sens des moindres carrés d'un processus AR(L) vérifient les équations normales (1.9). En pratique, pour des signaux de longueur finie, l'estimée des paramètres AR au sens des moindres carrés est solution de :

$$\left(\sum_{k=1}^N \mathbf{x}_{k-1} \mathbf{x}_{k-1}^\top \right) \hat{\mathbf{a}} = \sum_{k=1}^N \mathbf{x}_{k-1} x_k \quad (1.32)$$

où \mathbf{x}_{n-1} est le vecteur régresseur à l'instant n .

Lorsque des échantillons sont manquants, la solution la plus simple est de ne considérer dans la somme que les instants k pour lesquels \mathbf{x}_{k-1} et x_k sont disponibles. Par contre, pour certains schémas de perte ou lorsque le nombre d'échantillons manquants est élevé, il devient rare d'avoir \mathbf{x}_{k-1} et x_k disponibles à la fois. Ceci conduit à une mauvaise estimation des paramètres.

Une autre alternative serait d'estimer les L premières fonctions d'autocorrélation du signal à échantillons manquants grâce à l'équation (1.26), et d'utiliser ces estimées pour résoudre les équations de Yule-Walker (1.9). Cette méthode dépend fortement du schéma de perte. Ainsi, certaines fonctions d'autocorrélation peuvent être estimées à partir d'un faible nombre de données. Ceci conduit à une grande variance des paramètres estimés.

Janssen *et al.* [36] proposent un algorithme itératif pour la reconstruction de séries chronologiques à échantillons manquants pouvant être modélisés par des processus AR d'ordre L . Ils supposent que la perte des échantillons est aléatoire. Elle se produit à des instants connus n_1, \dots, n_M , tels que $1 < L+1 \leq n_1 < \dots < n_M \leq N-L-1$ où M est le nombre des échantillons manquants et N la taille du signal à analyser. Il s'agit donc de prédire $\mathbf{s} = [x_{n_1}, \dots, x_{n_M}]^\top$ le vecteur des échantillons manquants ainsi que d'estimer \mathbf{a} , le vecteur des paramètres AR. Pour cela, ils proposent de minimiser par rapport aux paramètres et aux données manquantes la somme des erreurs quadratiques de prédiction donnée par :

$$Q(\mathbf{a}, \mathbf{s}) = \sum_{k=L+1}^N e_k^2 = \sum_{k=L+1}^N \left| x_k - \sum_{i=1}^L a_i x_{k-i} \right|^2. \quad (1.33)$$

La minimisation de cette fonction par rapport aux paramètres et aux données manquantes simultanément n'est pas évidente du fait de la présence de termes d'ordre 4 tel que $a_1^2 x_{n_M}^2$. Pour cela ils proposent d'utiliser une approche itérative alternant des étapes de minimisation de $Q(\mathbf{a}, \hat{\mathbf{s}})$ par rapport aux paramètres sachant la dernière estimée des données manquantes, ensuite de $Q(\hat{\mathbf{a}}, \mathbf{s})$ par rapport aux données manquantes sachant les paramètres déjà estimés. Une initialisation des données manquantes est donc nécessaire. Ils proposent d'initialiser le vecteur \mathbf{s} à un vecteur nul. On note $\{z_k\}$ la reconstruction du signal $\{x_k\}$, il est donné pour tout instant n par :

$$z_n = \begin{cases} x_n & \text{si l'échantillon est disponible i.e. } c_n = 1 \\ \hat{x}_n & \text{sinon, i.e. } c_n = 0 \end{cases} \quad (1.34)$$

À chaque étape une fonction de coût quadratique est minimisée. Ainsi, pour l'estimation des paramètres, on exprime la fonction de coût sous la forme suivante dans laquelle les échantillons manquants sont remplacés par leurs estimées :

$$Q(\mathbf{a}, \hat{\mathbf{s}}) = \mathbf{a}^\top \hat{\mathbf{R}}_z \mathbf{a} + 2\mathbf{a}^\top \hat{\mathbf{r}}_z + \hat{\mathbf{r}}_z(0) \quad (1.35)$$

où $\hat{\mathbf{R}}_z$ est la matrice de Toeplitz constituée des estimées des L premières fonctions d'autocorrélation de $\{z_k\}$. L'estimation des fonctions d'autocorrélation se fait grâce à l'équation (1.10) dans

laquelle x_k est remplacée par z_k . Le vecteur $\hat{\mathbf{r}}_z$ est constitué des estimées des L premières fonctions d'autocorrélation de $\{z_k\}$. Ainsi, l'estimée du vecteur des paramètres qui minimise $Q(\mathbf{a}, \hat{\mathbf{s}})$ est la solution du système matriciel suivant :

$$\hat{R}_z \hat{\mathbf{a}} = -\hat{\mathbf{r}}_z \quad (1.36)$$

L'estimation des paramètres par cette méthode revient donc à résoudre le système d'équations (1.32) dans lequel les échantillons manquants sont remplacés par leurs prédictions.

D'autre part, pour estimer les échantillons manquants, la fonction de coût est exprimée sous la forme suivante dans laquelle les paramètres sont remplacés par leurs estimées :

$$Q(\hat{\mathbf{a}}, \mathbf{s}) = \mathbf{s}^\top B(\hat{\mathbf{a}}) \mathbf{s} + 2\mathbf{s}^\top \mathbf{p}(\hat{\mathbf{a}}) + \mathbf{q}(\hat{\mathbf{a}}) \quad (1.37)$$

où $B(\hat{\mathbf{a}}) = (b_{n_i} - b_{n_j})_{i,j=1,\dots,M}$ avec $b_l = \sum_{k=0}^L a_k a_{k+l}$ et $\mathbf{p}(\hat{\mathbf{a}})$ est le vecteur constitué des éléments $p_i(\mathbf{a}) = \sum_{k=-L}^L b_k x_{n_i-k}$ et $\mathbf{q}(\hat{\mathbf{a}})$ est une fonction des paramètres. L'estimation des échantillons manquants minimisant $Q(\hat{\mathbf{a}}, \mathbf{s})$ est la solution du système suivant :

$$B(\hat{\mathbf{a}}) \hat{\mathbf{s}} = -\mathbf{p}(\hat{\mathbf{a}}) \quad (1.38)$$

Janssen *et al.* proposent d'appliquer cet algorithme à la reconstruction de signaux audio qui seront découpés en tranches stationnaires. Chaque tranche est alors modélisée par un processus AR. Ils considèrent le cas où les échantillons manquants sont entourés par un grand nombre d'échantillons disponibles.

Dans le but d'identifier un modèle ARX à partir de données à échantillons manquants, Wallin *et al.* [69] proposent une méthode itérative similaire à celle proposée par Janssen *et al.* [36]. Ainsi, ils alternent jusqu'à la convergence des étapes d'estimation au sens des moindres carrés des données manquantes et des paramètres. Ils concluent que cette méthode conduit à une estimation biaisée des paramètres. Ils proposent donc d'ajouter une étape qui permet de calculer le biais et de l'annuler. Pour cela, ils proposent d'écrire le modèle mathématique de deux façons différentes. On s'intéressera ici au cas d'un modèle AR d'ordre L , les équations seront donc déduites de celles proposées dans [69] par simple annulation de la partie exogène.

Premièrement, les équations décrivant le modèle sont linéaires par rapport aux paramètres, elles s'écrivent sous la forme matricielle suivante :

$$\mathbf{x}^- = X\mathbf{a} + \boldsymbol{\varepsilon} \quad (1.39)$$

où $\mathbf{x}^- = [x_N, x_{N-1}, \dots, x_{L+1}]^\top$ est le vecteur constitué de tous les échantillons du signal à l'exception des L premiers échantillons. X est la matrice de régression de dimensions $(N-L) \times L$ dont les lignes sont constituées des vecteurs de régression suivants : $\mathbf{x}_{N-1}^\top, \dots, \mathbf{x}_L^\top$. $\boldsymbol{\varepsilon} = [\varepsilon_N, \dots, \varepsilon_L]^\top$ est le vecteur constitué des valeurs du bruit blanc à l'exception des L premières valeurs. Cette façon d'écrire les équations du modèle est utilisée lorsqu'il s'agit d'estimer les paramètres.

D'autre part, les équations sont aussi linéaires par rapport aux données, elles s'écrivent donc sous la forme suivante :

$$\Omega \mathbf{x} + \boldsymbol{\varepsilon} = \mathbf{0} \quad (1.40)$$

où $\mathbf{x} = [x_N, \dots, x_1]$ est un vecteur contenant tous les échantillons du signal. Ω est une matrice de dimensions $(N-L) \times N$. La première ligne de cette matrice est constituée du vecteur

$[-1 \ a_1 \ \dots \ a_L \ 0 \ \dots \ 0]$. Chacune des lignes de la matrice est le résultat d'un décalage circulaire de la ligne précédente d'une colonne à droite. Ainsi, la matrice Ω s'écrit sous la forme :

$$\Omega = \begin{bmatrix} -1 & a_1 & \dots & a_L & 0 & \dots & \dots & 0 \\ 0 & -1 & a_1 & \dots & a_L & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \dots & \ddots & \ddots & \dots & \vdots \\ 0 & \dots & 0 & -1 & a_1 & \dots & a_L & 0 \\ 0 & \dots & \dots & 0 & -1 & a_1 & \dots & a_L \end{bmatrix} \quad (1.41)$$

On note par Q_m et Q_d les matrices qui sélectionnent, respectivement, les échantillons manquants et les échantillons disponibles du vecteur \mathbf{x} . Ainsi, si M est le nombre d'échantillons manquants, la matrice Q_m est de dimensions $M \times N$ alors que la matrice Q_d est de dimensions $(N - M) \times N$. Chaque ligne i de la matrice Q_m (respectivement Q_d) contient un 1 à la position n_i correspondant à l'instant du i^{eme} échantillon manquant (respectivement disponible) et des zéros partout ailleurs. On a alors $Q_m Q_m^\top + Q_d Q_d^\top = I_d$, ceci permet d'écrire (1.40) sous la forme :

$$\Omega Q_m Q_m^\top \mathbf{x} + \Omega Q_d Q_d^\top \mathbf{x} + \boldsymbol{\varepsilon} = \mathbf{0} \quad (1.42)$$

On note par $\mathbf{s} = Q_m^\top \mathbf{x}$ le vecteur constitué des données manquantes et $\mathbf{x}^d = Q_d^\top \mathbf{x}$ celui des données disponibles. L'estimation au sens des moindres carrés des échantillons manquants sachant les paramètres est alors donnée par :

$$\hat{\mathbf{s}} = -(\Omega Q_m)^\dagger \Omega Q_d \mathbf{x}^d. \quad (1.43)$$

L'erreur de prédiction correspondante est donnée par :

$$\tilde{\mathbf{s}} = (\Omega Q_m)^\dagger \boldsymbol{\varepsilon}. \quad (1.44)$$

où \dagger désigne la pseudo-inverse d'une matrice, $(\Omega Q_m)^\dagger = ((\Omega Q_m)^\top (\Omega Q_m))^{-1} (\Omega Q_m)^\top$.

Soit $\mathbf{z} = [z_N, \dots, z_n, \dots, z_1]^\top$ le vecteur constitué des échantillons du signal reconstruit $\{z_n\}$ défini pour tout n par (1.34). Il s'exprime en fonction du vecteur \mathbf{x} contenant tous les échantillons du signal par :

$$\hat{\mathbf{x}} = \mathbf{z} = \left(I_d - Q_m (\Omega Q_m)^\dagger \Omega \right) Q_d Q_d^\top \mathbf{x}. \quad (1.45)$$

L'erreur de reconstruction d'une donnée disponible est nulle. En multipliant $\tilde{\mathbf{s}}$ par Q_m , on retrouve un vecteur de la taille du signal contenant les erreurs de prédiction des échantillons manquants chacune à sa propre position et des zéros ailleurs. Ce vecteur contient donc les erreurs de reconstruction du signal, il s'exprime par :

$$\tilde{\mathbf{z}} = Q_m \tilde{\mathbf{s}} = Q_m (\Omega Q_m)^\dagger \boldsymbol{\varepsilon} \quad (1.46)$$

En supposant que les données prédites sont correctes, les paramètres AR sont estimés en appliquant le principe des moindres carrés sur le vecteur des données $\hat{\mathbf{x}}^- = \mathbf{z}^-$. Ainsi, l'estimée du vecteur des paramètres est donnée par :

$$\hat{\mathbf{a}} = (\hat{X}^\top \hat{X})^{-1} \hat{X}^\top \hat{\mathbf{x}}^- \quad (1.47)$$

où \hat{X} est la matrice de régression dans laquelle les échantillons manquants sont remplacés par leurs prédictions.

Les paramètres ainsi estimés sont biaisés. Wallin *et al.* calculent une expression de ce biais et en tirent une condition pour une estimation non biaisée. Ainsi, en utilisant les égalités $\tilde{\mathbf{x}}^- = \hat{\mathbf{x}}^- - \mathbf{x}^-$ et $\tilde{X} = \hat{X} - X$, l'équation (1.39) s'écrit sous la forme suivante :

$$\hat{\mathbf{x}}^- = \mathbf{z}^- = \hat{X} \mathbf{a} + (\tilde{\mathbf{x}}^- - \tilde{X} \mathbf{a} + \boldsymbol{\varepsilon}) = \hat{X} \mathbf{a} + V \quad (1.48)$$

Après une estimation au sens des moindres carrés des paramètres supposant que les échantillons prédits sont corrects, on déduit qu'ils s'écrivent sous la forme suivante :

$$\begin{aligned} \mathbf{a} &= (\hat{X}^\top \hat{X})^{-1} \hat{X}^\top \hat{\mathbf{x}}^- - (\hat{X}^\top \hat{X})^{-1} \hat{X}^\top V \\ &= \hat{\mathbf{a}} - \tilde{\mathbf{a}} \end{aligned} \quad (1.49)$$

Ainsi une estimation au sens des moindres carrés des paramètres est non biaisée pour toute prédiction des échantillons manquants si et seulement si :

$$E \left\{ \hat{X}^\top V \right\} = E \left\{ \hat{X}^\top (\tilde{\mathbf{x}}^- - \tilde{X} \mathbf{a} + \boldsymbol{\varepsilon}) \right\} = 0 \quad (1.50)$$

Après développement de l'égalité précédente, Wallin *et al.* concluent que pour la prédiction des échantillons manquants utilisée (1.43), le biais d'identification est donné par :

$$\Delta = \left(\hat{X}^\top \hat{X} \right)^{-1} E \left\{ -X^\top \Omega \tilde{\mathbf{z}} \right\} = -\Omega Q_1 (\Omega Q_1)^\dagger E \left\{ X^\top \boldsymbol{\varepsilon} \right\} \quad (1.51)$$

Ceci nécessite donc une estimation des $N - L$ premières fonctions d'autocorrélation du signal et par la suite de la variance du bruit. Le biais d'identification ainsi calculé dépend fortement des paramètres. Ils proposent d'inclure dans l'algorithme itératif une étape de compensation du biais. Ainsi le biais calculé par (1.51) est retranché de l'estimation des paramètres donnée par (1.47).

1.3.2 Méthodes de traitement on-line

Nous résumons, dans cette partie, les principales méthodes de traitement paramétrique adaptatif de signaux à échantillons manquants. Les principales approches de prédiction de processus à échantillons manquants sont tout d'abord présentées, suivies d'un résumé des méthodes d'identification en ligne de signaux à échantillons manquants.

Prédiction des signaux à échantillons manquants

Soit $\{x_k\}$ une série chronologique stationnaire qui admet une représentation autorégressive donnée par l'équation (1.1). Lorsqu'un nombre fini d'échantillons est manquant, le prédicteur optimal donné par l'équation (1.6) ne peut pas être utilisé. Plusieurs méthodes ont déjà été proposées dans la littérature pour la prédiction d'une série chronologique à échantillons manquants. Nous présentons dans ce qui suit les principales méthodes.

1. Prédiction par régression linéaire modifiée.

Nous avons noté, au paragraphe 1.2, la mise en œuvre aisée du prédicteur linéaire d'un modèle AR donné par l'équation (1.6). Pour cela, celui-ci a été modifié pour l'adapter à la prédiction de processus à échantillons manquants. Il s'agit de remplacer dans le vecteur régresseur un échantillon manquant par sa prédiction pour la prédiction des échantillons

futurs. Nous parlerons alors de prédiction par régression linéaire modifiée. Ainsi, à un instant n la prédiction de l'échantillon x_n à passé incomplet s'exprime par :

$$\hat{x}_n = \sum_{i=1}^L a_i z_{n-i} \quad (1.52)$$

où $\{z_n\}$ est le signal reconstruit donné pour tout n par (1.34), il prend la valeur du signal lorsque l'échantillon est disponible et sa prédiction dans le cas contraire.

Ce prédicteur s'écrit aussi sous la forme suivante :

$$\hat{x}_n = \sum_{i=1}^L a_i [(1 - c_{n-i}) \hat{x}_{n-i} + c_{n-i} x_{n-i}]. \quad (1.53)$$

où c_n est la fonction binaire de modulation d'amplitude définie par l'équation (1.25).

Le prédicteur défini par l'équation (1.53) utilise les prédictions des échantillons manquants précédents qui dépendent des paramètres. Il n'est donc pas linéaire par rapport aux paramètres.

L'estimation des sorties non mesurées de systèmes modélisés par des processus linéaires stochastiques ARX en utilisant cette approche de prédiction est adressée par Albertos *et al.* [2]. Dans cet article, une analyse de la stabilité de ce prédicteur est menée dans le cas de systèmes à sous échantillonnage, lorsque la sortie est mesurée régulièrement à une période multiple de celle d'échantillonnage de l'entrée. La régression linéaire modifiée a été utilisée par Adams *et al.* [1] pour l'identification de processus ARX à échantillons manquants par le principe des moindres carrés. Cette approche de prédiction est également utilisée par Mirsaidi *et al.* [51, 53] dans un algorithme de gradient pour l'identification et la reconstruction adaptatives de processus AR à échantillons manquants.

2. Prédicteur à passé infini incomplet.

Soit à prédire x_n en fonction de son passé infini incomplet, en supposant que les données manquantes, $x_{n-n_1}, \dots, x_{n-n_M}$ pour $0 < n_1 < \dots < n_m < \dots < n_M$, sont d'un nombre fini M . P. Bondon [13] montre (théorème 3.1) que \hat{x}_n , la projection orthogonale de x_n sur l'espace engendré par les observations précédentes disponibles, vérifie l'équation suivante :

$$x_n - \hat{x}_n = - \sum_{s=0}^M \psi_s \sum_{j=0}^{n_s} a_{n_s-j} \varepsilon_{n-j}, \quad (1.54)$$

où les coefficients (ψ_s) vérifient l'équation matricielle suivante :

$$U [\psi_0, \psi_1, \dots, \psi_M]^\top = [1, 0, \dots, 0]^\top \quad (1.55)$$

U est la matrice non singulière de dimensions $(M+1) \times (M+1)$, dont les éléments sont donnés par :

$$u_{v,w} = \sum_{j=0}^{n_v \wedge n_w} a_{n_v-j} a_{n_w-j} \quad v, w = 0, \dots, M, \quad (1.56)$$

où $n_v \wedge n_w$ est la valeur minimale entre n_v et n_w .

\hat{x}_n ainsi défini est le prédicteur optimal au sens des moindres carrés de x_n sachant son passé incomplet. L'erreur quadratique moyenne de prédiction est donnée par :

$$E \{|x_n - \hat{x}_n|^2\} = \sigma_\epsilon^2 \psi_0. \quad (1.57)$$

P. Bondon [13] (théorème 3.2) démontre aussi que si le processus x_n admet une représentation autorégressive, \hat{x}_n admet aussi une représentation AR en fonction des échantillons passés disponibles. Celle-ci est unique, elle est donnée par :

$$\hat{x}_n = \sum_{i \in \mathbb{N} - \mathcal{M}} g_i x_{n-i}, \quad (1.58)$$

où $\mathcal{M} = \{n_1, \dots, n_M\}$ et

$$g_i = \delta_{i,0} - \sum_{s=0}^M \psi_s \sum_{j=0}^{n_s \wedge i} a_{n_s-j} a_{i-j}. \quad (1.59)$$

$\delta_{i,0}$ étant le symbole de Kronecker.

Ce théorème est une généralisation du corollaire 1 [11].

Dans le cas d'un processus AR d'ordre fini L , \hat{x}_n admet la représentation AR donnée par l'équation (1.58) avec $g_i = 0$ pour $i > n_M + L$ (Bondon [13], Exemple 4.2). Ainsi, dans ce cas, le prédicteur optimal à passé infini incomplet s'écrit sous la forme suivante :

$$\hat{x}_n = \sum_{i=0}^{n_M+L} c_{n-i} g_i x_{n-i} \quad (1.60)$$

où c_n est la fonction binaire qui vérifie l'équation (1.25), elle permet de n'utiliser dans la représentation autorégressive que les échantillons précédents disponibles.

Les équations décrites ici fournissent donc des expressions analytiques du prédicteur optimal au sens des moindres carrés d'une série chronologique à échantillons manquants. Ces expressions permettent une prédiction adaptative du signal, néanmoins le calcul n'est pas récursif. Ainsi la prédiction à l'instant $n + 1$ ne peut pas être déduite de la prédiction à l'instant n .

On s'intéressera à ce prédicteur dans le chapitre 2, particulièrement à sa représentation autorégressive pour la prédiction de processus AR d'ordre fini L .

Filtrage de Kalman

D'autres approches de prédiction se basent sur une représentation d'état du processus lorsque le modèle le permet. L'intérêt d'une telle représentation est qu'une estimation optimale en moyenne quadratique de l'état peut être obtenue grâce à un filtre de Kalman.

Dans cette optique, plusieurs filtres de Kalman ont été décrits dans la littérature [3, 27, 33, 34]. Dans le but d'implémenter un algorithme EM pour identifier des modèles ARX à échantillons manquants, Isaksson [33, 34] utilise, pour la prédiction des échantillons, une représentation d'état contenant un bruit d'observation qui est le même que celui d'état. D'autre part, pour estimer les sorties non mesurées de systèmes modélisés par des processus linéaires stochastiques ARX, Albertos *et al.*[3] proposent d'utiliser un filtre de Kalman comme alternative à l'approche de prédiction décrite par l'équation (1.53). Ils considèrent un bruit stochastique sur l'observation qui est de variance infinie lorsqu'un échantillon est manquant. Le gain de Kalman est alors calculé dans le cas d'un sous échantillonnage régulier pour un système linéaire stationnaire observable.

Dans [27], Fletcher *et al.* posent le problème de l'estimation d'un processus à partir de mesures obtenues grâce à un capteur à perte d'échantillons. Ils le considèrent comme un cas particulier du problème général de l'estimation de l'état dans les systèmes linéaires à sauts markoviens. Ainsi, le signal mesuré par le capteur est supposé être la sortie d'un système linéaire stochastique. Celui-ci est représenté par les équations d'état suivantes :

$$\begin{aligned}\mathbf{x}_{n+1} &= A\mathbf{x}_n + Bw_n^s \\ x_n^s &= C\mathbf{x}_n + Dw_n^s\end{aligned}\tag{1.61}$$

où \mathbf{x}_n est l'entrée du système linéaire décrit par les matrices (A, B, C, D) , et x_n^s la sortie du système linéaire, c'est le processus mesuré par le capteur.

Le capteur est modélisé par un gain linéaire avec un bruit additif. Il est donné par l'équation suivante :

$$\begin{aligned}y_n^c &= C_c x_n^s + D_c w_n^c \\ &= C_c C \mathbf{x}_n + [C_c D \quad D_c] [w_n^s \ w_n^c]^\top \\ &= C_2 \mathbf{x}_n + D_2 [w_n^s \ w_n^c]^\top\end{aligned}\tag{1.62}$$

où w_n^c est un bruit blanc de variance unité, C_c est le gain du capteur et D_c est un facteur d'échelle pour le bruit dépendant du capteur.

Finalement, la perte d'échantillons est modélisée par une mise à zero des échantillons manquants. Ainsi, la valeur observée y_n s'écrit :

$$y_n = c_n y_n^c = \begin{cases} y_n^c & \text{si l'échantillon est disponible,} \\ 0 & \text{sinon} \end{cases}\tag{1.63}$$

Ils proposent d'utiliser un filtre de Kalman pour une estimation optimale en moyenne quadratique de l'état. Celui-ci permet une estimation récursive du vecteur d'état \mathbf{x}_n et de la sortie du filtre linéaire x_n^s à partir de l'observation y_n . Les valeurs estimées minimisent l'erreur quadratique moyenne d'estimation. Le filtre de Kalman a été décrit dans de nombreux textes dont [14, 70]. Soit $\hat{x}_{n+1|n}$ et $\hat{x}_{n+1|n+1}$ les estimées *a priori* et filtrée de x_{n+1} , respectivement, $P_{n+1|n}$ et $P_{n+1|n+1}$, les matrices de covariance des erreurs d'estimation correspondantes. Pour le système décrit par les équations (1.61), (1.62) et (1.63), les équations du filtre de Kalman s'écrivent :

$$\begin{aligned}\hat{x}_{n+1|n} &= A\hat{x}_{n|n}, \\ P_{n+1|n} &= AP_{n|n}A^\top + BB^\top,\end{aligned}\tag{1.64}$$

Lorsqu'un échantillon est disponible, l'estimée *a priori* de l'état ainsi que de la matrice de covariance de l'erreur d'estimation sont filtrées selon les équations suivantes :

$$\begin{aligned}\hat{x}_{n+1|n+1} &= \hat{x}_{n+1|n} + K_{n+1}(y_{n+1} - C_2\hat{x}_{n+1|n}), \\ P_{n+1|n+1} &= (I_d - K_{n+1}C_2)P_{n+1|n},\end{aligned}\tag{1.65}$$

où K_{n+1} est le gain du filtre de Kalman donné, à l'instant $n + 1$, par l'équation suivante :

$$K_{n+1} = \left(AP_{n+1|n}C_2^\top + BC_cD \right) \left(C_2P_{n+1|n}C_2^\top + D_2D_2^\top \right)^{-1}\tag{1.66}$$

et

$$\hat{y}_{n+1|n} = c_{n+1}^\top \hat{x}_{n+1|n}.\tag{1.67}$$

Par contre, si l'échantillon est manquant, les valeurs *a priori* calculées par les équations (1.64) ne sont pas filtrées. On aura alors : $\hat{x}_{n+1|n+1} = \hat{x}_{n+1|n}$ et $P_{n+1|n+1} = P_{n+1|n}$.

Dans les deux cas, l'estimée de la sortie du filtre est donnée par $\hat{x}_{n+1|n}^s = C\hat{x}_{n+1|n}$.

Le filtre de Kalman ainsi décrit permet une prédiction récursive optimale au sens des moindres carrés de la sortie d'un système linéaire stationnaire ayant subi une perte d'échantillons connaissant les paramètres du modèle. Le problème de prédiction d'un processus AR d'ordre fini L à perte d'échantillons peut être considéré comme un cas particulier du problème décrit ici. On s'intéressera à cette particularisation au chapitre 3.

Dans ce qui suit, un état de l'art des méthodes adaptatives d'identification de signaux à échantillons manquants est présenté.

Identification adaptative par la méthode du maximum de vraisemblance

– L'algorithme EM récursif

Dans le but d'identifier un modèle ARX à partir de données manquantes, Isaksson [34] propose une version adaptative de l'algorithme EM off-line proposé par le même auteur [33]. Il propose donc une expression récursive sur le temps pour la mise à jour des paramètres estimés par l'algorithme EM. Pour cela, il procède de façon analogue à celle utilisée pour la dérivation de l'algorithme MCR. Ainsi, l'estimation des paramètres grâce à l'algorithme EM est donnée par l'équation (1.21). Les matrices $\sum_{k=1}^N E \{ \mathbf{x}_{k-1} \mathbf{x}_{k-1}^\top \}$ et $\sum_{k=1}^N E \{ \mathbf{x}_{k-1} \mathbf{x}_k \}$ peuvent être calculées à chaque instant de manière récursive sur le temps à partir de leurs valeurs à l'instant précédent. Par contre, la matrice $\sum_{k=1}^N E \{ \mathbf{x}_{k-1} \mathbf{x}_{k-1}^\top \}$ est calculée à partir de matrices qui peuvent être de rang plein. Pour cela le lemme d'inversion matricielle ne peut pas être appliqué dans ce cas pour un calcul récursif de $\left(\sum_{k=1}^N E \{ \mathbf{x}_{k-1} \mathbf{x}_{k-1}^\top \} \right)^{-1}$. La mise à jour des paramètres est alors rendue récursive sur le temps, par contre l'inversion d'une matrice est nécessaire à chaque instant. Ce qui augmente considérablement la complexité de calcul de cet algorithme en comparaison à l'algorithme MCR.

Identification adaptative par l'approche des moindres carrés

Les méthodes adaptative d'identification par l'approche des moindres carrés, en particulier les méthodes du gradient et l'algorithme MCR, étant simples et efficaces, des extensions de ces méthodes au cas de signaux à échantillons manquants ont été proposées. Il est à noter que la mise à jour des paramètres en utilisant ces méthodes est proportionnelle à l'erreur de prédiction qui ne peut pas être calculée lorsqu'un échantillon est manquant, les paramètres sont alors conservés à cet instant là. Un résumé de ces méthodes est présenté dans ce qui suit :

1. Algorithmes de type gradient

Les algorithmes de gradient étant simples, Misaidi *et al.* [51, 53] s'intéressent à leur extension pour l'identification de processus AR à échantillons manquants. Ils proposent alors un ensemble d'algorithmes de gradient qui diffèrent par l'approximation du critère à minimiser que ce soit l'erreur quadratique de prédiction instantanée [51], cumulée sur un horizon fini [51] (fenêtre glissante) ou cumulée sur tous les instants précédents (fenêtre croissante) [53]. Dans tous les cas, ils utilisent la prédiction par régression linéaire modifiée décrite ci-dessus au paragraphe 1.3.2. Ce prédicteur étant non linéaire par rapport aux paramètres, il s'en suit de même pour le critère quadratique à minimiser. Ainsi, dans le cas de signaux à échantillons manquants, malgré son apparence quadratique, le critère à minimiser est un polynôme en \mathbf{a} dont le degré dépend de l'ordre du modèle AR choisi et du schéma de perte. Or, la mise à jour des paramètres par la méthode de gradient nécessite le calcul du gradient

du critère qui n'est donc pas simple dans ce cas. Pour cela, Mirsaidi *et al.* proposent une méthode récursive selon laquelle le gradient du critère est calculé à un instant donné à partir des gradients aux L instants précédents.

Ces algorithmes sont simples à mettre en œuvre. Par contre, ils convergent typiquement vers une estimation biaisée des paramètres. Ce biais d'identification est prévu, il est dû à l'approche de prédiction utilisée qui n'est pas optimale au sens des moindres carrés. Ce résultat est déduit de l'expression du biais d'identification par le principe des moindres carrés calculée par Wallin *et al.* [68].

Les différents algorithmes proposés par Mirsaidi *et al.* seront décrits et analysés notamment en terme de biais dans le chapitre 2. Dans ce chapitre, des alternatives sont également proposées afin de résoudre le problème du biais d'identification. Les alternatives sont basées sur d'autres prédictions que celle par régression linéaire modifiée.

Cet algorithme a été adapté par Asswad *et al.* [6] à l'identification et à la reconstruction de signaux bidimensionnels AR(2D) à échantillons manquants. L'algorithme a été appliqué à la reconstruction d'images et a montré de bonnes performances.

2. Algorithmes de type MCR

L'algorithme MCR est simple et offre une convergence rapide ainsi qu'une adaptation rapide aux variations des paramètres. Il est donné à l'instant $n+1$ par les équations (1.17). Dans le cas de signaux à échantillons manquants, le vecteur régresseur (1.17a) ne peut pas être construit uniquement par des échantillons disponibles. Plusieurs approches ont été proposées pour son extension à l'identification de signaux à échantillons manquants. A l'instar des méthodes offline, l'approche la plus simple est la suivante : si à un instant $n+1$, l'échantillon x_{n+1} ainsi que tous les échantillons constituant le vecteur régresseur \mathbf{x}_n sont disponibles, les paramètres sont mis à jour grâce à l'équation (1.17d). Par contre, si l'un de ces échantillons est manquant, les paramètres sont conservés, $\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n$. Néanmoins, lorsque la probabilité de perte d'échantillons est élevée ou l'ordre du modèle à identifier est élevé, il devient rare d'avoir à la fois l'échantillon et le vecteur régresseur correspondant. Dans ce cas, la convergence de l'algorithme est très lente ainsi que le suivi de la variation des paramètres.

Adams *et al.* [1] proposent de remplacer dans le vecteur de régression les échantillons manquants par leurs prédictions. Celui-ci est alors noté à l'instant n par $\hat{\Psi}_n = \mathbf{z}_{n-1} = [z_{n-1}, \dots, z_{n-L}]^T$ où z_n prend la valeur de l'échantillon lorsque celui-ci est disponible et sa prédiction dans le cas contraire. La prédiction utilisée est celle par "régression linéaire modifiée". Le vecteur régresseur ainsi obtenu dépend des paramètres. L'algorithme MCR pseudo-linéaire est alors obtenu en négligeant la dépendance du vecteur régresseur des paramètres [1]. Le principe de l'algorithme se résume par : lorsqu'un échantillon est manquant, il est prédit par régression linéaire modifiée en utilisant l'estimée actuelle des paramètres, ces derniers restant inchangés (ne sont pas mis à jour). Par contre, lorsqu'un échantillon est disponible les équations de l'algorithme MCR sont exécutées après avoir remplacé dans le vecteur régresseur les échantillons manquants par leurs prédictions.

Cet algorithme est simple et offre une convergence rapide. Par contre, il converge typiquement vers une estimation biaisée des paramètres. Ce biais d'identification était prévisible pour les mêmes raisons que les algorithmes du gradients étendus au cas de signaux à échantillons manquants.

L'algorithme MCR pseudo-linéaire est généralisé par Sanchis *et al.* [62, 63]. Ils proposent de ne pas limiter la prédiction à celle par régression linéaire modifiée. Dans [63], une analyse de la convergence de cet algorithme est présentée. Ils s'intéressent au cas du sous

échantillonnage, et montrent que dans ce cas l'algorithme peut converger vers des minimums locaux non globaux. Cet algorithme sera présenté et analysé dans le chapitre 3. Dans ce chapitre, pour résoudre son problème de biais d'identification, nous utilisons, conjointement avec l'algorithme MCR pseudo-linéaire, un filtre de Kalman pour une prédiction optimale en moyenne quadratique de processus $AR(L)$ à échantillons manquants.

Chapitre 2

Algorithmes de gradient

2.1 Extension des algorithmes de gradient au cas d'échantillons manquants

Les algorithmes de gradient sont simples et permettent l'identification en temps réel d'un modèle paramétrique [44]. Ils ont été introduits de façon brève au cours du chapitre précédent. Pour plus de détails sur ces algorithmes se référer à [45]. Il s'agit d'estimer récursivement les paramètres optimaux d'un modèle au sens d'un critère d'erreur quadratique. Le principe est de mettre à jour, à chaque itération, les paramètres en se dirigeant dans la direction opposée au gradient du critère avec un facteur homothétique μ selon l'équation (1.12).

Dans le but d'identifier et de reconstruire en ligne un processus AR à échantillons manquants, Mirsaidi *et al.* [51, 52, 53] proposent des extensions de ces algorithmes à l'identification de processus AR à échantillons manquants. Il est évident, dans ce cas, que la mise à jour des paramètres n'a lieu que lorsqu'un échantillon est disponible. Ceci nécessite le calcul du gradient du critère à minimiser. Tout comme dans le cas sans échantillons manquants, le critère optimisé est l'erreur quadratique moyenne de prédiction. Par contre, dans ce cas, le prédicteur optimal donné par (1.6) ne peut pas être utilisé. Pour cela, Mirsaidi *et al.* proposent d'utiliser le prédicteur par régression linéaire modifiée explicité dans le paragraphe 1.3.2 du chapitre 1. Ainsi, ayant une estimée à l'instant n des paramètres, la prédiction de l'échantillon à l'instant $n + 1$ est donnée par :

$$\hat{x}_{n+1} = \sum_{i=1}^L \hat{a}_{i,n} z_{n-i+1} \quad (2.1)$$

où $\{z_n\}$ est le signal résultant de la reconstruction de $\{x_n\}$, il est donné pour tout n par (1.34). Il prend la valeur du signal lorsque l'échantillon est disponible et sa prédiction dans le cas contraire.

Dans ce qui suit, nous analysons tout d'abord le critère quadratique minimisé dans le cas de signaux à échantillons manquants. Par suite, nous présentons les différents algorithmes de gradient résultants des différentes approximations du critère.

2.1.1 Analyse du critère minimisé

Comme il a été évoqué dans le paragraphe 1.3.2 du chapitre 1, le prédicteur donné par l'équation (2.1) n'est pas linéaire par rapport aux paramètres. En effet, pour prédire un échantillon, les prédictions des échantillons manquants précédents, dépendants des paramètres, sont utilisées. On en déduit que l'erreur quadratique moyenne de prédiction, bien que d'apparence quadratique, est un polynôme en \mathbf{a} dont le degré dépend de l'ordre du modèle AR choisi ainsi que du schéma d'observation des échantillons. Ceci peut entraîner l'existence de plusieurs optimums globaux potentiels de la fonction de coût. Dans ce qui suit, nous utilisons un résultat obtenu par Wallin *et al.* pour déduire que le critère minimisé admet un minimum global unique dans le cas d'une perte d'échantillons aléatoire.

Le problème de minimums globaux multiples dans l'identification de modèles ARX à échantillons manquants est étudié dans [69]. Ils considèrent l'identification d'un système par le principe du maximum de vraisemblance. Ils utilisent un filtre de Kalman pour la prédiction des échantillons. Pour des processus gaussiens, les erreurs de prédictions obtenues en utilisant un filtre de Kalman sont indépendantes et gaussiennes. La fonction de vraisemblance est donc le produit de gaussiennes. Elle s'écrit pour un processus AR sous la forme suivante [69] :

$$V_{\mathbf{a}}(\mathbf{x}) = \prod_{i \in O} (2\pi s_i)^{(-1/2)} \exp\left(-\frac{(x_i - \hat{x}_i)^2}{2s_i}\right) \quad (2.2)$$

où O est l'ensemble des instants où les échantillons sont observés, et s_i est la covariance de l'erreur de prédiction.

Wallin *et al.* [69] écrivent la fonction de vraisemblance précédente sous la forme d'une distribution exponentielle. De plus, ils utilisent pour la prédiction d'un échantillon une représentation linéaire en fonction des échantillons précédents. Ils en déduisent qu'une statistique suffisante pour l'estimation des paramètres dans le cas d'une perte d'échantillons périodique, est l'estimation de toutes les fonctions d'autocorrélation du signal [69]. Or, pour la famille des distributions exponentielles, une statistique suffisante est à la fois minimale et complète [64]. Dans le cas où la perte des échantillons est aléatoire, asymptotiquement toutes les fonctions d'autocorrélations peuvent être estimés de manière consistante. Ils concluent donc qu'une perte d'échantillons aléatoire ne cause pas asymptotiquement un problème d'optimums globaux multiples.

Dans le cas de processus gaussiens, tel qu'un processus AR, la maximisation de la vraisemblance des observations est équivalente asymptotiquement à la minimisation de l'erreur quadratique moyenne de prédiction [14]. De plus, tout comme la prédiction utilisant le filtre de Kalman, la prédiction par régression linéaire modifiée admet une représentation linéaire en fonction des données précédentes. On en déduit que, lorsque la perte des échantillons est aléatoire, l'erreur quadratique moyenne de prédiction par régression linéaire modifiée admet un seul minimum global.

En pratique, comme dans le cas sans perte d'échantillons, plusieurs approximations de l'erreur quadratique moyenne de prédiction sont considérées donnant lieu ainsi à plusieurs algorithmes. Or, la mise à jour des paramètres par les algorithmes de gradient nécessite le calcul du gradient du critère minimisé, ceci n'est pas évident dans le cas de signaux à échantillons manquants. Ainsi, dans ce qui suit, pour chacun de ces algorithmes, l'approximation du critère, le calcul de son gradient ainsi que les caractéristiques de l'algorithme sont présentées. Asswad *et al.* [6, 7, 8], en adaptant ces algorithmes à l'identification de signaux 2D, proposent d'autres expressions récursives pour le calcul du critère et de son gradient que celles utilisées par Mirsaidi *et al.* [51, 52, 53]. On propose d'utiliser dans le cas 1D les expressions récursives utilisées par Asswad *et al.* pour leur simplicité.

2.1.2 Erreur quadratique instantanée

Dans ce cas, la fonction de coût minimisée est l'erreur quadratique instantanée, définie uniquement aux instants de disponibilité des échantillons. Lorsqu'un échantillon est disponible à l'instant $n + 1$, l'erreur quadratique est donnée par :

$$J_{n+1} = (x_{n+1} - \hat{x}_{n+1})^2. \quad (2.3)$$

Cet algorithme est donc une extension de l'algorithme du gradient stochastique au problème d'identification de signaux à échantillons manquants. Or la prédiction utilisée est non linéaire par rapport aux paramètres. Il s'en suit de même pour le critère (2.3) dont le gradient est donné par :

$$G_{n+1} = \frac{\partial J_{n+1}}{\partial \mathbf{a}} = -2(x_{n+1} - \hat{x}_{n+1}) \frac{\partial \hat{x}_{n+1}}{\partial \mathbf{a}}. \quad (2.4)$$

Le terme $\partial \hat{x}_{n+1} / \partial \mathbf{a}$ représente la dérivée partielle de la prédiction de l'échantillon x_{n+1} par rapport au vecteur des paramètres. C'est un vecteur de dimensions L dont l'élément d'ordre j est donné par :

$$\frac{\partial \hat{x}_{n+1}}{\partial a_j} = \frac{\partial \sum_{i=1}^L a_i z_{n-i+1}}{\partial a_j} = z_{n-j+1} + \sum_{i=1}^L a_i \frac{\partial z_{n-i+1}}{\partial a_j} \quad j = 1, \dots, L \quad (2.5)$$

où,

$$\frac{\partial z_{n-i}}{\partial a_j} = \begin{cases} 0 & \text{si } x_{n-i} \text{ est disponible} \\ \frac{\partial \hat{x}_{n-i}}{\partial a_j} & \text{sinon.} \end{cases} \quad (2.6)$$

Ainsi, en écrivant les L relations sous une forme vectorielle, le terme $\partial \hat{x}_n / \partial \mathbf{a}$ se calcule récursivement grâce à l'équation suivante [51] :

$$\frac{\partial \hat{x}_{n+1}}{\partial \mathbf{a}} = \mathbf{z}_n + \left[\frac{\partial z_n}{\partial \mathbf{a}} \mid \dots \mid \frac{\partial z_{n-L+1}}{\partial \mathbf{a}} \right] \mathbf{a} = \mathbf{z}_n + \frac{\partial \mathbf{z}_n}{\partial \mathbf{a}} \mathbf{a} \quad (2.7)$$

où $\mathbf{z}_n = [z_n \dots z_{n-L+1}]^\top$ est le vecteur de régression du signal reconstruit à l'instant $n + 1$.

Cet algorithme est simple à mettre en œuvre en pratique, sa complexité est de l'ordre de $O(L^2)$ à chaque itération. Il permet de plus le suivi en temps réel des paramètres d'un signal non stationnaire ayant subi une perte d'échantillons. Cet algorithme a évidemment les inconvénients de l'algorithme du gradient stochastique son homologue dans le cas sans perte d'échantillons, à savoir la faible vitesse de convergence ainsi que la variance élevée des paramètres estimés. Le choix de la valeur du pas d'adaptation μ est en effet un bon compromis entre la vitesse de convergence et la variance des paramètres [44, 52].

2.1.3 Erreur quadratique moyenne (fenêtre croissante)

Dans ce cas, à chaque instant, la moyenne statistique de toutes les erreurs quadratiques de prédiction précédentes est minimisée. C'est donc une extension de l'algorithme SGA (*'Statistical Gradient Averaging'*) au problème d'identification de signaux à échantillons manquants. Il est clair que seules les erreurs de prédiction correspondants à des échantillons disponibles doivent

être considérées dans le calcul de la fonction de coût. Cette dernière n'est pas définie aux instants d'échantillons manquants. Elle est donc donnée à un instant $n + 1$ par :

$$\begin{aligned} J_{n+1} &= \frac{1}{N_{n+1}^d} \sum_{h=L+1}^{n+1} c_h (x_h - \hat{x}_h)^2 \\ &= \frac{1}{N_{n+1}^d} \sum_{h=L+1}^{n+1} c_h \left(x_h - \sum_{i=1}^L a_i z_{h-i} \right)^2. \end{aligned} \quad (2.8)$$

où $\{c_h\}$ est la variable binaire définie par l'équation (1.25). Elle est utilisée pour ne tenir compte en calculant la fonction de coût que des erreurs de prédiction correspondant à des échantillons disponibles. $N_n^d = \sum_{h=L+1}^n c_h$ est le nombre d'échantillons disponibles jusqu'à l'instant n .

Le gradient de la fonction de coût précédente est donné par :

$$\frac{\partial J_{n+1}}{\partial \mathbf{a}} = \frac{-2}{N_{n+1}^d} \sum_{h=L+1}^{n+1} c_h (x_h - \hat{x}_h) \frac{\partial \hat{x}_h}{\partial \mathbf{a}} \quad (2.9)$$

où le terme $\partial \hat{x}_h / \partial \mathbf{a}$ est calculé grâce à l'équation (2.7).

Le gradient donné par (2.9) peut se calculer récursivement. Ainsi si un échantillon est disponible à l'instant $n + 1$, le gradient est mis à jour grâce à l'équation récursive suivante :

$$\frac{\partial J_{n+1}}{\partial \mathbf{a}} = \frac{N_n^d}{N_n^d + 1} \frac{\partial J_n}{\partial \mathbf{a}} + \frac{1}{N_n^d + 1} G_{n+1} \quad (2.10)$$

avec $G_{n+1} = -2(x_{n+1} - \hat{x}_{n+1}) \frac{\partial \hat{x}_{n+1}}{\partial \mathbf{a}}$. Par contre, si à l'instant $n + 1$, l'échantillon est manquant, le gradient de la fonction de coût est conservé $\frac{\partial J_{n+1}}{\partial \mathbf{a}} = \frac{\partial J_n}{\partial \mathbf{a}}$.

Ici, contrairement au gradient stochastique, toutes les erreurs quadratiques de prédictions précédentes sont prises en compte également pour le calcul du gradient. Pour cela, la variance sur l'estimation des paramètres est plus faible que celle obtenue grâce à l'algorithme précédent. Par contre, ceci implique une faible capacité de poursuite des variations des paramètres dans un environnement non stationnaire. Pour avoir un compromis entre une faible variance sur l'estimation des paramètres et une bonne capacité de poursuite des paramètres dans un contexte non stationnaire, la fonction de coût est définie sur un horizon limité des erreurs de prédictions [44, 52]. Cet algorithme est présenté dans ce qui suit.

2.1.4 Erreur quadratique sur une fenêtre glissante

Ici, le critère à minimiser est la moyenne des erreurs quadratiques de prédiction sur un horizon fini de durée H . Le critère est donc calculé sur une fenêtre glissante dont la durée dépend évidemment du degré de non stationnarité du signal. Les paramètres d'un processus non stationnaire varient en fonction du temps et non en fonction du nombre d'échantillons disponibles. Pour cela, la taille de la fenêtre n'est pas un nombre précis d'échantillons disponibles mais par contre une durée temporelle. Le critère à minimiser est alors donné, à un instant $n + 1$, par :

$$J_{n+1} = \frac{1}{N_{H,n+1}^d} \sum_{h=n-H+1}^{n+1} c_h (x_h - \hat{x}_h)^2. \quad (2.11)$$

où $N_{H,n}^d = \sum_{h=n-H+1}^{n+1} c_h$ est le nombre d'échantillons disponibles existants dans la fenêtre de largeur H . La modulation d'amplitude binaire c_h est utilisée pour les mêmes raisons que précédemment.

Le gradient de (2.11) par rapport aux paramètres est donné par :

$$\frac{\partial J_{n+1}}{\partial \mathbf{a}} = \frac{-2}{N_{H,n+1}^d} \sum_{h=n-H+1}^{n+1} c_h G_{n+1} \quad (2.12)$$

où G_{n+1} est définie par (2.4). Ceci nécessite donc à chaque instant la connaissance des H derniers gradients calculés. Cet algorithme permet d'obtenir un compromis entre les deux algorithmes précédents en termes de variance sur l'estimation des paramètres et de la capacité de poursuite de la variation des paramètres. Par contre il est de complexité calculatoire légèrement plus grande.

2.1.5 Résumé global des algorithmes

Les algorithmes présentés dans la section précédente ne se distinguent que par le critère optimisé et par conséquent son gradient. Ils se résument donc à l'instant $n + 1$ aux étapes suivantes :

Si l'échantillon est manquant

- prédire l'échantillon \hat{x}_{n+1} grâce à l'équation (2.1)
- les paramètres ne sont pas mis à jour : $\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n$
- calculer $\partial \hat{x}_{n+1} / \partial \mathbf{a}$ grâce à (2.7) pour $\mathbf{a} = \hat{\mathbf{a}}_n$
- actualiser le vecteur $\partial \mathbf{z}_{n+1} / \partial \mathbf{a}$ via :

$$\frac{\partial \mathbf{z}_{n+1}}{\partial \mathbf{a}} = \left[\frac{\partial \hat{x}_{n+1}}{\partial \mathbf{a}} \left| \frac{\partial z_n}{\partial \mathbf{a}} \right| \cdots \left| \frac{\partial z_{n-L+2}}{\partial \mathbf{a}} \right| \right]^\top. \quad (2.13)$$

Si l'échantillon est disponible

- prédire l'échantillon \hat{x}_{n+1} grâce à l'équation (2.1)
- calculer $\partial \hat{x}_{n+1} / \partial \mathbf{a}$ grâce à (2.7) pour $\mathbf{a} = \hat{\mathbf{a}}_n$
- calculer $G_{n+1} = -2(x_{n+1} - \hat{x}_{n+1}) \frac{\partial \hat{x}_{n+1}}{\partial \mathbf{a}}$ et en déduire le gradient du critère en fonction de l'algorithme choisi :

1. Erreur quadratique instantanée :

$$\frac{\partial J_{n+1}}{\partial \mathbf{a}} = \hat{G}_{n+1}. \quad (2.14)$$

2. Erreur quadratique moyenne (fenêtre croissante) :

$$\frac{\partial J_{n+1}}{\partial \mathbf{a}} = \frac{N_n^d}{N_n^d + 1} \frac{\partial J_n}{\partial \mathbf{a}} + \frac{1}{N_n^d + 1} G_{n+1}. \quad (2.15)$$

3. Erreur quadratique moyenne (fenêtre glissante) :

$$\frac{\partial J_{n+1}}{\partial \mathbf{a}} = \frac{-2}{N_{H,n+1}^d} \sum_{k=n-H+1}^{n+1} c_k G_{n+1} \quad (2.16)$$

- mettre à jour les paramètres grâce à l'équation (1.12).

– actualiser le vecteur $\partial z_{n+1}/\partial \mathbf{a}$ via :

$$\frac{\partial z_{n+1}}{\partial \mathbf{a}} = \left[0 \left| \frac{\partial z_n}{\partial \mathbf{a}} \right| \dots \left| \frac{\partial z_{n-L+2}}{\partial \mathbf{a}} \right| \right]^\top. \quad (2.17)$$

En se référant aux équations (2.10) et (2.12) on constate que le gradient dans le cas de fenêtres croissantes ou fenêtres glissantes est calculé récursivement en utilisant à chaque instant le gradient stochastique.

2.1.6 Observations et analyse des performances des algorithmes

L'intérêt principal des algorithmes présentés ci-dessus est la simplicité de l'implémentation ainsi que la faible complexité de calcul permettant ainsi d'envisager des applications en temps réel. Par contre, les simulations montrent que ces algorithmes convergent vers une estimation biaisée des paramètres. Le biais semble être indépendant de l'initialisation des paramètres pour une perte aléatoire des échantillons. Il n'est donc pas dû à une convergence vers un minimum local de la fonction de coût. Ceci est confirmé par l'analyse faite au paragraphe (2.1.1) où l'on conclut que lorsque la perte des échantillons est aléatoire, l'erreur quadratique moyenne de prédiction par régression linéaire modifiée admet un seul minimum global.

De plus, ce biais croît avec la probabilité q de perte des échantillons et dépend des paramètres à identifier. Cette observation est confirmée par Wallin *et al.* [68]. Ainsi, ils montrent que l'application du principe des moindres carrés sur le signal reconstruit (où les échantillons manquants sont remplacés par leur prédiction supposée correcte) conduit à une estimation biaisée des paramètres. Ils calculent l'expression de ce biais et en déduisent une condition pour une estimation non biaisée. Ceci est fait l'objet du paragraphe 1.3.1 du chapitre 1. Or, les paramètres estimés par les algorithmes présentés ci-dessus sont égaux asymptotiquement et en moyenne à la solution au sens des moindres carrés des paramètres qui caractérisent le signal reconstruit (où les échantillons manquants sont remplacés par leur prédiction par régression linéaire modifiée). Ils convergent donc vers une estimation biaisée des paramètres [68]. D'autre part, on déduit de l'équation (1.50) que lorsque l'approche de prédiction utilisée minimise l'erreur quadratique moyenne de prédiction, l'estimation des paramètres par le principe des moindres carrés est non biaisée. Pour cela, nous proposons dans ce qui suit des alternatives aux algorithmes présentés ci-dessus utilisant d'autres approches de prédiction afin de résoudre leur problème de biais d'identification.

2.2 Alternatives proposées

Afin de résoudre le problème de biais des algorithmes précédents, on propose ici de nouvelles alternatives basées sur d'autres approches de prédiction. Les approches de prédiction sont tout d'abord présentées et illustrées à l'aide d'un exemple. Ensuite, les algorithmes de type gradient basés sur ces approches de prédictions sont décrits.

2.2.1 Approches de prédiction utilisées

Les différentes approches de prédiction qu'on propose d'utiliser dans des algorithmes de gradient sont le prédicteur à k -pas et le prédicteur à passé infini incomplet proposé par Bondon [13].

Prédicteur à k -pas

Soit P_n la projection orthogonale d'un échantillon sur le sous espace vectoriel \mathcal{S} engendré par $1, x_1, \dots, x_n$. Le prédicteur à k -pas consiste à prédire un échantillon à l'instant $n+k$ en fonction de son passé jusqu'à l'instant n . $P_n x_{n+k}$ est le prédicteur à k -pas optimal de x_{n+k} [14]. C'est la combinaison linéaire optimale de $1, x_1, \dots, x_n$ permettant la prédiction de x_{n+k} . Selon le théorème de la projection ([14] Théorème 2.3.1), le prédicteur à k -pas ainsi défini est l'élément unique de \mathcal{S} qui vérifie :

$$\|x_{n+k} - \hat{x}_{n+k}\| = \inf_{y \in \mathcal{S}} \|x_{n+k} - y\|. \quad (2.18)$$

C'est le meilleur prédicteur au sens des moindres carrés de x_{n+k} appartenant au sous espace vectoriel \mathcal{S} .

Les équations récursives du prédicteur à k -pas optimal d'un modèle ARMA sont données dans [14]. Par analogie, en appliquant l'opérateur P_n aux deux membres de l'équation (1.1) écrite à l'instant $n+k$, on déduit que dans le cas d'un modèle AR(L), l'équation du prédicteur à k -pas optimal s'écrit à un instant $n > L$ par :

$$\hat{x}_{n+k} = P_n x_{n+k} = \sum_{i=1}^L a_i P_n x_{n+k-i}, \quad (2.19)$$

Il est à noter que dans le cas où $k < L$, lorsque $n+k-i \leq n$, $P_n x_{n+k-i} = x_{n+k-i}$.

De plus, on sait que dans le cas d'un processus AR d'ordre L , $P_n x_{n+1} = \sum_{i=1}^L a_i x_{n+1-i}$ [14]. On déduit donc de l'équation récursive (2.19) que le prédicteur à k -pas optimal est une combinaison linéaire de x_{n-L+1}, \dots, x_n uniquement. Il admet donc la représentation autorégressive suivante [12] :

$$\hat{x}_{n+k} = \sum_{i=1}^L g_i^k x_{n-i+1}. \quad (2.20)$$

Dans [12], Bondon établit des relations récursives permettant de calculer, à partir des coefficients autorégressifs (a_i), les coefficients de prédiction à k -pas, (g_i^k), ainsi que la variance de l'erreur de prédiction.

On montre aussi que dans le cas d'un processus AR(2), g_1^k, g_2^k et la variance, Var , de l'erreur de prédiction sont donnés par (voir Annexe A) :

$$\begin{aligned} g_1^k &= \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} C_{k-i}^i a_1^{h-2i} a_2^i \\ g_2^k &= \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} C_{k-i-1}^i a_1^{h-2i-1} a_2^{i+1} \end{aligned} \quad (2.21)$$

$$Var = \mathbf{E} \{ (x_{n+k} - \hat{x}_{n+k})^2 \} = \left(\sum_{i=0}^{\lfloor \frac{k-2}{2} \rfloor} C_{k-i}^i a_1^{h-2i} a_2^i \right)^2 \sigma_\epsilon^2$$

On propose d'adapter la prédiction à k -pas au problème de prédiction d'un processus AR d'ordre L à échantillons manquants. Dans le cas général où la perte d'échantillons est aléatoire, on propose de prédire un échantillon en fonction des L échantillons consécutifs disponibles précédents les plus proches. Ainsi, on ne tient pas compte des échantillons disponibles entre l'échantillon à prédire et la suite des L échantillons consécutifs disponibles. Le principe est représenté à la figure

2.1. Le pas k est donc le nombre d'échantillons séparant l'échantillon à prédire (à l'instant $n+k$) et la séquence de L échantillons consécutifs disponibles la plus proche (aux instants $n-L+1, \dots, n$). Celui-ci dépend donc du schéma de perte et peut devenir arbitrairement grand ce qui mène à une large variance de l'erreur de prédiction. Ayant défini le pas k et l'instant n , on utilise donc l'équation récurrente (2.19) pour la prédiction. Il est à noter que dans ce cas, tout comme dans le cas de l'identification sans échantillons manquants, la minimisation par rapport aux paramètres de l'erreur quadratique de prédiction revient à résoudre un système de L équations à L inconnues (qui sont les paramètres du modèle AR à identifier) et ceci quelque soit le schéma de perte.

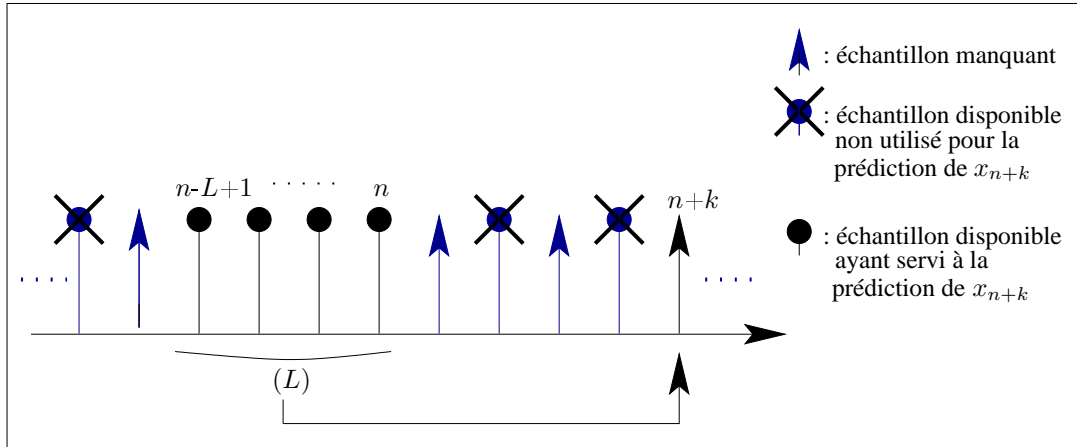


FIG. 2.1 – Adaptation du prédicteur à k -pas à la prédiction de signaux à échantillons manquants.

Prédicteur à passé infini incomplet

Le prédicteur à passé infini incomplet est proposé par Bondon [11, 13]. Cette approche de prédiction a été décrite dans la section 2 du premier chapitre. C'est la projection orthogonale d'un échantillon sur son passé infini incomplet pour un nombre fini d'échantillons manquants. Il est optimal au sens des moindres carrés et admet une représentation autorégressive unique en fonction des données précédentes disponibles. Celle-ci est donnée par (1.58). On s'intéresse ici particulièrement à la prédiction d'un processus AR d'ordre fini L à échantillons manquants. Dans ce cas, on montre que la prédiction optimale d'un échantillon est une combinaison linéaire de toutes les observations précédentes jusqu'à ce que L échantillons consécutifs disponibles soient rencontrés. Ainsi, soit x_n l'échantillon à prédire en fonction des échantillons précédents disponibles et soit $x_{n-k-L+1}, \dots, x_{n-k}$ la séquence de L échantillons consécutifs disponibles la plus proche de x_n . Il s'agit donc de montrer que les coefficients g_i de la représentation autorégressive de \hat{x}_n donnés par (1.59) s'annulent pour $i > n - k - L + 1$. Cette démonstration est présentée dans l'Annexe B.

Ainsi, le prédicteur d'un processus AR(L) à passé infini incomplet admet la représentation autorégressive donnée par l'équation (1.60) dans laquelle M désigne le nombre d'échantillons manquants entre l'échantillon à prédire et la séquence de L échantillons consécutifs disponibles la plus proche. Les échantillons manquants sont donc aux instants $n - n_M, \dots, n - n_1$. Les L échantillons consécutifs disponibles les plus proches de l'échantillon à prédire sont aux instants $n - n_M - L, \dots, n - n_M - 1$. Ainsi les échantillons ayant servi à la prédiction de x_n sont tous les échantillons disponibles entre les instants $n - n_M - L$ et n . Ceci justifie la multiplication dans

(1.60) par la modulation d'amplitude c_i qui permet de sélectionner uniquement les échantillons disponibles. Ceci est représenté à la figure 2.2.

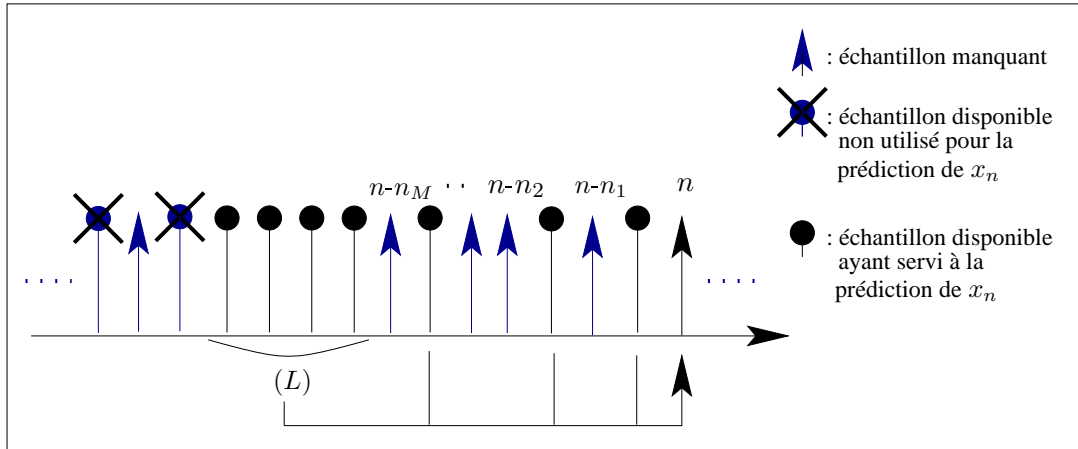


FIG. 2.2 – Prédiction d'un processus AR(L) à échantillons manquants utilisant le prédicteur à passé fini incomplet.

Ainsi, la prédiction à k -pas est alors une projection orthogonale sur un sous espace de celui sur lequel on projette grâce à ce prédicteur. L'erreur quadratique moyenne de prédiction obtenue pour ce prédicteur est alors inférieure à celle obtenue pour le prédicteur à k -pas.

La prédiction d'un processus AR(L) utilisant cette approche de prédiction nécessite à chaque instant l'inversion de la matrice U . On rappelle que U est la matrice non singulière de dimensions $(M + 1) \times (M + 1)$, dont les éléments sont donnés par :

$$u_{v,w} = \sum_{j=0}^{n_v \wedge n_w} a_{n_v-j} a_{n_w-j} \quad v, w = 0, \dots, M, \quad (2.22)$$

où $n_v \wedge n_w$ est la valeur minimale entre n_v et n_w .

Les dimensions de cette matrice et la valeur de ses éléments dépendent à chaque instant du schéma de perte. Son calcul à un instant donné ne peut donc pas être déduit récursivement de son expression à l'instant précédent. Pour des probabilités de pertes élevées ou pour des processus AR d'ordre L élevés, il devient rare d'avoir L échantillons consécutifs disponibles. Par conséquent, les dimensions $(M + 1) \times (M + 1)$ de la matrice U deviennent souvent élevées ce qui rend le calcul de la prédiction complexe.

Exemple de comparaison et conclusions

Les deux prédicteurs présentés ci-dessus ainsi que le prédicteur par régression linéaire modifiée sont illustrés et comparés ici à travers un exemple particulier. Il s'agit de prédire l'échantillon x_7 d'un processus AR(2) ayant subi une perte d'échantillons suivant le schémas de perte particulier donné par la figure 2.3.

Nous utilisons pour les trois prédictions une représentation autorégressive en fonction des échantillons précédents disponibles afin de pouvoir les comparer. Les représentations autorégressives des prédictions à k -pas et à passé incomplet ont été discutées ci-dessus 2.2.1. La prédiction par régression linéaire modifiée admet elle aussi une représentation autorégressive. Ainsi, considérons son expression donnée par l'équation (2.1) dans laquelle nous remplaçons à chaque fois et

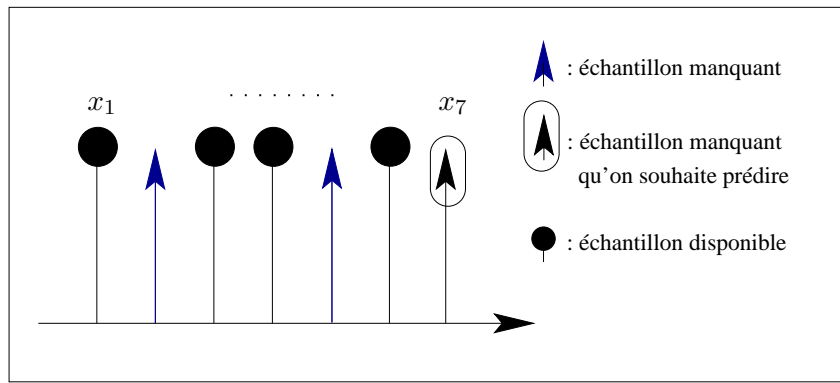


FIG. 2.3 – Schémas de perte particulier permettant la comparaison des trois prédicteurs utilisés dans ce chapitre.

de façon récursive la prédiction d'un échantillon par son expression donnée par la même équation. On obtient alors que la prédiction par régression linéaire modifiée admet, comme le prédicteur à passé incomplet, une représentation autorégressive en fonction des données disponibles précédentes jusqu'à ce que L échantillons consécutifs disponibles soient rencontrés. Nous présentons dans ce qui suit la prédiction de x_7 utilisant les différentes approches de prédiction.

1. La prédiction par régression linéaire modifiée :

$$\hat{x}_7 = a_1 x_6 + a_1 a_2 x_4 + a_2^2 x_3. \quad (2.23)$$

2. Le prédicteur à k -pas : ici, x_7 est prédit en fonction de x_4 et de x_3 (L échantillons consécutifs disponibles). Les équations (2.21) pour $k = 3$ donnent :

$$\hat{x}_7 = (a_1^3 + 2a_1 a_2) x_4 + (a_1^2 a_2 + a_2^2) x_3. \quad (2.24)$$

3. Le prédicteur à passé incomplet : ici $M = 1, n_M = 2$

$$\hat{x}_7 = \left(a_1 + \frac{a_1 a_2}{1 + a_1^2} \right) x_6 + \left(a_1 a_2 - \frac{a_1^3 a_2 + a_1 a_2^2}{1 + a_1^2} \right) x_4 + \left(a_2^2 - \frac{a_1^2 a_2^2}{1 + a_1^2} \right) x_3. \quad (2.25)$$

Comme prévu, la prédiction par régression linéaire modifiée et celle à passé incomplet utilisent plus d'informations pour la prédiction de x_7 que celle à k -pas. De plus, d'après l'exemple, le prédicteur par régression linéaire modifiée a une représentation autorégressive différente de celui à passé incomplet. En effet, les coefficients autorégressifs g_i de ce dernier contiennent des termes additifs correctifs en comparaison aux coefficients autoregressifs obtenus pour la prédiction par régression linéaire modifiée. Or, on sait que la projection orthogonale d'un échantillon sur son passé incomplet admet une représentation autorégressive unique dont les coefficients sont donnés par l'équation (1.58) [11, 13]. On en déduit que la prédiction d'un échantillon par régression linéaire modifiée n'est pas sa projection orthogonale sur son passé à échantillons manquants. Cette approche de prédiction n'est donc pas optimale au sens des moindres carrés. Cette observation est confirmée dans [34]. En effet, il est noté dans [34] que l'espérance d'un état basée sur les données observées ne revient pas à remplacer simplement un état inconnu par son estimée mais des termes correctifs basés sur la matrice de covariance de l'erreur de prédiction doivent être pris en compte. Or, les paramètres estimés utilisant l'un des algorithmes proposés par [53] sont égaux asymptotiquement à la solution au sens des moindres carrés des paramètres qui caractérisent le signal

reconstruit (où les échantillons manquants sont remplacés par leur prédiction). Ils convergent donc typiquement vers une estimation biaisée des paramètres si l'approche de prédiction utilisée n'est pas optimale au sens des moindres carrés [68]. La prédiction par régression linéaire modifiée n'étant pas optimale au sens des moindres carrés, le biais existant sur l'estimation des paramètres utilisant les algorithmes proposés dans [51, 52, 53] est justifié.

Il est à noter ici que la prédiction à k -pas et celle à passé infini incomplet sont toutes les deux optimales au sens des moindres carrés. Par contre, elles effectuent des projections orthogonales sur des espaces différents. Ceci nous mène à les utiliser dans ce qui suit dans des algorithmes de gradient afin de résoudre le problème de biais d'identification.

2.2.2 Algorithmes de gradient utilisant le prédicteur à k -pas

La prédiction à k -pas est équivalente à la prédiction par régression linéaire modifiée lorsque tous les échantillons entre le dernier bloc de L échantillons consécutifs disponibles, x_{n-L+1}, \dots, x_n , et l'échantillon à l'instant courant x_{n+k} sont manquants. Utilisant cette propriété du prédicteur à k -pas, les algorithmes de gradient utilisant le prédicteur à k -pas peut donc être déduit simplement, et de façon récursive, à partir de ceux proposés par Mirsaidi *et al.* [51]. Ce dernier est modifié en considérant un échantillon disponible comme manquant au cours des instants suivants jusqu'à ce qu'un bloc de L échantillons consécutifs disponibles soit formé. Les prédictions futures se feront donc à partir de ce dernier bloc. Ce principe est représenté par la figure 2.4.

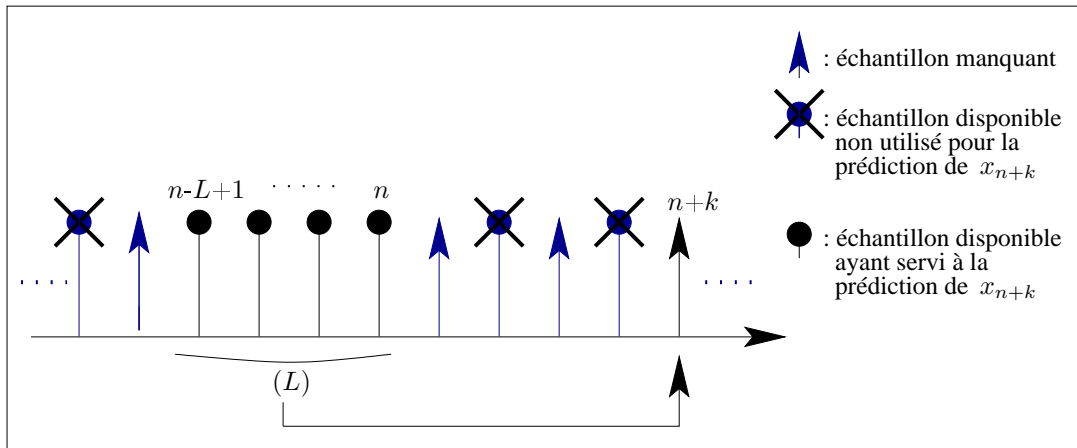


FIG. 2.4 – Principe de l'algorithme LMS-like basé sur la prédiction à k -pas.

Ainsi, si un échantillon disponible à l'instant $n+k$ ne forme pas avec les $L-1$ échantillons précédents un bloc de L échantillons consécutifs disponibles, il est considéré comme manquant au cours des instants suivants. Ceci intervient dans la prédiction des échantillons futurs et dans le calcul du gradient à ces instants là. Ainsi, la prédiction de cet échantillon est utilisée pour les prédictions futures, i.e. $z_{n+k} = \hat{x}_{n+k}$, de plus, $\partial \hat{x}_{n+k} / \partial \mathbf{a}$ sera utilisée dans l'équation (2.7) au lieu de 0 pour la mise à jour du gradient.

Par contre, si $x_{n+k}, \dots, x_{n+k-L+1}$ sont disponibles, les échantillons suivants seront prédits en fonction de ce bloc, le vecteur de régresseurs est alors $\mathbf{z}_{n+k} = \mathbf{x}_{n+k}$. Puisque ces échantillons ne sont pas considérés comme manquants dans la suite, leurs gradients par rapport aux paramètres, $\partial z_{n+h} / \partial \mathbf{a}, \dots, \partial z_{n+h-p+1} / \partial \mathbf{a}$ sont donc égaux à zéro. Il est à noter que dans ce cas, se référant à l'équation (2.1), la prédiction à l'instant $n+k+1$ est linéaire par rapport aux paramètres. Ainsi le gradient $\partial \hat{x}_{n+k+1} / \partial \mathbf{a}$ calculé par l'équation (2.7) est bien égal à \mathbf{x}_{n+k} .

L'algorithme se résume donc à l'instant $n + k$ aux étapes suivantes :

Si l'échantillon est disponible

- prédire l'échantillon \hat{x}_{n+k} grâce à l'équation (2.1)
- calculer $\partial\hat{x}_{n+k}/\partial\mathbf{a}$ grâce à (2.7) pour $\mathbf{a} = \hat{\mathbf{a}}_{n+k-1}$
- calculer $G_{n+k} = -2(x_{n+k} - \hat{x}_{n+k})\frac{\partial\hat{x}_{n+k}}{\partial\mathbf{a}}$ et en déduire le gradient du critère en fonction de l'approximation de la fonction de coût choisie.
- mettre à jour les paramètres grâce à l'équation (1.12).
- vérifier si un nouveau bloc de L échantillons consécutifs disponibles est formé :
 1. si $x_{n+k}, \dots, x_{n+k-L+1}$ sont disponibles, ils serviront à la prédiction des échantillons futurs :
 - tous les éléments du vecteur $\partial\mathbf{z}_{n+k}/\partial\mathbf{a}$ sont mis à zéros.
 - $\mathbf{z}_{n+k} = \mathbf{x}_{n+k}$.
 2. dans le cas contraire, l'échantillon à l'instant courant est considéré manquant pour les étapes suivantes :
 - actualiser le vecteur $\partial\mathbf{z}_{n+1}/\partial\mathbf{a}$ par l'équation :

$$\frac{\partial\mathbf{z}_{n+k}}{\partial\mathbf{a}} = \left[\frac{\partial\hat{x}_{n+k}}{\partial\mathbf{a}} \mid \frac{\partial z_{n+k-1}}{\partial\mathbf{a}} \mid \dots \mid \frac{\partial z_{n+k-L+1}}{\partial\mathbf{a}} \right]^\top. \quad (2.26)$$

- $z_{n+k} = \hat{x}_{n+k}$.

Si l'échantillon est manquant

- prédire l'échantillon \hat{x}_{n+k} grâce à l'équation (2.1)
- les paramètres ne sont pas mis à jour : $\hat{\mathbf{a}}_{n+k} = \hat{\mathbf{a}}_{n+k-1}$
- calculer $\partial\hat{x}_{n+k}/\partial\mathbf{a}$ grâce à l'équation (2.7) pour $\mathbf{a} = \hat{\mathbf{a}}_{n+k}$
- actualiser le vecteur $\partial\mathbf{z}_{n+k}/\partial\mathbf{a}$ selon l'équation (2.26).

Cet algorithme est simple. De plus, grâce à l'utilisation d'une prédiction optimale au sens des moindres carrés, il converge vers une estimation non biaisée des paramètres. Par contre, cette méthode dépend fortement du schéma d'observation des échantillons. Ainsi, lorsque la perte des échantillons est aléatoire et de probabilité élevée ou lorsque l'ordre du modèle AR est élevé, les blocs de L échantillons consécutifs disponibles sont rarement formés. Par suite, le pas k de prédiction peut devenir arbitrairement élevé. Ceci conduit à une erreur quadratique moyenne de prédiction élevée et par conséquent à une large variance sur l'estimation des paramètres croissantes avec q , la probabilité de perte des échantillons. Or, l'erreur de prédiction obtenue en utilisant le prédicteur à passé infini incomplet est inférieure à celle obtenue en utilisant la prédiction à k -pas. Pour cela, nous proposons dans ce qui suit de l'utiliser dans un algorithme de type gradient.

2.2.3 Algorithmes de type gradient utilisant le prédicteur à passé incomplet

Puisque le prédicteur à passé infini incomplet est optimal au sens des moindres carrés, on propose de l'utiliser dans un algorithme de gradient pour une estimation non biaisée des paramètres et une reconstruction optimale au sens des moindres carrés. Les algorithmes résultants diffèrent de ceux décrits précédemment par l'approche de prédiction utilisée et par conséquent par le calcul du gradient du critère à minimiser. On s'intéresse dans ce qui suit au calcul du gradient de l'erreur quadratique de prédiction instantanée puisque le gradient des autres approximations du critère en découlent simplement grâce aux équations récursives (2.8) et (2.12).

On considère la représentation autorégressive du prédicteur donnée par l'équation (1.60). L'erreur quadratique instantanée de prédiction est alors donnée par :

$$J_n = \left(x_n - \sum_{i=0}^{n_M+L} c_{n-i} g_i x_{n-i} \right)^2 \quad (2.27)$$

La dérivée partielle du critère par rapport à un paramètre (a_j) est donc donnée par :

$$G_n = \frac{\partial J_n}{\partial a_j} = \sum_{i=0}^{n_M+L} c_{n-i} \frac{\partial g_i}{\partial a_j} x_{n-i} \quad (2.28)$$

où le terme $\partial g_i / \partial a_j$ est calculé en différentiant l'équation (1.59) par rapport au paramètre a_j , ce qui donne :

$$\frac{\partial g_i}{\partial a_j} = \sum_{s=0}^M \frac{\partial \psi_s}{\partial a_j} \sum_{h=0}^{n_s \wedge i} a_{n_s-h} a_{i-h} + \sum_{s=0}^M \psi_s \left[\frac{\partial \sum_{h=0}^{n_s \wedge i} a_{n_s-h} a_{i-h}}{\partial a_j} \right] \quad (2.29)$$

$$\text{où } \frac{\partial \sum_{h=0}^{n_s \wedge i} a_{n_s-h} a_{i-h}}{\partial a_j} = A(j, i, n_s) + A(j, n_s, i) \quad (2.30)$$

avec

$$A(j, i, l) = \begin{cases} a_{i-l+j} & \text{si } 0 < l - j < l \wedge i \\ 0 & \text{sinon} \end{cases} \quad (2.31)$$

Il est à noter ici que $a_0 = 1$ et que $a_j = 0$ pour $j > L$ ou $j < 0$.

Afin de calculer la dérivée partielle de ψ_s par rapport à un paramètre a_j , $\partial \psi_s / \partial a_j$, le système matriciel (1.55) est différentié par rapport à a_j . Ce qui conduit à :

$$U \frac{\partial (\psi_0, \dots, \psi_M)^\top}{\partial a_j} = - \frac{\partial U}{\partial a_j} (\psi_0, \dots, \psi_M)^\top. \quad (2.32)$$

La matrice $\partial U / \partial a_j$ est formée par la dérivée partielle de chacun des éléments de U par rapport au paramètre a_j . On obtient ainsi L matrices correspondant à la dérivée partielle de U par rapport à chacun des L paramètres du modèle AR. Il reste donc à calculer la dérivée de chacun des éléments de la matrice U par rapport a_i . Celle-ci est donnée de la même façon que dans (2.30) par :

$$\frac{\partial u_{vw}}{\partial a_j} = A(j, n_v, n_w) + A(j, n_w, n_v) \quad (2.33)$$

Ainsi, l'algorithme de gradient utilisant la prédiction à passé infini incomplet ne diffère des algorithmes présentés dans la section 2.1.5 que par la prédiction des échantillons qui est alors donnée par l'équation (1.58) et le calcul du gradient de la fonction de coût donné par (2.28). Puisque la prédiction ne peut pas être calculée récursivement, il s'en suit de même pour le calcul du gradient qui nécessite lui aussi l'inversion d'une matrice de la même taille que U .

Cet algorithme permet d'estimer sans biais les paramètres du processus et offre une reconstruction optimale au sens des moindres carrés des échantillons manquants. Pour autant, son coût calculatoire reste élevé.

2.3 Simulations et Comparaison des performances

Dans ce qui suit, les algorithmes du gradient en utilisant les différentes approches de prédiction discutées ci-dessus sont comparés à l'aide des différentes simulations suivantes. Nous nous intéressons dans les simulations en particulier aux algorithmes de gradient stochastique pour leur simplicité et pour leur capacité de poursuite dans des contextes non stationnaires.

On s'intéresse dans un premier temps au régime permanent des différents estimateurs. Pour cela le signal utilisé dans la première série de simulations est un processus AR stationnaire. Au cours de la deuxième simulation, un signal non stationnaire est utilisé pour tester les capacités de poursuite des différents algorithmes comparés.

Simulation 1 Le signal test utilisé est un processus AR d'ordre 2 dont les paramètres sont $[1.5, -0.7]$ générés sur 10^5 échantillons. La perte d'échantillons est considérée aléatoire suivant une loi de Bernoulli indépendante du signal de probabilité de perte $q = 0.3$. On considère un pas de convergence $\mu = 7.10^{-5}$. Cette valeur empirique permet un bon compromis entre la variance sur l'estimation des paramètres et la vitesse de convergence. Dans le cas de signaux à échantillons manquants ce facteur est choisi plus petit que dans le cas sans perte évitant ainsi une forte variance sur l'estimation des paramètres pouvant conduire à une divergence. Dans ce qui suit l'algorithme de "Mirsaidi", "A" et "B" désignent respectivement l'algorithme du gradient stochastique utilisant la prédiction par régression linéaire modifiée proposé par Mirsaidi *et al.* [53, 51] et ses alternatives utilisant la prédicteur à k -pas et le prédicteur à passé infini incomplet.

La figure 2.5 montre l'estimation du paramètre a_1 utilisant les trois algorithmes comparés. La figure 2.6 est un zoom sur le comportement des trois algorithmes après la convergence.

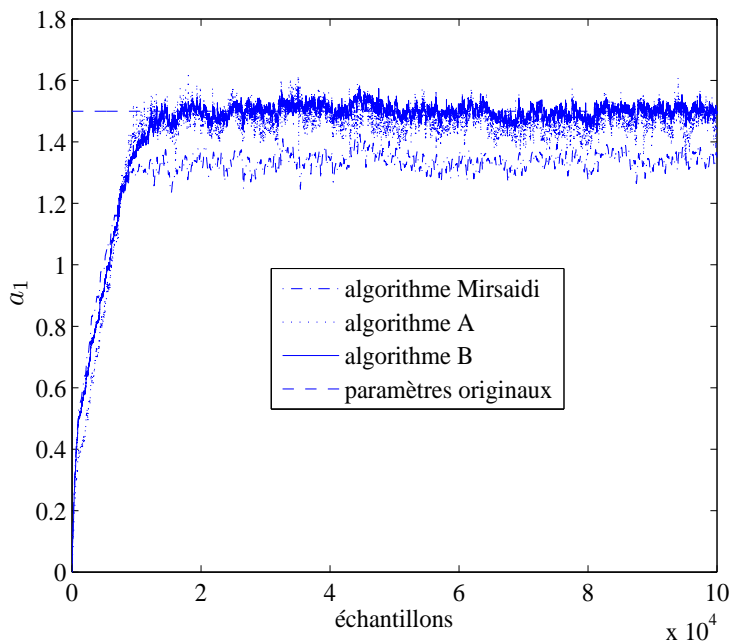


FIG. 2.5 – Comparaison de l'estimation du paramètre a_1 en utilisant les trois algorithmes.

Les performances des 3 algorithmes comparés sont résumés dans le tableau 2.1, où b_i désigne le biais sur l'estimation du paramètre a_i et σ_i l'écart type correspondant. Ils sont estimés après la convergence des algorithmes par les équations suivantes :

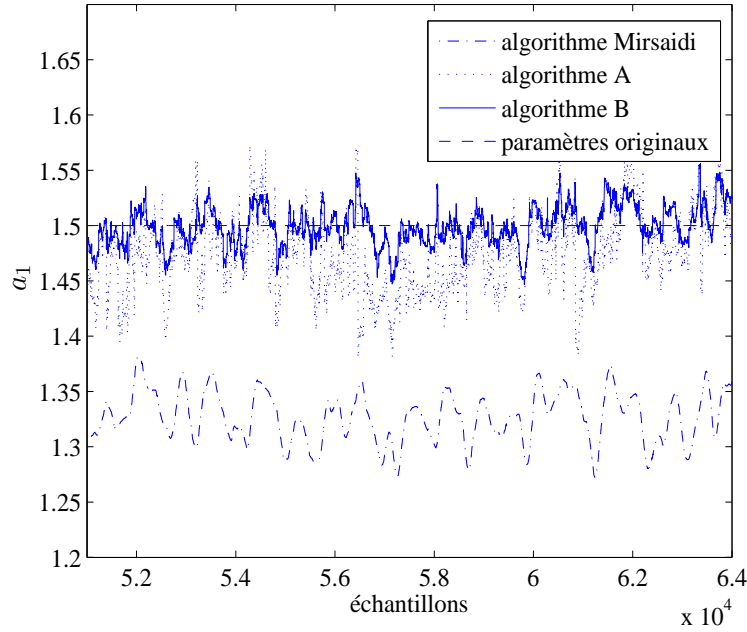


FIG. 2.6 – Estimation du paramètre a_1 après la convergence des trois algorithmes.

$$b_i = a_i - \bar{a}_i = a_i - \frac{1}{N - T_c + 1} \sum_{t=T_c}^N \hat{a}_{i,t} \quad (2.34)$$

$$\sigma_i = \frac{1}{N - T_c + 1} \sqrt{\sum_{t=T_c}^N (\hat{a}_{i,t} - \bar{a}_i)^2} \quad (2.35)$$

où T_c est un instant choisi après la convergence de l'algorithme (Cf. le temps de convergence au chapitre 1) et \bar{a}_i est la valeur de convergence en moyenne de l'estimateur du paramètre a_i .

L'EQMR désigne l'erreur quadratique moyenne de reconstruction normalisée du signal. La normalisation est faite par rapport à la puissance du signal. Ainsi pour un signal généré sur N échantillons, celle-ci est donnée par :

$$EQMR = \frac{\sum_{i=1}^N (x_i - z_i)^2}{\sum_{i=1}^N x_i^2}. \quad (2.36)$$

Le CPU est le temps de calcul en secondes nécessaire à la simulation de chacun des trois algorithmes pour le signal test décrit ci-dessus implémentés sous MATLAB sur un processeur de 3GHz.

Le tableau 2.1 et les figures 2.5 et 2.6 montrent que l'algorithme Mirsaidi converge comme prévu vers une estimation biaisée des paramètres alors que celle-ci l'est bien moins (théoriquement pas) en utilisant les deux autres algorithmes. De plus, le tableau 2.1 montre que l'algorithme B est coûteux en temps de calcul. Les simulations montrent que son CPU croît exponentiellement avec la probabilité q de perte des échantillons. Ceci est notamment dû à l'augmentation des dimensions des matrices à inverser lorsque la probabilité q augmente. Par contre, les algorithmes Mirsaidi et A sont à peu près huit fois plus rapide, et leur CPU n'augmente pas avec q . D'autre part,

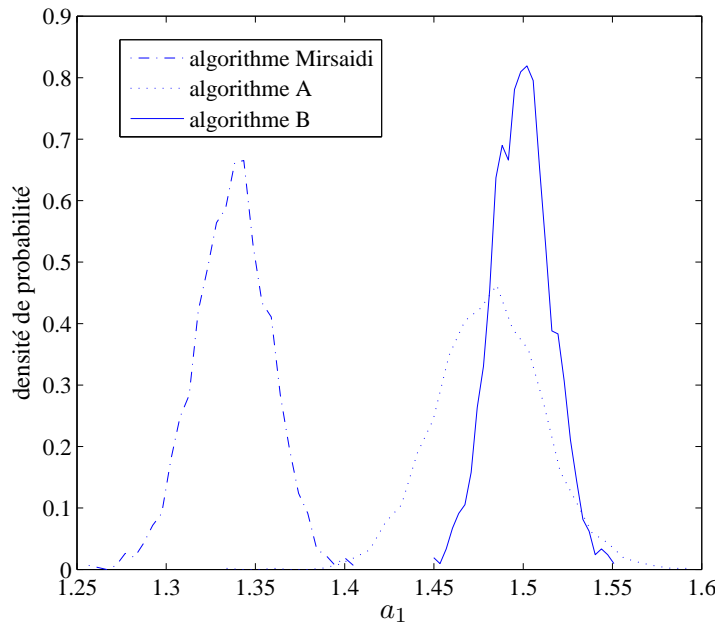
TAB. 2.1 – Performances des 3 algorithmes comparés pour la simulation 1.

Algorithme	b_1	b_2	σ_1	σ_2
Mirsaidi	0.166	0.137	0.019	0.018
A	0.026	0.014	0.029	0.0298
B	0.004	0.009	0.014	0.016

Algorithme	EQMR	CPU
Mirsaidi	0.07	5
A	0.11	4
B	0.061	38

l'algorithme B offre comme prévu l'erreur de prédiction la plus faible par comparaison à celles obtenues grâce aux autres algorithmes alors que l'algorithme A souffre de la plus élevée. Il est à noter que l'EQMR fourni par l'algorithme Mirsaidi est proche de celui fourni par l'algorithme B pour un temps de calcul beaucoup plus faible.

La figure 2.7 montre pour chacun des algorithmes comparés, une densité de probabilité empirique de l'estimation du paramètre a_1 . Pour cela, le paramètre a_1 est estimé en utilisant les trois algorithmes comparés pour chacune des 2000 réalisations du signal et du schéma de perte. Ainsi, pour chacun des algorithmes, une densité de probabilité empirique de l'estimation du paramètre a_1 est déduite à un instant T_c choisi après la convergence.

FIG. 2.7 – Densité de probabilité empirique de l'estimation du paramètre a_1 utilisant les trois algorithmes

La figure 2.7 montre que la densité de probabilité de l'estimation du paramètre a_1 utilisant l'algorithme B est bien centrée sur 1.5. Ceci confirme que cet estimateur n'est pas biaisé. D'autre part, on remarque que l'estimation du paramètre a_1 utilisant l'algorithme A est légèrement biaisée, la densité de probabilité de cet estimateur étant centrée sur 1.49. Or, cet estimateur

est supposé être non biaisé puisque l'approche de prédiction utilisée est optimale au sens des moindres carrés. Ceci peut être dû à la grande variance sur l'estimation des paramètres utilisant cet algorithme et à l'estimation empirique de la densité de probabilité qui est limitée à 2000 réalisations du signal et du schéma de perte. Par contre, pour l'algorithme Mirsaidi, la densité de probabilité de l'estimation de a_1 est centrée sur 1.34 ce qui montre que cet estimateur souffre d'un biais.

La figure 2.7 nous donne aussi des indications sur la variance de l'estimation de chacun des algorithmes. En effet, se référant au tableau 2.1 et aux figures 2.6 et 2.7, pour les algorithmes de type gradient comparés, la variance sur l'estimation des paramètres est liée, comme prévu, à l'erreur de prédiction du signal. En effet, pour les algorithmes de type gradient comparés, la mise à jour des paramètres est proportionnelle à l'erreur de prédiction telle qu'on le voit à travers les équations (1.12) et (2.4). Ainsi, la variance sur l'estimation des paramètres utilisant l'algorithme B est le plus faible alors que celle correspondant à l'algorithme A est la plus élevée.

Finalement, la figure 2.8 présente l'EQMR obtenu pour la reconstruction du signal test décrit ci-dessus utilisant chacun des trois algorithmes en fonction de la probabilité de perte des échantillons q .

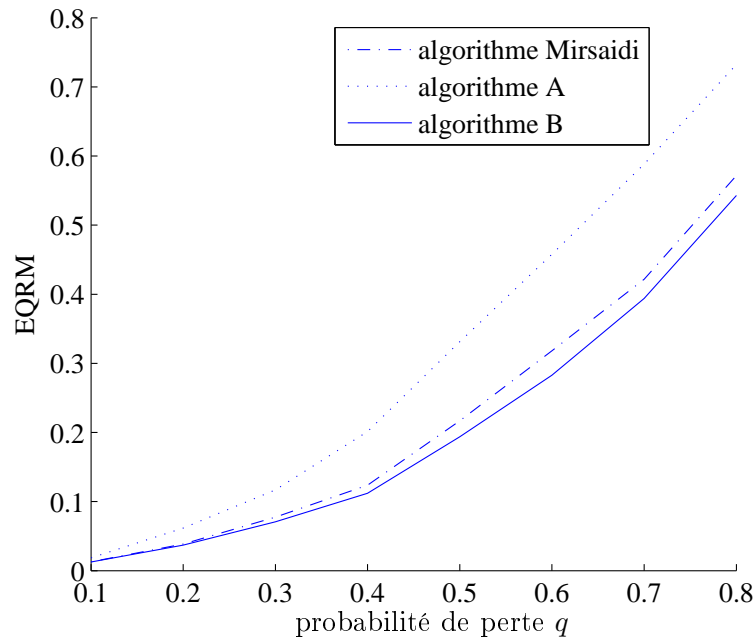


FIG. 2.8 – EQMR des trois algorithmes en fonction de la probabilité de perte q

On constate grâce à la figure 2.8 que l'erreur de reconstruction obtenue en utilisant l'algorithme de Mirsaidi reste proche de l'erreur optimale même pour des probabilités de pertes élevées.

Simulation 2 Le signal test utilisé ici est un processus AR non stationnaire composite généré sur $5 \cdot 10^4$ échantillons. Pour les premiers $25 \cdot 10^3$ échantillons, la valeur des paramètres est $[1.5; -0.7]$, alors que pour les échantillons restants, les paramètres sont $[1; -0.5]$. La perte d'échantillons est considérée aléatoire suivant une loi de Bernoulli indépendante du signal de probabilité $q = 0.3$. Les figures 2.10 et 2.9 montrent l'estimation moyenne du paramètre a_1 en utilisant

les algorithmes A et B pour différentes valeurs du pas μ . L'estimation instantanée moyenne est déduite grâce à un Monte Carlo de 1000 réalisations du signal et du schéma de perte. Ainsi pour chaque réalisation, les paramètres sont estimés en utilisant les deux algorithmes pour différentes valeurs du pas μ .

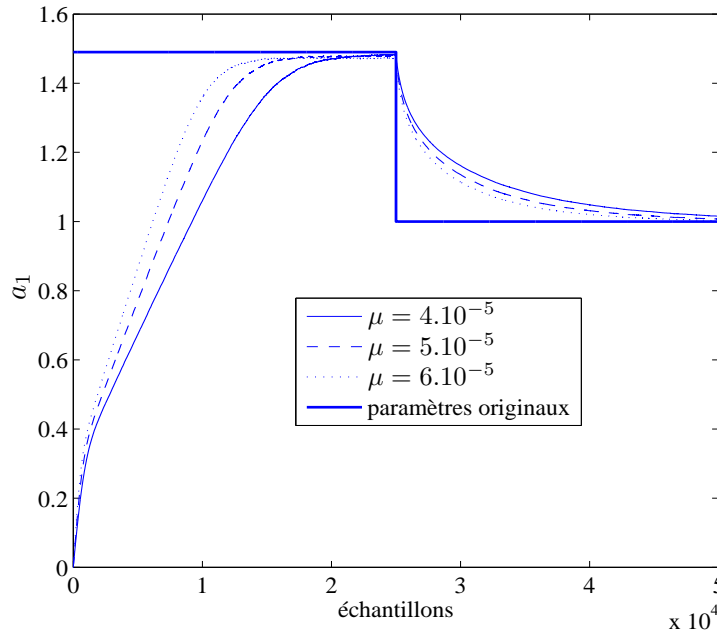


FIG. 2.9 – Estimation instantanée moyenne du paramètre a_1 pour différentes valeurs de μ grâce à l'algorithme de gradient utilisant le prédicteur à k -pas.

L'algorithme A est simulé pour les valeurs de μ suivantes : 4.10^{-5} , 5.10^{-5} et 6.10^{-5} . Pour des valeurs supérieures à cette dernière, l'estimation des paramètres utilisant cet algorithme diverge pour certains schémas de perte. Par contre, pour l'algorithme B les valeurs de μ sont choisies plus élevées sans risquer une divergence de l'estimation des paramètres : 1.10^{-4} , 2.10^{-4} et 25.10^{-5} . Ceci s'explique par ce qui suit :

En augmentant la valeur de μ , la vitesse de convergence augmente par contre ceci se fait aux dépens d'une augmentation de la variance sur l'estimation des paramètres qui peut mener à partir d'un certain seuil à la divergence de l'algorithme. Or, l'erreur de prédiction du prédicteur à k -pas est plus élevée que celle du prédicteur à passé infini incomplet. La variance sur l'estimation des paramètres en utilisant l'algorithme A est donc plus élevée, pour un même μ , que celle obtenue en utilisant l'algorithme B. Pour cela, il est possible de simuler l'algorithme B avec des valeurs de μ supérieures à celle tolérée par l'algorithme A avant d'atteindre le seuil de divergence.

Les figures 2.9 et 2.10 montrent l'évolution de la vitesse de convergence avec le pas μ . On remarque de plus qu'on peut obtenir en utilisant l'algorithme B de meilleures vitesses de convergence en moyenne que celles permises en utilisant l'algorithme A. Ainsi, l'algorithme B converge en moyenne vers la bonne valeur du paramètre à partir de l'instant 8000 et ceci pour $\mu = 25.10^{-5}$. Par contre, pour l'algorithme A, la convergence en moyenne la plus rapide est obtenue pour $\mu = 6.10^{-5}$, pour laquelle la convergence se fait à partir de l'instant 15000.

Il est à noter ici la faiblesse des algorithmes du gradient en vitesse de convergence et en vitesse d'adaptation aux variations de paramètres. Une analyse de la convergence des algorithmes de

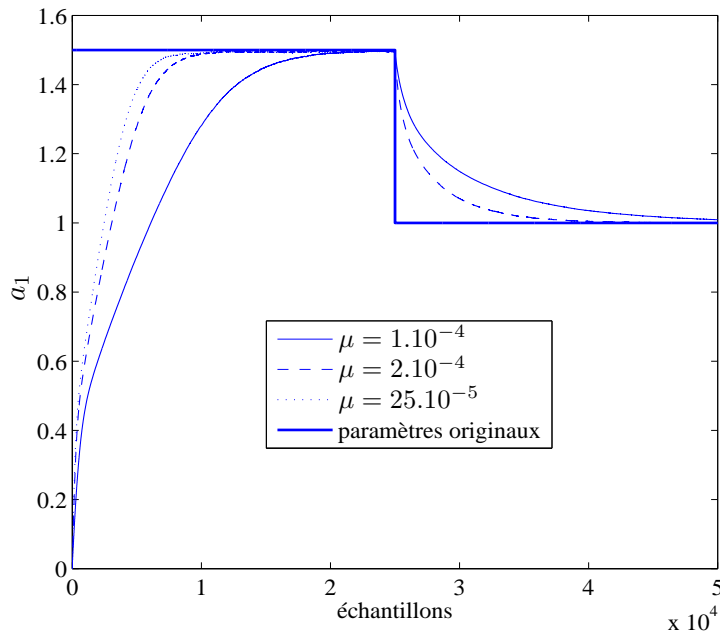


FIG. 2.10 – Estimation instantanée moyenne du paramètre a_1 pour différentes valeurs de μ grâce à l’algorithme de gradient utilisant le prédicteur à passé infini incomplet.

gradient est présentée dans [24, 44, 45]. On en déduit que la vitesse de convergence est proportionnelle à μ , par contre celle-ci n’est pas uniforme sur l’espace des paramètres. Ainsi, pour une direction donnée correspondant à l’un des vecteurs propres de la matrice de covariance du signal, le temps de convergence dépend de la valeur propre correspondant à ce vecteur propre. Celle-ci dépend aussi des paramètres (et donc de leur nombre) et de la puissance du bruit d’innovation. L’expression du temps de convergence moyen de l’algorithme du gradient stochastique est donnée au chapitre 1 par l’équation (1.15). Dans le cas de signaux à échantillons manquants, la vitesse de convergence en moyenne dépend aussi de la probabilité de perte des échantillons. Le temps de convergence en moyenne augmente avec la probabilité de perte des échantillons.

2.4 Conclusions

L’extension des algorithmes de gradient à l’identification adaptative de processus AR à échantillons manquants est étudiée dans ce chapitre. Mirsaidi *et al.* [51, 52, 53] ont proposé dans cette optique un ensemble d’algorithmes de gradient utilisant la prédiction par régression linéaire modifiée. Ces algorithmes sont simples et permettent un traitement temps réel des signaux. Par contre, ils convergent vers une estimation biaisée des paramètres. Une expression du biais d’identification par le principe des moindres carrés est calculé par Wallin *et al.* [68]. Ils trouvent en plus une condition pour une identification non biaisée. On en déduit que l’utilisation d’une prédiction optimale au sens des moindres carrés permet une estimation sans biais des paramètres. Ainsi, de nouvelles alternatives fondées sur deux autres approches de prédiction sont proposées afin de résoudre le problème du biais d’identification. Le premier algorithme est basé sur la prédiction à k -pas. Cet algorithme est simple et permet une estimation non biaisée des paramètres. Par contre, il présente une erreur quadratique de reconstruction élevée qui mène à une large variance

sur l'estimation des paramètres. La deuxième alternative est basée sur la prédiction à passé infini incomplet. Cet algorithme permet, à la fois, une estimation sans biais des paramètres et une reconstruction optimale au sens des moindres carrés. On remarque ceci en se référant à la figure 2.8 où l'erreur quadratique de reconstruction obtenue grâce à l'algorithme B est effectivement minimale par rapport aux autres approches comparées. Par contre, cet algorithme souffre d'une complexité de calcul due à l'inversion de matrices à chaque pas de temps.

En conclusion, l'analyse des trois algorithmes montre que l'algorithme Mirsaidi, bien qu'il ne convienne pas à l'identification, peut être utile pour une reconstruction en temps réel du signal. L'algorithme A convient davantage à une identification en temps réel. Pour les applications où le temps de calcul n'est pas une contrainte, l'algorithme B offre une identification non biaisée et une reconstruction optimale au sens des moindres carrés. Puisque les trois algorithmes se basent sur le principe du gradient, ils présentent une faible vitesse de convergence. Or, l'un des grands avantages de l'algorithme des moindres carrés récursifs est sa vitesse de convergence. Il permet de plus un bon suivi de l'évolution des signaux non stationnaires grâce à l'introduction dans la fonction de coût d'un facteur d'oubli [43, 44, 45]. Pour cela, nous étudions dans le chapitre suivant l'extension de l'algorithme des moindres carrés récursifs au cas de signaux à échantillons manquants.

Chapitre 3

Algorithme MCR pseudo linéaire

3.1 Extension de l'algorithme MCR au cas de signaux à échantillons manquants

Nous avons mentionné dans le chapitre précédent la faiblesse des algorithmes de gradient en vitesse de convergence. Pour cela, on s'intéresse dans ce chapitre à l'extension de l'algorithme des moindres carrés récursifs à l'identification de signaux à échantillons manquants.

L'algorithme des moindres carrés récursifs (MCR) permet d'estimer récursivement les paramètres du modèle AR qui minimisent à chaque instant la moyenne statistique de toutes les erreurs quadratiques de prédiction précédentes. Pour donner à l'algorithme la capacité de suivre l'évolution des paramètres, les erreurs de prédiction sont pondérées par un facteur d'oubli exponentiel. Les équations récursives de l'algorithme MCR sont données à l'instant $n + 1$ par les équations (1.17). L'un des avantages de cet algorithme est que la constante de temps ne dépend que du facteur d'oubli selon l'équation (1.18). Il offre donc une bonne vitesse de convergence et un bon suivi des variations des paramètres pour les signaux non stationnaires [9, 44].

Dans ce qui suit, on commence par présenter l'extension des moindres carrés à l'identification de signaux à échantillons manquants en utilisant une régression pseudo-linéaire. Par la suite, l'algorithme des moindres carrés récursifs pseudo-linéaires est présenté et analysé.

3.1.1 Régression pseudo-linéaire

Le critère optimisé par l'algorithme des moindres carrés récursifs est donné à l'instant $n + 1$ par :

$$J_{n+1}(\mathbf{a}) = \sum_{k=L+1}^{n+1} \lambda^{n+1-k} (x_k - \hat{x}_k)^2 = \sum_{k=L+1}^{n+1} \lambda^{n+1-k} \left(x_k - \mathbf{a}^\top \mathbf{x}_{k-1} \right)^2. \quad (3.1)$$

où $0 < \lambda \leq 1$ est un facteur d'oubli permettant de donner plus d'importance aux valeurs récentes dans le traitement de signaux non stationnaires, $\hat{x}_k = \mathbf{a}^\top \mathbf{x}_{k-1}$ est la prédiction par régression linéaire de l'échantillon x_k .

Dans le cas de signaux à échantillons manquants, les échantillons manquants du vecteur de régression \mathbf{x}_n sont remplacés par leurs prédictions. Le vecteur de régression ainsi obtenu est dépendant des paramètres du modèle ce qui conduit à une prédiction non linéaire par rapport aux paramètres. Il est alors donné à l'instant $n + 1$ par :

$$\hat{\Psi}_{n+1} = \mathbf{z}_n = [z_n \dots z_{n-L+1}]^\top \quad (3.2)$$

où $\{z_n\}$ est le signal reconstruit donné pour tout n par (1.34), il prend la valeur du signal lorsque l'échantillon est disponible et sa prédiction dans le cas contraire.

En outre, l'erreur de prédiction n'est pas définie aux instants où les échantillons sont manquants. Le critère minimisé est alors la moyenne pondérée des erreurs quadratiques de prédiction aux instants où les échantillons sont disponibles. La question du choix de la pondération de ces erreurs se pose. La variation des paramètres d'un signal non stationnaire étant fonction du temps, on propose de pondérer les erreurs en tenant compte de leurs éloignements temporels de l'instant d'estimation des paramètres et non du nombre d'échantillons disponibles qui les séparent de celui-ci. Le critère s'exprime alors par :

$$J_{n+1}(\mathbf{a}) = \sum_{k=L+1}^{n+1} c_k \lambda^{n+1-k} (x_k - \hat{x}_k)^2 \quad (3.3)$$

où la pondération par la modulation d'amplitude c_k permet de ne considérer dans le critère que les erreurs de prédiction aux instants où les échantillons sont disponibles.

La fonction de coût précédente s'écrit sous la forme vectorielle suivante :

$$J_{n+1}(\mathbf{a}) = \left\{ (\mathbf{x} - Z_{n+1}\mathbf{a})^\top (C_{n+1}\Lambda_{n+1})(\mathbf{x} - Z_{n+1}\mathbf{a}) \right\} \quad (3.4)$$

où le vecteur $\mathbf{x} = [x_{L+1}, \dots, x_{n+1}]^\top$ est constitué des échantillons du signal de l'instant $L+1$ jusqu'à l'instant $n+1$. Les matrices C_{n+1} et Λ_{n+1} sont diagonales constituées respectivement des modulations d'amplitudes binaires jusqu'à l'instant $n+1$ et des pondérations par le facteur d'oubli à l'instant $n+1$. La matrice Z_{n+1} est dite de régression, ses dimensions sont $(n-L+1) \times (L)$, elle est constituée de tous les vecteurs de régression du signal reconstruit jusqu'à l'instant $n+1$. Ces trois matrices sont données à l'instant $n+1$ par :

$$C_{n+1} = \begin{bmatrix} c_{L+1} & & 0 \\ & \ddots & \\ 0 & & c_{n+1} \end{bmatrix}, \quad \Lambda_{n+1} = \begin{bmatrix} \lambda^{n-L} & & 0 \\ & \ddots & \\ 0 & & \lambda & \\ & & & 1 \end{bmatrix} \quad \text{et} \quad Z_{n+1} = \begin{bmatrix} z_L \\ \vdots \\ \vdots \\ z_n \end{bmatrix}^\top$$

Une analyse de l'erreur quadratique moyenne de prédiction dans le cas de signaux à échantillons manquants est faite au paragraphe 2.1.1 du chapitre 2. Elle reste valable pour la fonction de coût donnée par l'équation (3.4). Ainsi le critère à minimiser admet un seul minimum global lorsque la perte des échantillons est aléatoire. En négligeant la dépendance de la matrice Z_{n+1} par rapport aux paramètres, l'estimée pseudo-linéaire des paramètres qui minimisent le critère (3.4) est alors donnée par :

$$\hat{\mathbf{a}}_{n+1} = \left(Z_{n+1}^\top \Lambda_{n+1} C_{n+1} Z_{n+1} \right)^{-1} Z_{n+1}^\top \Lambda_{n+1} C_{n+1} \mathbf{x} \quad (3.5)$$

On note $\hat{R}_{z,n+1}$ la matrice de dimensions $L \times L$ et $\hat{\mathbf{r}}_{z,n+1}$ le vecteur de longueur L , définis à l'instant $n+1$ comme suit :

$$\hat{R}_{z,n+1} = \left(Z_{n+1}^\top \Lambda_{n+1} C_{n+1} Z_{n+1} \right) = \sum_{k=L+1}^{n+1} c_k \lambda^{n+1-k} \mathbf{z}_k \mathbf{z}_k^\top \quad (3.6)$$

$$\hat{\mathbf{r}}_{xz,n+1} = \mathbf{Z}_{n+1}^\top \Lambda_{n+1} \mathbf{C}_{n+1} \mathbf{x} = \sum_{k=L+1}^{n+1} c_k \lambda^{n+1-k} x_k \mathbf{z}_k \quad (3.7)$$

Or, se référant à l'équation (3.7), les termes $\lambda^{n+1-k} x_k \mathbf{z}_k$ ne sont pris en considération dans le calcul de $\hat{\mathbf{r}}_{xz,n+1}$ que lorsque $c_k = 1$ et par la suite lorsque $x_k = z_k$. Le vecteur $\hat{\mathbf{r}}_{xz,n+1}$ est alors égal à $\hat{\mathbf{r}}_{z,n+1}$ donné par $\hat{\mathbf{r}}_{z,n+1} = \sum_{k=L+1}^{n+1} c_k \lambda^{n+1-k} z_k \mathbf{z}_k$. Le vecteur des paramètres qui minimise le critère (3.4) s'écrit alors sous la forme suivante :

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{R}}_{z,n+1}^{-1} \hat{\mathbf{r}}_{z,n+1}. \quad (3.8)$$

On montre, au paragraphe 3.3.2 de ce chapitre, que lorsque le facteur d'oubli est égal à 1, les paramètres calculés via l'équation (3.8) sont égaux asymptotiquement à la solution selon les moindres carrés des paramètres qui caractérisent le signal reconstruit (où les échantillons manquants sont remplacés par leur prédiction). Par contre, les paramètres qui minimisent les moindres carrés pondérés par un facteur d'oubli exponentiel sont différents en moyenne de ceux qui minimisent les moindres carrés sans pondération. La convergence en moyenne des paramètres AR estimés grâce à l'algorithme MCR pseudo-linéaire sera discutée ultérieurement au paragraphe 3.3.2 de ce chapitre.

On s'intéresse dans le cadre de cette thèse à l'identification adaptative des signaux à échantillons manquants. Pour cela on présente dans ce qui suit l'algorithme des moindres carrés récursifs pseudo-linéaires qui permet de calculer récursivement les paramètres donnés par l'équation (3.8).

3.1.2 Algorithme des MCR pseudo-linéaire

L'estimée du vecteur des paramètres donnée par l'équation (3.5) peut être rendue récursive en utilisant la même approche que celle utilisée pour le développement de l'algorithme des moindres carrés récursifs. Ainsi, en exploitant la mise à jour récursive de la matrice $\hat{\mathbf{R}}_{z,n+1}$ et en appliquant le lemme d'inversion matriciel [29], la matrice inverse de $\hat{\mathbf{R}}_{z,n+1}$, notée \mathbf{G}_{n+1} , est calculée récursivement à partir de sa valeur à l'instant précédent. Ainsi, on aboutit à l'algorithme des moindres carrés récursifs pseudo-linéaires donné par les équations suivantes :

$$\hat{\Psi}_{n+1} = \mathbf{z}_n = [z_n \dots z_{n-L+1}]^\top, \quad (3.9a)$$

$$\hat{\mathbf{x}}_{n+1} = \hat{\Psi}_{n+1}^\top \hat{\mathbf{a}}_n, \quad (3.9b)$$

$$\gamma_{n+1} = \frac{c_{n+1} \mathbf{G}_n \hat{\Psi}_{n+1}}{\lambda + \hat{\Psi}_{n+1}^\top \mathbf{G}_n \hat{\Psi}_{n+1}}, \quad (3.9c)$$

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n + \gamma_{n+1} (x_{n+1} - \hat{\mathbf{a}}_n^\top \hat{\Psi}_{n+1}), \quad (3.9d)$$

$$\mathbf{G}_{n+1} = \frac{1}{\lambda} (\mathbf{I}_d - \gamma_{n+1} \hat{\Psi}_{n+1}^\top) \mathbf{G}_n \quad (3.9e)$$

où \mathbf{I}_d est la matrice identité et le vecteur $\gamma_{n+1} = \mathbf{G}_{n+1} \mathbf{z}_n$ est le gain d'adaptation de l'algorithme MCR à l'instant $n + 1$.

L'identification de processus ARX à échantillons manquants par le principe des moindres carrés récursifs pseudo-linéaire est étudiée dans [1, 62, 63]. L'algorithme de base est généralisé dans [62, 63] par l'introduction de plusieurs prédicteurs possibles à la place de la prédiction par régression linéaire modifiée. Ainsi l'équation (3.9b) est remplacée par l'équation suivante :

$$\hat{x}_{n+1} = f(\hat{\mathbf{a}}_n, \mathbf{z}_n) \quad (3.10)$$

où f représente l'approche de prédiction utilisée en fonction de l'estimation actuelle des paramètres et des données précédentes disponibles ou prédites.

Pour le fonctionnement de cet algorithme, nous supposons disposer des L premiers échantillons. Ainsi, l'algorithme MCR pseudo-linéaire se résume aux étapes suivantes :

Initialisation à l'instant L :

- Le vecteur des paramètres est initialisé à zéro : $\hat{\mathbf{a}}_L = [0, \dots, 0]$;
- La matrice G_L est une matrice diagonale dont les éléments sont de valeur élevée (afin d'éliminer rapidement l'effet des conditions initiales supposées fausses) : $G_L = \alpha I_d$ où α est une constante positive généralement élevée (de l'ordre de 10^4 par exemple) et I_d est la matrice identité de dimensions $L \times L$.
- On suppose que les L premiers échantillons sont disponibles, le vecteur de régresseurs est alors initialisé à : $\hat{\Psi}_{L+1} = \mathbf{x}_L = [x_L, \dots, x_1]^\top$.
- Le facteur d'oubli λ est généralement choisi très proche de 1 (par exemple 0.99).

A l'instant $n + 1$ pour $L + 1 \leq n + 1 \leq N$, l'algorithme se résume aux étapes suivantes :

- Le vecteur de régression est mis à jour en utilisant la donnée précédente ou sa prédiction si celle-ci est manquante via l'équation (3.9a).
- L'échantillon x_{n+1} est prédit en fonction des données précédentes et l'estimation actuelle des paramètres.

Si l'échantillon x_{n+1} est disponible, i.e. $c_{n+1} = 1$,

- Le gain d'adaptation à l'instant $n + 1$ est calculé par l'équation (3.9c) ;
- Le vecteur des paramètres est mis à jour via l'équation (3.9d) ;
- La matrice G_{n+1} , inverse de $\hat{R}_{z,n+1}$, est mise à jour grâce à l'équation (3.9e).

Si l'échantillon x_{n+1} est manquant, i.e. $c_{n+1} = 0$

- Le gain de correction est nul : $\gamma_{n+1} = 0$;
- Les paramètres sont conservés : $\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n$;
- Le facteur d'oubli agit quand même sur la matrice G_{n+1} qui est alors mise à jour par :

$$G_{n+1} = \frac{1}{\lambda} G_n. \quad (3.11)$$

3.1.3 Observations et analyse des performances

L'algorithme des moindres carrés récursifs pseudo-linéaires est simple bien que d'une complexité de calcul plus grande que celle des algorithmes du gradient. De plus, il offre un meilleur temps de convergence que ceux fondés sur le gradient et un bon suivi des paramètres. Le temps de convergence en moyenne de l'algorithme MCR pseudo-linéaire dépend du facteur d'oubli et de la probabilité de perte des échantillons. Celui-ci est croissant avec la probabilité de perte. L'étude du temps de convergence en moyenne en fonction du facteur d'oubli et de la probabilité de perte des échantillons est réalisée au paragraphe 3.4, consacré aux simulations.

D'autre part, lorsque les échantillons manquants sont remplacés par leur prédiction par régression linéaire modifiée, l'algorithme converge typiquement vers une estimation biaisée des paramètres. Wallin *et al.* [69] calculent une expression du biais d'identification par application des moindres carrés sur le signal reconstruit (où les échantillons manquants sont remplacés par leurs prédictions). Ils en déduisent une condition pour une identification sans biais donnée par l'équation (1.50). Selon cette condition si l'approche de prédiction utilisée minimise l'erreur quadratique moyenne de prédiction, l'identification est non biaisée. Or, les paramètres estimés par l'algorithme MCR pseudo-linéaire convergent en moyenne vers ceux estimés par l'approche proposée par Wallin *et al.* lorsque le facteur d'oubli est égal à 1. Pour cela, on propose dans ce qui suit d'utiliser avec l'algorithme MCR pseudo-linéaire une approche de prédiction optimale au sens des moindres carrés pour résoudre le problème de biais. L'analyse de convergence en moyenne des paramètres estimés grâce à l'algorithme résultant sera réalisée au paragraphe 3.3.2 de ce chapitre.

Dans [11, 13], Bondon propose une expression analytique de la projection orthogonale d'un échantillon sur son passé infini incomplet ainsi qu'une représentation autorégressive de celui-ci en fonction de toutes les observations précédentes. Cette approche de prédiction a été décrite au paragraphe 2 du chapitre 1. Au paragraphe 2.2.1 du chapitre 2, on montre que dans le cas particulier des processus AR d'ordre fini L , la prédiction se fait en fonction de toutes les observations précédentes jusqu'à ce que L échantillons consécutifs disponibles soient rencontrés. Par contre, cette approche de prédiction n'est pas récursive et nécessite à chaque instant l'inversion de matrices dont les dimensions dépendent du schéma de perte, ce qui la rend complexe en calcul. Par ailleurs, d'autres approches de prédiction se basent sur une représentation d'état du processus lorsque le modèle choisi en admet une. L'intérêt d'une telle représentation est qu'une estimation optimale en moyenne quadratique de l'état peut être obtenue grâce à un filtre de Kalman. Nous avons cité dans le paragraphe 1.3.2 du chapitre 1 les principaux filtres de Kalman utilisés dans la littérature. Nous nous intéressons particulièrement ici au système d'état décrit par Fletcher *et al.* [27]. Un processus AR d'ordre fini L à échantillons manquants peut être considéré comme un cas particulier du système décrit par Fletcher *et al.* [27]. Une prédiction récursive de celui-ci, optimale au sens des moindres carrés, est donc possible grâce à un filtre de Kalman. Nous l'utilisons alors conjointement avec l'algorithme MCR pseudo-linéaire pour une identification sans biais des paramètres et une reconstruction récursive optimale au sens des moindres carrés du processus AR(L) à échantillons manquants.

3.2 Prédiction en utilisant un filtre de Kalman

Dans ce qui suit, nous commençons par définir une représentation d'état du processus AR(L) à échantillons manquants. Par suite, les équations du filtre de Kalman permettant l'estimation de l'état, dont le premier élément est l'échantillon à prédire, sont présentées et analysées.

3.2.1 Représentation d'état d'un processus AR à échantillons manquants

Dans [27], Fletcher *et al.* présentent le problème de l'estimation d'un processus à partir de mesures obtenues grâce à un capteur à perte d'échantillons. Ils le considèrent comme un cas particulier du problème général de l'estimation de l'état dans les systèmes linéaires à sauts markoviens. Ainsi, le signal mesuré par le capteur est supposé être la sortie d'un système linéaire stochastique. Le système considéré est représenté à la figure 3.1. La représentation d'état du système linéaire considéré et les équations du filtre de Kalman correspondant sont présentés au paragraphe 1.3.2 du chapitre 1. Un processus AR(L) à échantillons manquants peut être considéré comme la sortie d'un système tel que celui représenté à la figure 3.1 lorsque le bruit d'observation est annulé et l'effet du capteur est limité à la mise à zéro des échantillons manquants. Ceci revient à considérer dans les équations (1.61), (1.62) ce qui suit :

$$\begin{aligned} D &= 0 \\ C_c &= 1 \\ D_c &= 0 \end{aligned}$$

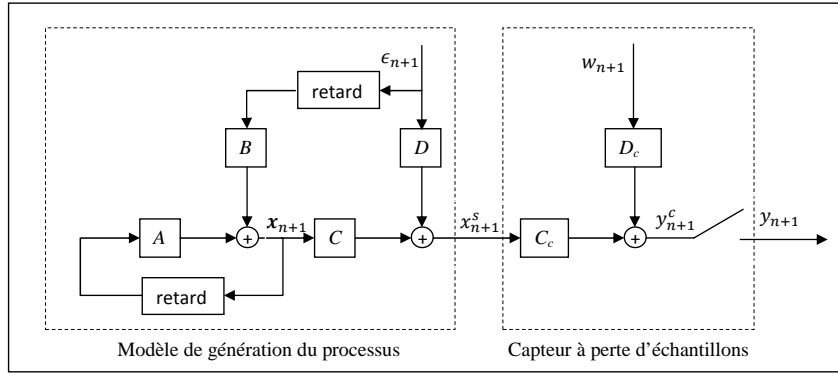


FIG. 3.1 – Modélisation du système global.

Ainsi, le processus AR(L) à échantillons manquants, $\{x_n\}$, dont l'observation $\{y_n\}$ est donnée par l'équation (1.23), admet la représentation d'état suivante :

$$\begin{cases} \mathbf{x}_{n+1} = A \mathbf{x}_n + B \epsilon_n \\ x_{n+1}^s = C \mathbf{x}_{n+1} \\ y_{n+1} = c_{n+1} C \mathbf{x}_{n+1} \end{cases} \quad (3.12)$$

où A est la matrice d'état de dimensions $L \times L$, B , C et \mathbf{x}_n sont des vecteurs de dimensions L . Ils sont donnés par :

$$A = \left[\begin{array}{ccc|c} a_1 & \dots & \dots & a_L \\ 1 & & & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \end{array} \right], B = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, C = B^\top \quad \text{et} \quad \mathbf{x}_n = \begin{bmatrix} x_n \\ \vdots \\ x_{n-L+1} \end{bmatrix}.$$

La sortie du système x_{n+1}^s est donc l'échantillon x_{n+1} qu'on souhaite prédire.

3.2.2 Filtre de Kalman

La principale motivation pour une représentation d'état du processus $AR(L)$ à échantillons manquants est que l'estimation optimale au sens des moindres carrés de l'état peut être obtenue grâce à un filtre de Kalman. Le filtre de Kalman est classique, il est décrit dans de nombreux textes dont [14, 70]. Il permet l'estimation de l'état minimisant le critère suivant :

$$J_{n+1|n} = E \left\{ |x_{n+1}^s - \hat{x}_{n+1|n}^s|^2 \right\} \quad (3.13)$$

où $\hat{x}_{n+1|n}^s$ est l'estimée *a priori* de la sortie du système donc en fonction des observations précédentes. C'est donc la prédiction de l'échantillon x_{n+1} sachant son passé disponible.

Ainsi, on note $\hat{\mathbf{x}}_{n+1|n}$ et $\hat{\mathbf{x}}_{n+1|n+1}$ les estimées *a priori* et filtrée de l'état. Elles sont définies par les espérances conditionnelles suivantes :

$$\begin{aligned} \hat{\mathbf{x}}_{n+1|n} &= E \{ \mathbf{x}_{n+1} | y_1, \dots, y_n \} \\ \hat{\mathbf{x}}_{n+1|n+1} &= E \{ \mathbf{x}_{n+1} | y_1, \dots, y_{n+1} \} \end{aligned}$$

Les matrices de covariance de l'erreur d'estimation correspondantes sont notées respectivement $P_{n+1|n}$ et $P_{n+1|n+1}$. Elles sont définies par :

$$P_{n+1|n} = E \left\{ (\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1|n}) (\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1|n})^\top \right\} \quad (3.14)$$

$$P_{n+1|n+1} = E \left\{ (\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1|n+1}) (\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1|n+1})^\top \right\} \quad (3.15)$$

Le développement des équations du filtre de Kalman standard peut être trouvé dans [14, 70].

Ici, le bruit d'observation est nul. De plus, il faut distinguer deux cas correspondant à la disponibilité ou non de l'échantillon. Ainsi, les équations du filtre de Kalman permettant l'estimation de l'état défini par l'équation (3.12), se résument à tout instant $n + 1$ à :

$$P_{n+1|n} = AP_{n|n}A^\top + R_\epsilon, \quad (3.16)$$

$$\text{où, } R_\epsilon = \sigma_\epsilon^2 BB^\top = \begin{bmatrix} \sigma_\epsilon^2 & 0 & \dots & 0 \\ 0 & 0 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix}$$

$$\hat{\mathbf{x}}_{n+1|n} = A\hat{\mathbf{x}}_{n|n}, \quad (3.17)$$

Si l'échantillon est disponible, i.e. $c_{n+1} = 1$ et donc $y_{n+1} = x_{n+1}$, l'estimée de l'état est filtrée, et la matrice de covariance de l'erreur d'estimation de l'état est mise à jour selon les équations suivantes :

$$\hat{\mathbf{x}}_{n+1|n+1} = \hat{\mathbf{x}}_{n+1|n} + K_{n+1}(y_{n+1} - c_{n+1}C\hat{\mathbf{x}}_{n+1|n}), \quad (3.18)$$

$$P_{n+1|n+1} = (I_d - c_{n+1}K_{n+1}C^\top)P_{n+1|n}, \quad (3.19)$$

où K_{n+1} est le gain du filtre de Kalman donné par :

$$K_{n+1} = \frac{c_{n+1}P_{n+1|n}C^\top}{c_{n+1}CP_{n+1|n}C^\top}. \quad (3.20)$$

Si l'échantillon est manquant, i.e. $c_{n+1} = 0$, le gain du filtre de Kalman est nul $K_{n+1} = 0$. Il s'en suit que l'estimée de l'état ainsi que la matrice de covariance de l'erreur d'estimation sont conservées *a posteriori* selon :

$$\hat{\mathbf{x}}_{n+1|n+1} = \hat{\mathbf{x}}_{n+1|n}, \quad (3.21)$$

$$P_{n+1|n+1} = P_{n+1|n}. \quad (3.22)$$

Dans les deux cas, l'estimée de la sortie du système x_{n+1}^s , qui correspond donc à l'échantillon à prédire, et l'erreur quadratique moyenne d'estimation sont donnés à tout instant $n + 1$ par :

$$\hat{x}_{n+1|n+1}^s = C\hat{\mathbf{x}}_{n+1|n+1} \quad (3.23)$$

$$J_{n+1|n} = \mathbf{Tr} \left[CP_{n+1|n}C^\top \right]. \quad (3.24)$$

où \mathbf{Tr} est la trace d'une matrice.

On montre dans ce qui suit que le signal reconstruit $\{z_n\}$ est donné à l'instant $n + 1$ par :

$$z_{n+1} = \hat{x}_{n+1|n+1}^s. \quad (3.25)$$

3.2.3 Analyse du filtre et conclusions

Grâce à la représentation d'état choisie, les équations du filtre de Kalman énoncées ci-dessus, telle que celle donnant le gain du filtre, peuvent être simplifiées. Ainsi, si un échantillon est disponible à l'instant $n + 1$, i.e., $c_{n+1} = 1$, le numérateur du gain du filtre, $c_{n+1}P_{n+1|n}C^\top$, est un vecteur égal à la première colonne de la matrice $P_{n+1|n}$. De plus, le dénominateur de celui-ci, $c_{n+1}CP_{n+1|n}C^\top$, est alors un scalaire égal au premier élément de la matrice $P_{n+1|n}$. Par conséquent, le gain du filtre de Kalman est obtenu en divisant les éléments de la première colonne de $P_{n+1|n}$ par le premier élément de celle-ci. Ceci nécessite L opérations au lieu de $2L(L + 1)$ dans le cas classique. De plus, on en déduit que le premier élément du vecteur K_{n+1} , qu'on note $K_{n+1}(1)$, est toujours égal à 1 lorsqu'un échantillon est disponible. Ainsi, en faisant appel à l'équation de mise à jour de l'état par le filtre de Kalman (3.18), le premier élément de l'état filtré est alors donné par :

$$\begin{aligned} \hat{x}_{n+1|n+1} &= \hat{x}_{n+1|n} + K_{n+1}(1) (y_{n+1} - c_{n+1}C\hat{\mathbf{x}}_{n+1|n}) \\ &= \hat{x}_{n+1|n} + (y_{n+1} - \hat{x}_{n+1|n}) \\ &= y_{n+1} = x_{n+1} \end{aligned}$$

Ainsi, l'estimée filtrée d'un échantillon disponible est égale à la valeur de l'échantillon. Par conséquent, le signal reconstruit, qui prend la valeur de l'échantillon lorsque celui-ci est disponible, est alors égal dans ce cas à la sortie du filtre, $z_{n+1} = \hat{x}_{n+1|n+1}$.

Par contre, si l'échantillon à l'instant $n + 1$ est manquant, le gain de Kalman est nul, il s'en suit que l'estimée de l'état ainsi que la matrice de covariance de l'erreur d'estimation sont inchangées *a posteriori* (équation (3.21)). Par conséquent, l'estimation de la sortie du système qui est donc celle de l'échantillon à prédire est donnée par :

$$\hat{x}_{n+1|n+1} = C\hat{\mathbf{x}}_{n+1|n} = CA\hat{\mathbf{x}}_{n|n} = \mathbf{a}^\top \hat{\mathbf{x}}_{n|n}. \quad (3.26)$$

La prédiction de l'échantillon manquant à l'instant $n + 1$ ainsi calculée s'écrit alors sous la forme suivante :

$$\hat{x}_{n+1|n+1} = \sum_{i=1}^L a_i \hat{x}_{n-i|n}. \quad (3.27)$$

Dans le vecteur $\hat{\mathbf{x}}_{n|n}$, les échantillons disponibles sont inchangés ($\hat{x}_{n-i|n} = x_{n-i}$), alors que les échantillons manquants sont remplacés par leurs prédictions actualisées. Il est à noter ici que si l'échantillon x_{n-i} est manquant, sa prédiction actualisée, $\hat{x}_{n-i|n}$, est différente de sa prédiction fournie par le filtre de Kalman à l'instant $n - i$, $\hat{x}_{n-i|n-i}$. Ceci est dû au filtrage de l'état par le filtre de Kalman lorsqu'un échantillon est disponible au cours des $L - 1$ pas de temps suivants. En effet, si un échantillon est disponible, l'état (constitué des L derniers échantillons ou leur prédiction) est filtré par le filtre Kalman selon l'équation (3.18). Par conséquent, des termes correctifs s'ajoutent à la prédiction des échantillons manquants existants dans cet état. Ainsi, les valeurs $\hat{x}_{n-i|n-i+t}$ pour $1 \leq t \leq L - 1$ (les valeurs actualisées de la prédiction de x_{n-i} au cours des $L - 1$ pas de temps suivants) s'expriment par :

$$\begin{aligned} \hat{x}_{n-i|n-i+t} &= \hat{x}_{n-i|n-i} + \sum_{j=1}^t K_{n-i+j}(j) (y_{n-i+j} - c_{n-i} C \hat{\mathbf{x}}_{n-i+j|n-i+j-1}) \\ \text{pour } 1 \leq t \leq L - 1. \end{aligned} \quad (3.28)$$

Ceci rappelle les termes additifs correctifs du prédicteur à passé infini incomplet qui apparaissent dans la représentation autorégressive de celui-ci dans l'exemple 2.2.1 du chapitre 2. Les termes additifs dus au filtrage de l'état par le filtre de Kalman s'ajoutent uniquement à la prédiction des échantillons manquants existants dans l'état, les échantillons disponibles restant toutefois inchangés. Ceci est dû au fait que, dans la matrice de covariance de l'erreur d'estimation de l'état, les termes correspondant à des échantillons disponibles sont nuls. Particulièrement, les éléments de la première colonne de celle-ci sont nuls lorsqu'ils sont aux mêmes positions que celles des échantillons disponibles dans l'état à filtrer. Il en est de même pour les éléments du gain du filtre de Kalman. Il s'en suit que les échantillons disponibles de l'état ne sont pas changés lorsque celui-ci est filtré. L'estimée filtrée de l'état, $\hat{\mathbf{x}}_{n+1|n+1}$, est donc constituée des L échantillons précédents du signal où les échantillons manquants sont remplacés par leurs prédictions optimales à l'instant $n + 1$ (actualisées à l'instant $n + 1$). Il est donc égal au vecteur régresseur du signal reconstruit à l'instant $n + 1$. On a donc :

$$\mathbf{z}_{n+1} = \hat{\mathbf{x}}_{n+1|n+1}. \quad (3.29)$$

Il est à noter que dans le cas de la reconstruction d'un signal en temps réel, la prédiction actualisée des échantillons manquants ne peut pas être utilisée pour la reconstruction du signal, la mise à jour de celle-ci ayant lieu aux instants suivants. Par contre, la prédiction actualisée contenue dans l'état servira à l'estimation de l'état et à la prédiction optimale des échantillons suivants. Ainsi, le signal reconstruit est donné à l'instant $n + 1$ par :

$$z_{n+1} = \hat{x}_{n+1|n+1} = \begin{cases} x_{n+1} & \text{si } x_{n+1} \text{ est disponible} \\ \hat{x}_{n+1|n} & \text{sinon.} \end{cases} \quad (3.30)$$

Le filtre de Kalman décrit ci-dessus permet donc une prédiction du processus AR(L) à échantillons manquants minimisant le critère donné par l'équation (3.13). Or, Bondon [13, 11] propose une expression analytique du prédicteur optimal au sens des moindres carrés d'un processus à passé infini incomplet ayant subi la perte d'un nombre fini d'échantillons. Ainsi, le filtre de Kalman décrit ci-dessus, permet un calcul numérique récursif de la prédiction à passé infini incomplet dans le cas particulier d'un processus AR d'ordre fini L . Par analogie entre les deux approches

de prédiction, on déduit des équations (3.23) et (1.57) que l'erreur quadratique moyenne de prédiction est égale à :

$$E \{(x - \hat{x})^2\} = \mathbf{Tr} [CP_{n+1|n}C^\top] = \sigma_\epsilon^2 \psi_0. \quad (3.31)$$

où ψ_0 est définie au paragraphe 2 du chapitre 1 comme étant le premier élément du vecteur qui vérifie l'équation matricielle (1.55).

On rappelle ici que l'erreur quadratique moyenne de prédiction dans le cas de signaux à échantillons manquants est supérieure à celle obtenue dans le cas de signaux sans échantillons manquants. Dans ce dernier cas, elle est égale à σ_ϵ^2 .

Dans ce qui suit, l'algorithme Kalman MCR pseudo-linéaire, utilisant conjointement avec l'algorithme MCR pseudo-linéaire un filtre de Kalman pour la prédiction, est décrit et analysé.

3.3 Algorithme Kalman MCR pseudo-linéaire

3.3.1 Description de l'algorithme

Le filtre de Kalman décrit ci-dessus permet une reconstruction optimale d'un processus AR(L) à échantillons manquants. On propose alors de l'utiliser conjointement avec l'algorithme MCR pseudo-linéaire décrit au paragraphe 3.1.2 pour une identification non biaisée et une reconstruction optimale en ligne du processus AR(L) à échantillons manquants.

Afin de pouvoir estimer l'état, le filtre de Kalman utilise à chaque instant les estimées des paramètres obtenus à l'instant précédent grâce à l'algorithme MCR pseudo-linéaire. Ainsi, à l'instant $n + 1$, les éléments de la première ligne de la matrice A sont remplacés par les éléments du vecteur $\hat{\mathbf{a}}_n^\top$, l'estimée du vecteur des paramètres obtenue à l'instant n par l'algorithme MCR pseudo-linéaire. La matrice A est alors notée A_{n+1} . Ayant l'estimation de l'état, $\hat{\mathbf{x}}_{n+1|n}$, on en déduit la prédiction de l'échantillon x_{n+1} qui est alors donnée par $\hat{x}_{n+1|n} = C\hat{\mathbf{x}}_{n+1|n}$. Si l'échantillon est disponible, les paramètres sont mis à jour grâce à l'algorithme MCR pseudo-linéaire. Pour cela, ce dernier utilise alors comme vecteur régresseur l'état filtré à l'instant précédent, $\hat{\mathbf{x}}_{n|n}$, et la prédiction de l'échantillon fournie par le filtre de Kalman. Par contre, lorsque l'échantillon est manquant, le signal est reconstruit par sa prédiction $\hat{x}_{n+1|n}$, mais les paramètres ne sont pas mis à jour.

Ainsi, l'algorithme Kalman MCR pseudo-linéaire se résume aux étapes suivantes :

Initialisation à l'instant L :

- Initialisation de l'algorithme des MCR pseudo-linéaire :

$$\begin{aligned} \hat{\mathbf{a}}_L &= [0, \dots, 0]^\top; \\ G_L &= \alpha I_d; \\ \hat{\Psi}_{L+1} &= \mathbf{x}_L = [x_L, \dots, x_1]^\top \end{aligned} \quad (3.32)$$

- Initialisation du filtre de Kalman :

$$\begin{aligned} P_{L|L} &= \alpha I_d; \\ \hat{\mathbf{x}}_{L|L} &= \mathbf{x}_L = [x_L, \dots, x_1]^\top; \\ R_\epsilon &= [1, 0, \dots, 0]^\top [1, 0, \dots, 0]; \end{aligned} \quad (3.33)$$

A l'instant $n + 1$ pour $L + 1 \leq n + 1 \leq N + 1$, l'algorithme se résume aux étapes suivantes :

$$A_{n+1} = \left[\begin{array}{ccc|c} \hat{a}_{n,1} & \cdots & \cdots & \hat{a}_{n,L} \\ \hline 1 & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \end{array} \right], \quad (3.34)$$

$$P_{n+1|n} = A_{n+1}P_{n|n}A_{n+1}^\top + R_\epsilon,$$

$$\hat{\mathbf{x}}_{n+1|n} = A_{n+1}\hat{\mathbf{x}}_{n|n},$$

$$\hat{\Psi}_{n+1} = \hat{\mathbf{x}}_{n|n},$$

$$\hat{x}_{n+1|n} = C\hat{\mathbf{x}}_{n+1|n}.$$

Si x_{n+1} est disponible, i.e. $c_{n+1} = 1$. Dans ce cas, $y_{n+1} = x_{n+1}$.

- Les paramètres ainsi que la matrice G_{n+1} sont mis à jour grâce à l'algorithme MCR pseudo-linéaires :

$$\gamma_{n+1} = \frac{G_n \hat{\Psi}_{n+1}}{\lambda + \hat{\Psi}_{n+1} G_n \hat{\Psi}_{n+1}}, \quad (3.35a)$$

$$G_{n+1} = \frac{1}{\lambda} (I_d - \gamma_{n+1} \hat{\Psi}_{n+1}^\top) G_n, \quad (3.35b)$$

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n + \gamma_n (x_{n+1} - \hat{x}_{n+1|n}), \quad (3.35c)$$

$$(3.35d)$$

- L'état ainsi que la matrice de covariance de l'erreur d'estimation sont mises à jour grâce au filtre de Kalman.

$$K_{n+1} = \frac{\text{1ère colonne de } P_{n+1|n}}{\text{1er élément de } P_{n+1|n}}, \quad (3.36a)$$

$$P_{n+1|n+1} = (I_d - K_{n+1} \mathbf{c}_{n+1}^\top) P_{n+1|n}, \quad (3.36b)$$

$$\hat{\mathbf{x}}_{n+1|n+1} = \hat{\mathbf{x}}_{n+1|n} + K_{n+1} (x_{n+1} - \hat{x}_{n+1|n}), \quad (3.36c)$$

Contrairement, si x_{n+1} est absent, $c_{n+1} = 0$,

- l'estimation de l'état *a priori*, $\hat{\mathbf{x}}_{n+1|n}$, n'est pas filtrée par le filtre de Kalman, la matrice de covariance de l'erreur d'estimation est donc conservée *a posteriori*.

$$K_{n+1} = 0, \quad (3.37a)$$

$$P_{n+1|n+1} = P_{n+1|n}, \quad (3.37b)$$

$$\hat{\mathbf{x}}_{n+1|n+1} = \hat{\mathbf{x}}_{n+1|n}, \quad (3.37c)$$

$$(3.37d)$$

- les paramètres AR ne sont pas mis à jour par l'algorithme MCR pseudo-linéaires. Par contre, le facteur d'oubli joue quand même sur la matrice G_{n+1} , puisque celui-ci dépend de l'éloignement temporel et non du nombre d'échantillons disponibles. On a alors :

$$G_{n+1} = \frac{1}{\lambda} G_n, \quad (3.38a)$$

$$\gamma_{n+1} = 0, \quad (3.38b)$$

$$\hat{\mathbf{a}}_{n+1} = \hat{\mathbf{a}}_n. \quad (3.38c)$$

3.3.2 Analyse de la convergence de l'algorithme

On se propose dans cette partie d'analyser la convergence de l'algorithme en utilisant comme critère la convergence en moyenne des paramètres estimés :

$$\lim_{n \rightarrow \infty} E \{ \hat{\mathbf{a}}_n \} \quad (3.39)$$

Pour cela on considère l'identification d'un processus AR(L) stationnaire ayant subi une perte aléatoire d'échantillons. Dans ce cas, la fonction de coût minimisée par l'algorithme, donnée par l'équation (3.4), admet un seul minimum global. Les paramètres correspondant à ce minimum sont donnés à l'instant $n+1$ par l'équation (3.8), ils sont calculés récursivement par l'algorithme. Il s'agit donc de vérifier si ce minimum correspond aux vrais paramètres du processus identifié. Donc, en d'autres termes, il s'agit de vérifier si en régime permanent les paramètres estimés sont biaisés en moyenne ou non. Or, les paramètres qui minimisent les moindres carrés pondérés par un facteur d'oubli exponentiel sont différents en moyenne de ceux qui minimisent les moindres carrés sans pondération. Ainsi, dans le cas classique sans échantillons manquants, l'algorithme MCR avec oubli exponentiel converge vers une estimation biaisée en moyenne des paramètres [4, 41, 42]. Arvaston *et al.* [4] calculent une expression asymptotique de ce biais dans le cas de l'identification de processus AR et ARX. Ce problème est aussi rencontré dans le cas de l'identification de signaux à échantillons manquants. Pour cela, nous analysons dans ce qui suit la convergence en moyenne de l'algorithme en distinguant les deux cas sans et avec pondération par un oubli exponentiel.

Cas ($\lambda = 1$) : Dans ce cas, en utilisant l'hypothèse de stationnarité du signal identifié, on a asymptotiquement :

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{1}{n-L} \hat{R}_{z,n} &= \lim_{n \rightarrow \infty} \frac{1}{n-L} \sum_{k=L+1}^n c_k \mathbf{z}_k \mathbf{z}_k^\top = R_z \\ \lim_{n \rightarrow \infty} \frac{1}{n-L} \hat{\mathbf{r}}_{z,n} &= \lim_{n \rightarrow \infty} \frac{1}{n-L} \sum_{k=L+1}^n c_k z_k \mathbf{z}_k = \mathbf{r}_z, \end{aligned} \quad (3.40)$$

$$\text{où } R_z = \begin{bmatrix} r_z(0) & r_z(1) & \dots & r_z(L-1) \\ r_z(1) & r_z(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_z(1) \\ r_z(L-1) & \dots & r_z(1) & r_z(0) \end{bmatrix} \quad \text{et } \mathbf{r}_z = \begin{bmatrix} r_z(1) \\ \vdots \\ \vdots \\ r_z(L) \end{bmatrix}$$

$r_z(i)$ étant le coefficient d'autocorrélation d'ordre i du processus reconstruit $\{z_n\}$.

La moyenne des paramètres estimés à la convergence de l'algorithme est donc donnée par :

$$\lim_{n \rightarrow \infty} E \{ \hat{\mathbf{a}}_n \} = R_z^{-1} \mathbf{r}_z \quad (3.41)$$

C'est la solution au sens des moindres carrés appliqués au signal reconstruit (où les échantillons manquants sont remplacés par leurs prédictions supposées correctes). Wallin *et al.* [69] ont calculé une expression du biais existant sur l'estimation des paramètres utilisant cette approche d'identification et en déduisent une condition pour une identification sans biais donnée par l'équation (1.50). Selon cette condition, si l'approche de prédiction utilisée est optimale au sens des moindres carrés, les paramètres estimés sont non biaisés. La prédiction en utilisant le filtre

de Kalman décrit ci-dessus permet de vérifier cette condition. On en déduit que lorsque le processus est reconstruit par une approche de prédiction optimale telle que celle utilisant le filtre de Kalman, on a asymptotiquement : $R_z^{-1} \mathbf{r}_z = \mathbf{a}$. Ainsi, l'algorithme Kalman MCR pseudo-linéaire sans pondération par un oubli exponentiel converge en moyenne vers les bons paramètres.

Cas ($\lambda < 1$) : Lorsqu'une pondération par un oubli exponentiel est considérée dans l'algorithme, les expressions asymptotiques des moyennes de la matrice $\hat{R}_{z,n}$ et du vecteur $\hat{\mathbf{r}}_{z,n}$ sont données par :

$$\begin{aligned} E \left\{ \hat{R}_{z,n} \right\} &= \sum_{k=L+1}^n \lambda^{n-k} E \left\{ c_k \mathbf{z}_k \mathbf{z}_k^\top \right\} \\ &= \sum_{k=L+1}^n \lambda^{n-k} (1-q) R_z = \frac{(1-\lambda^{n-L})(1-q)}{1-\lambda} R_z \\ E \left\{ \hat{\mathbf{r}}_{z,n} \right\} &= \frac{(1-\lambda^{n-L})(1-q)}{1-\lambda} \mathbf{r}_z \end{aligned} \quad (3.42)$$

En se référant à l'équation (3.42), on constate que :

$$E \left\{ \hat{R}_{z,n+1} \right\}^{-1} E \left\{ \hat{\mathbf{r}}_{z,n+1} \right\} = R_z^{-1} \mathbf{r}_z = \mathbf{a}. \quad (3.43)$$

Or l'estimée en moyenne des paramètres grâce à l'algorithme Kalman MCR pseudo-linéaire est donnée asymptotiquement par :

$$\lim_{n \rightarrow \infty} E \left\{ \hat{\mathbf{a}}_n \right\} = \lim_{n \rightarrow \infty} E \left\{ \hat{R}_{z,n+1}^{-1} \hat{\mathbf{r}}_{z,n+1} \right\} \quad (3.44)$$

On en déduit que lorsqu'une pondération par un facteur d'oubli exponentiel est utilisée, l'algorithme Kalman MCR pseudo-linéaire converge en moyenne vers une estimation biaisée des paramètres. Le biais des paramètres estimés est égal à :

$$\mathbf{a} - \lim_{n \rightarrow \infty} E \left\{ \hat{\mathbf{a}}_n \right\} = E \left\{ \hat{R}_{z,n+1}^{-1} \hat{\mathbf{r}}_{z,n+1} \right\} - E \left\{ \hat{R}_{z,n+1} \right\}^{-1} E \left\{ \hat{\mathbf{r}}_{z,n+1} \right\} \quad (3.45)$$

Arvaston *et al.* [4, 41, 42] établissent des expressions asymptotiques du biais ainsi que de la covariance des paramètres d'un modèle AR estimés grâce à l'algorithme MCR avec oubli exponentiel lorsque le facteur d'oubli tend vers 1 ($\lambda \rightarrow 1$). Ainsi, en procédant de la même façon, l'expression du biais asymptotique des paramètres estimés en utilisant l'algorithme Kalman MCR pseudo-linéaire est donnée par :

$$\begin{aligned} E \left\{ \hat{\mathbf{a}} \right\} - \mathbf{a} &= \frac{1-\lambda}{1+\lambda} \sum_{\substack{1 \leq \gamma < L \\ 0 \leq m \leq L \\ 1 \leq j, k, l \leq L}} \lambda^\gamma a_m R_z(i, j)^{-1} R_z(k, l)^{-1} \\ &\quad \times [r_z(l + \gamma - k) r_z(\gamma + m - j) + r_z(l + \gamma - j) r_z(\gamma + m - k)] \\ &\quad + \frac{1-\lambda}{1+\lambda} \sum_{\substack{0 \leq m \leq L \\ 1 \leq j, k, l \leq L \\ 1 \leq p, q \leq L}} a_m \frac{(\rho_q \rho_p \lambda)^L}{1 - \lambda \rho_p \rho_q} R_z(i, j)^{-1} R_z(k, l)^{-1} \alpha_p \alpha_q \left[\rho_p^{l-k} \rho_q^{m-j} + \rho_p^{l-j} \rho_q^{m-k} \right] \\ &\quad + \mathcal{O}((1-\lambda)^2). \end{aligned} \quad (3.46)$$

où $R_z(i, j)$ est l'élément d'ordre (i, j) de la matrice d'autocorrélation du signal reconstruit R_z , $r_z(l + \gamma - j)$ est le coefficient d'autocorrélation d'ordre $(l + \gamma - j)$ du signal reconstruit z_n . Les paramètres ρ_p correspondent aux racines de l'équation caractéristique du processus AR et les α_p sont les coefficients appropriés qui permettent d'écrire la fonction d'autocorrélation $r_z(i)$ du processus z_n sous la forme suivante :

$$r_z(i) = \sum_{p=1}^L \alpha_p \rho_p^i. \quad (3.47)$$

On déduit de l'expression du biais donnée par (3.46) qu'il est nul lorsqu'on ne considère pas un oubli exponentiel ($\lambda = 1$) et que celui-ci est croissant lorsqu'on diminue λ (avec $\lambda < 1$). Le biais est de l'ordre de $(1 - \lambda)$, il dépend d'une façon non linéaire des paramètres et de la dynamique du processus identifié.

D'autre part, l'expression de la covariance asymptotique de l'estimation des paramètres est donnée par :

$$\mathbf{Cov}_{\hat{a}}(\tau) = \mathbf{cov}(\hat{a}_n, \hat{a}_{n+\tau}) = \frac{1 - \lambda}{1 + \lambda} \lambda^{|\tau|} R_z^{-1}(0) \sigma_e^2 + \mathcal{O}((1 - \lambda)^2). \quad (3.48)$$

où σ_e^2 est l'erreur quadratique moyenne de prédiction, elle est donnée par :

$$\sigma_e^2 = E \{(x_n - \hat{x}_n)^2\} = E \left\{ \left(x_n - \sum_{i=1}^L a_i z_{n-i} \right)^2 \right\} \quad (3.49)$$

L'équation (3.48) montre que la covariance dépend principalement de l'inverse du RSB, i.e. du rapport signal sur bruit. Or dans le cas de signaux à échantillons manquants l'erreur quadratique moyenne de prédiction est supérieure à celle obtenue dans le cas classique sans perte d'échantillons. On en déduit que la covariance des paramètres estimés (ou la variance lorsque $\tau = 0$) croît avec la probabilité de perte des échantillons. De plus, on constate que la variance des paramètres estimés est de l'ordre de $(1 - \lambda)$. Ceci justifie l'augmentation de la variance des paramètres estimés pour des valeurs du facteur d'oubli plus faibles.

3.4 Simulations

Dans cette partie, l'algorithme Kalman MCR pseudo-linéaire est analysé par des simulations et comparé à l'algorithme du gradient stochastique utilisant le prédicteur à passé infini incomplet décrit dans le chapitre 2.

On s'intéresse tout d'abord aux performances en régime permanent de l'algorithme Kalman MCR pseudo-linéaire. Ainsi, on étudie dans les premières simulations le biais et la variance des paramètres estimés en fonction du facteur d'oubli λ et de la probabilité de perte q . On s'intéresse par la suite à la vitesse de convergence ainsi qu'au rapport signal sur bruit de reconstruction en fonction de λ et de la probabilité de perte q . Finalement, cet algorithme est comparé à l'algorithme du gradient stochastique dans le cas de signaux AR non stationnaires synthétiques et dans le cas de signaux réels de parole.

Simulation 1 On s'intéresse ici au biais et à la variance des paramètres estimés en utilisant l'algorithme Kalman MCR pseudo-linéaire en fonction du facteur d'oubli λ et de la probabilité de perte q . Pour cela le signal test utilisé est un processus AR stationnaire d'ordre 2 dont les paramètres sont $[1.5, -0.7]$ généré sur 10^5 points. La perte d'échantillons est considérée aléatoire suivant une loi de Bernoulli indépendante du signal. Le biais et la variance des paramètres estimés sont calculés après la convergence de l'algorithme grâce aux équations (2.34) et (2.35). La figure 3.2 montre l'évolution du biais de l'estimée moyenne du paramètre a_1 en fonction de la probabilité de perte q pour différentes valeurs du facteur d'oubli λ . La figure 3.3 montre l'évolution du biais en fonction de λ pour différentes valeurs de q .

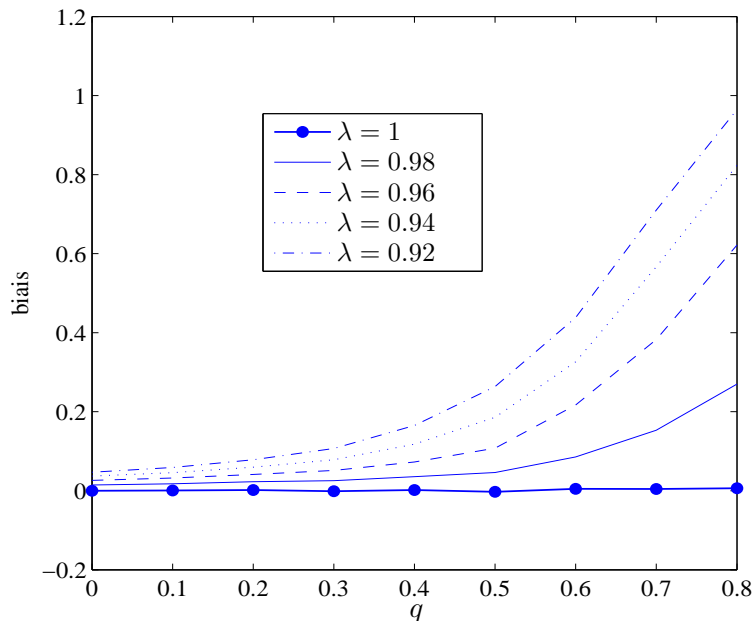


FIG. 3.2 – Biais de l'estimation de a_1 en fonction de la probabilité de perte q en utilisant l'algorithme Kalman MCR pseudo-linéaire pour différentes valeurs de λ .

En se référant à la figure 3.2, on vérifie que l'estimation du paramètre a_1 en utilisant l'algorithme Kalman MCR pseudo-linéaire est effectivement biaisée en moyenne pour $\lambda < 1$ (lorsqu'une pondération par un facteur d'oubli exponentiel est considérée). Ce biais croît de manière exponentielle avec la probabilité de perte q pour une valeur donnée de λ . La valeur de ce biais croît lorsque λ diminue. D'autre part, lorsque $\lambda = 1$ (sans oubli exponentiel), l'estimation du paramètre a_1 n'est pas biaisée conformément à l'analyse faite au paragraphe 3.3.2. Ainsi, grâce à l'utilisation du filtre de Kalman pour la prédiction, la valeur du biais d'identification est presque nulle pour toutes les valeurs de la probabilité de perte q lorsque $\lambda = 1$. Ces observations sont confirmées à la figure 3.3, où l'on voit bien que le biais décroît lorsque λ tend vers 1 et que celui-ci est nul lorsque $\lambda = 1$ pour toutes les valeurs de q . On constate de plus que la décroissance du biais en fonction de λ est linéaire vérifiant l'équation suivante :

$$Biais = (1 - \lambda)C(q, \mathbf{a}). \quad (3.50)$$

où $C(q, \mathbf{a})$ est une constante fonction de q et des paramètres \mathbf{a} . Ceci confirme que le biais est de l'ordre de $(1 - \lambda)$ tel qu'il a été dit au paragraphe 3.3.2.

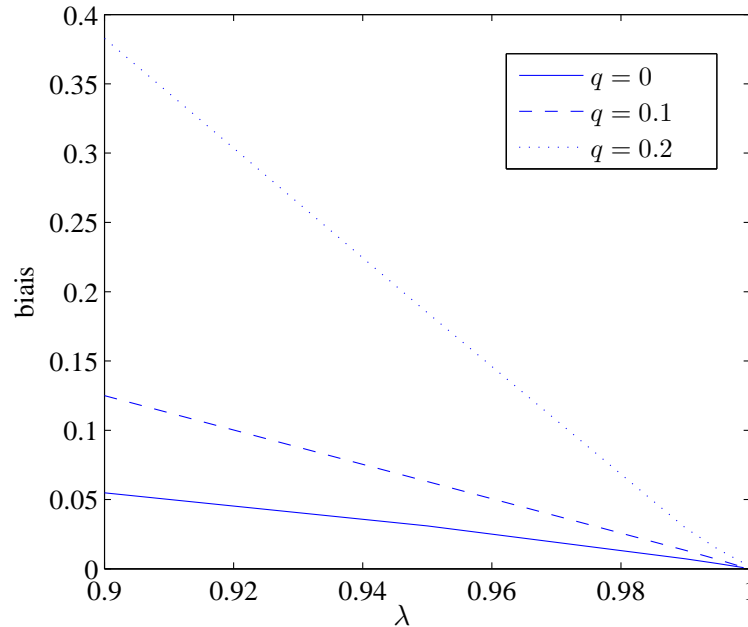


FIG. 3.3 – Biais de l'estimation de a_1 en utilisant l'algorithme Kalman MCR pseudo-linéaire en fonction du facteur d'oubli λ pour différentes probabilités de perte q .

La figure 3.4 montre, pour le même signal test, la variance de l'estimée de a_1 par l'algorithme Kalman MCR en fonction du facteur d'oubli λ pour différentes valeurs de la probabilité de perte q . Elle montre la décroissance de la variance du paramètre estimé lorsque λ croît vers 1. La variance est plus élevée lorsque la probabilité de perte est élevée. Ceci était prévu puisque la variance, donnée par l'équation (3.48) pour $\tau = 0$, est proportionnelle à l'erreur quadratique moyenne de prédiction. Cette dernière croît avec la probabilité de perte des échantillons.

Simulation 2 Le signal test utilisé ici est la juxtaposition de deux processus AR généré sur 5.10^4 échantillons. Pour la première partie du signal, la valeur des paramètres est $[1.5; -0.7]$, alors que pour les échantillons restants, les paramètres sont $[1; -0.5]$. On s'intéresse tout d'abord à l'effet de λ sur la vitesse de convergence et d'adaptation de l'algorithme aux changements de paramètres. La perte d'échantillons est considérée aléatoire suivant une loi de Bernoulli indépendante du signal. La figure 3.5 montre l'estimation instantanée moyenne du paramètre a_1 en utilisant l'algorithme Kalman MCR pseudo-linéaire pour différentes valeurs du facteur d'oubli λ , la probabilité de perte d'échantillons étant $q = 0.2$. L'estimation instantanée moyenne est déduite grâce à un Monte Carlo de 1000 réalisations du signal et du schémas de perte. Ainsi pour chaque réalisation, les paramètres sont estimés pour différentes valeurs de λ . Les différentes réalisations sont alors moyennées pour chaque valeur de λ .

La figure 3.5 montre que la vitesse de convergence et d'adaptation de l'algorithme croissent, comme il se doit, lorsque l'on diminue la valeur de λ . Ceci se fait aux dépens d'une estimation biaisée en moyenne ainsi que d'une variance élevée des paramètres. On peut constater de plus sur cette figure que même pour des valeurs élevées de λ , l'algorithme présente une bonne vitesse de convergence. Ainsi, par exemple, pour $\lambda = 0.99$, la vitesse de convergence est déjà voisine de celle obtenue pour $\lambda = 0.95$ alors que le biais et la variance des paramètres estimés sont faibles en comparaison de ceux obtenus pour $\lambda = 0.95$. Ainsi, $\lambda = 0.99$ semble être une valeur optimale

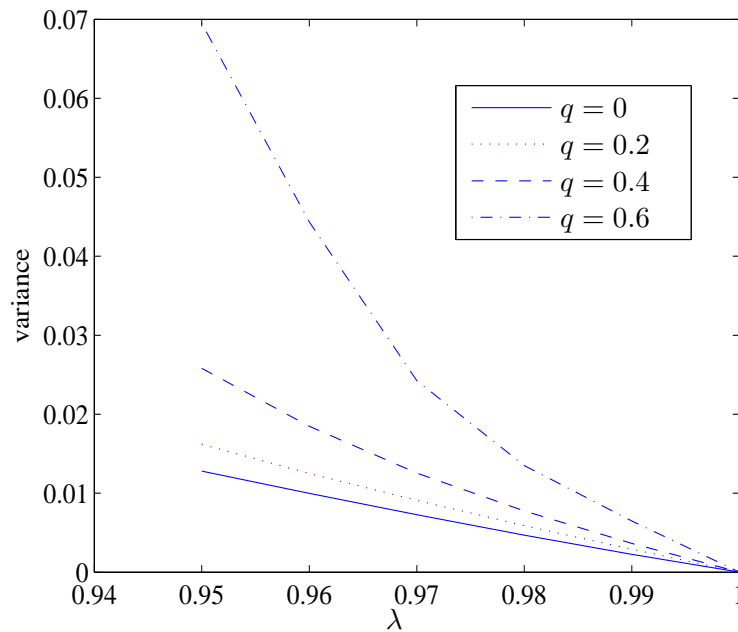


FIG. 3.4 – Variance de l'estimée de a_1 par l'algorithme Kalman MCR pseudo-linéaire en fonction du facteur d'oubli λ pour différentes probabilités de perte q .

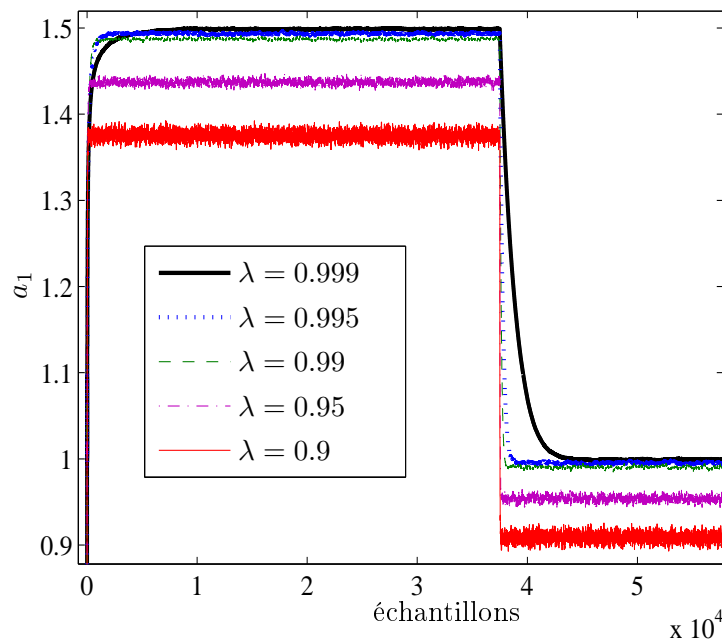


FIG. 3.5 – Estimation instantanée moyenne du paramètre $a_1 = 1.5$ en utilisant l'algorithme Kalman MCR pseudo-linéaire pour différentes valeurs du facteur d'oubli pour une probabilité de perte $q = 0.2$

du facteur d'oubli offrant une bonne vitesse de convergence pour des valeurs faibles du biais et de la variance des paramètres estimés.

La figure 3.6 montre l'estimée instantanée moyenne du paramètre a_1 en utilisant l'algorithme Kalman MCR pseudo-linéaire pour différentes valeurs de la probabilité de perte d'échantillons, le facteur d'oubli étant fixé à $\lambda = 0.99$. La valeur moyenne de l'estimée des paramètres se fait aussi grâce à un Monte Carlo de 1000 réalisations du signal et du schéma de perte et ceci pour différentes valeurs de la probabilité de perte.

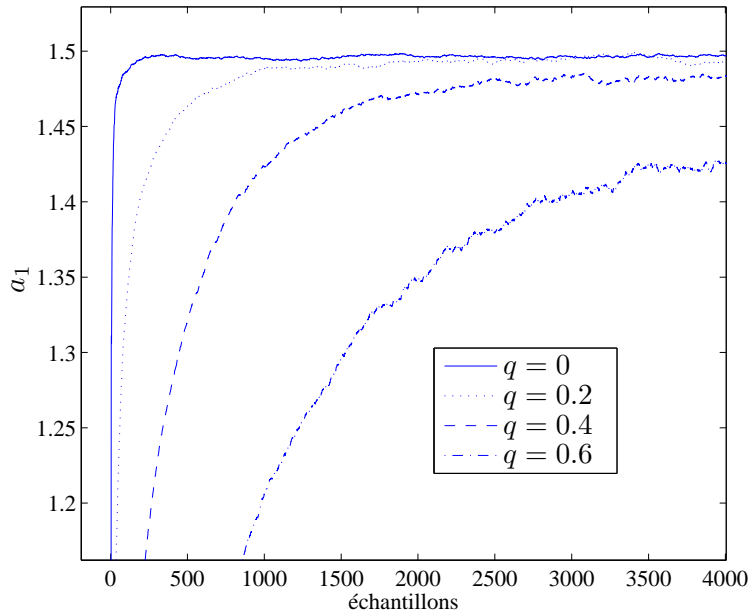


FIG. 3.6 – Estimation instantanée moyenne du paramètre $a_1 = 1.5$ pour différentes probabilités de perte en utilisant l'algorithme Kalman MCR pseudo-linéaire avec un oubli constant $\lambda = 0.99$

La figure 3.6 montre que la vitesse de convergence de l'algorithme décroît lorsque la probabilité de perte est plus élevée. Ceci est dû au fait que les paramètres ne sont mis à jour que lorsqu'un échantillon est disponible. Ainsi, lorsque la probabilité de perte d'échantillons est plus élevée, les paramètres sont mis à jour moins fréquemment ce qui ralentit la vitesse de convergence en moyenne. Ceci est accompagné comme il se doit d'une augmentation du biais des paramètres estimés. Ainsi, l'expression du biais des paramètres estimés est donnée par l'équation (3.46). Celui-ci est proportionnel à l'erreur quadratique moyenne de prédiction qui croît avec la probabilité de perte q .

simulation 3 On étudie ici les performances de l'algorithme Kalman MCR pseudo-linéaire dans la reconstruction d'un signal réel de parole en fonction du facteur d'oubli λ et de la probabilité de perte q . Ainsi, on considère la reconstruction du signal test usuel '*Mary had a little lamb, its fleece was white as snow*' échantillonné à la fréquence de 8000 kHz. Le signal de parole est caractérisé par un processus AR d'ordre 12. La perte d'échantillons est aléatoire suivant une loi de Bernoulli. Le rapport signal sur bruit de reconstruction (RSB) est utilisé pour mesurer les performances de l'algorithme dans la reconstruction du signal. Le RSB se mesure en dB, il est

donné par l'équation suivante :

$$RSB(dB) = 10 \log_{10} \left\{ \frac{\sum_{i=1}^N x_i^2}{\sum_{i=1}^N (x_i - z_i)^2} \right\}. \quad (3.51)$$

La figure 3.7 montre le RSB obtenu pour la reconstruction du signal de parole grâce à l'algorithme Kalman MCR pseudo-linéaire en fonction du facteur d'oubli λ pour différentes valeurs de la probabilité de perte q .

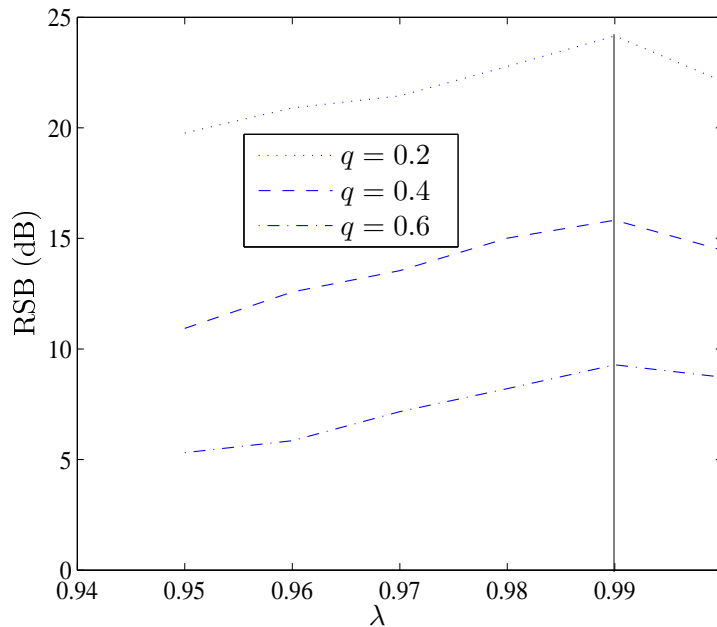


FIG. 3.7 – RSB fourni par l'algorithme Kalman MCR pseudo-linéaire, fonction de la probabilité de perte q

En se référant à la figure 3.7, on constate que, pour chaque valeur de q , le RSB croît avec le facteur d'oubli λ jusqu'à la valeur $\lambda = 0.99$ pour laquelle il admet un maximum. A partir de cette valeur de λ , le RSB décroît. Ceci est conforme à la constatation faite à la simulation 2 selon laquelle la valeur du facteur d'oubli $\lambda = 0.99$ semble être optimale. Ainsi, pour cette valeur de λ , l'algorithme s'adapte rapidement aux changements des paramètres caractérisant le signal de parole, tout en présentant un faible biais et une faible variance des paramètres estimés. Ceci permet d'obtenir un bon RSB. Par contre, pour des valeurs plus faibles de λ , la vitesse de convergence et d'adaptation de l'algorithme est élevée or ceci est aux dépens d'un biais et d'une variance élevés des paramètres estimés. Ceci entraîne une forte erreur de prédiction et par conséquent un faible RSB. Il est à noter ici que l'algorithme a été appliqué à la reconstruction d'autres signaux de paroles; pour chacun de ces signaux, le meilleur RSB est aussi obtenu pour des valeurs de λ voisines de $\lambda = 0.99$. D'autre part, le RSB décroît, comme il se doit, lorsque la probabilité de perte q augmente.

Simulation 4 L'algorithme Kalman MCR pseudo-linéaire est comparé ici aux algorithmes MCR pseudo-linéaire et celui du gradient stochastique qui utilise la prédiction à passé incomplet. Le signal test utilisé est un processus AR(2) non stationnaire, le même que celui utilisé dans la

simulation 2. La probabilité de perte des échantillons est de $q = 0.3$. Pour les algorithmes de type MCR, le facteur d'oubli est de $\lambda = 0.99$. Pour l'algorithme du gradient stochastique, le pas d'adaptation est de $\mu = 25.10^{-5}$, c'est une valeur empirique qui offre une bonne vitesse de convergence sans toutefois risquer la divergence de l'algorithme (Voir le paragraphe Simulation 2 du chapitre 2). Les performances des trois algorithmes comparés sont résumés dans le tableau 3.1 où b_i et σ_i désignent respectivement le biais et l'écart type de l'estimation du paramètre a_i . Ils sont calculés après la convergence des algorithmes par les équations respectives (2.34) et (2.35). Le CPU est le temps de calcul en secondes nécessaire à la simulation de chacun des trois algorithmes pour le signal test décrit ci-dessus implémentés sous MATLAB sur un processeur de $3GHz$.

TAB. 3.1 – Comparaison des algorithmes Kalman MCR pseudo-linéaire, MCR pseudo-linéaire et l'algorithme du gradient stochastique qui utilise la prédiction à passé incomplet.

Algorithme	b_1	b_2	σ_1	σ_2	RSB	CPU
Kalman MCR pseudo-linéaire	0.0006	0.003	0.014	0.015	12	1.5
MCR pseudo-linéaire	0.1811	0.151	0.018	0.017	11.3	1.3
gradient stochastique	0.0099	0.005	0.035	0.035	11.9	18.8

La figure 3.8 montre l'estimation instantanée moyenne du paramètre a_1 estimé grâce aux trois algorithmes comparés pour un Monté Carlo de 1000 réalisations du signal et du schémas de perte.

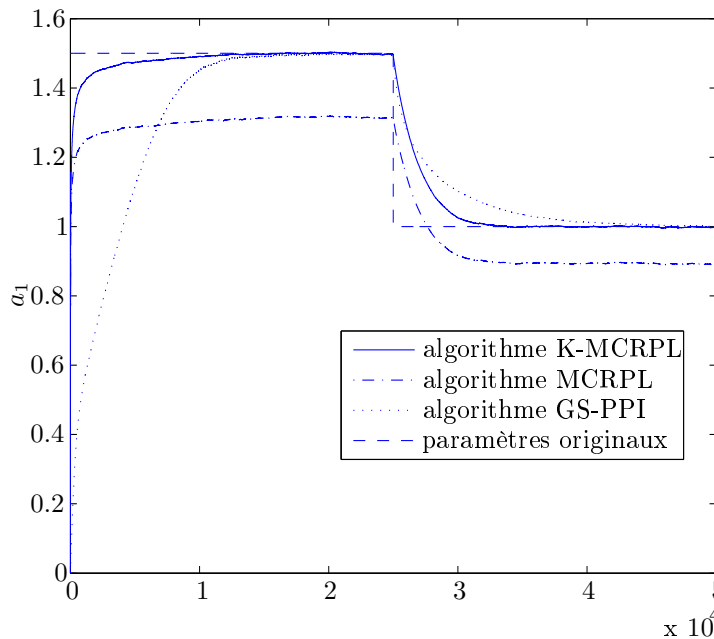


FIG. 3.8 – Estimation instantanée moyenne du paramètre a_1 en utilisant les algorithmes MCR pseudo-linéaire, Kalman MCR pseudo-linéaire et l'algorithme du gradient stochastique qui utilise la prédiction à passé incomplet.

La figure 3.8 et le tableau 3.1 montrent que l'algorithme Kalman MCR pseudo-linéaire et l'algorithme du gradient stochastique convergent en moyenne vers une estimation non biaisée des paramètres ce qui n'est pas le cas de l'algorithme MCR pseudo-linéaire. Il est à noter ici que

les paramètres estimés en utilisant l'algorithme Kalman MCR pseudo-linéaire sont légèrement biaisés pour $\lambda = 0.99$ mais ce biais est négligeable devant celui obtenu en utilisant l'algorithme MCR pseudo-linéaire. De plus, en se référant à la figure 3.8, on observe que les algorithmes de type MCR s'adaptent rapidement aux changements de paramètres comparés à l'algorithme du gradient stochastique. Il est vrai que les algorithmes fondés sur le gradient convergent plus rapidement pour des pas d'adaptation μ plus élevés. Or ceci se fait aux dépens d'une large variance des paramètres estimés qui peut mener à la divergence de l'algorithme.

Le tableau 3.1 montre que le meilleur RSB est obtenu en utilisant l'algorithme Kalman MCR pseudo-linéaire alors que celui obtenu grâce à l'algorithme MCR pseudo-linéaire est le plus bas. On constate aussi que l'algorithme du gradient stochastique qui utilise le prédicteur à passé infini incomplet est d'une complexité calculatoire plus grande que celles des deux autres algorithmes. Ceci est dû à l'inversion de matrices à chaque pas de temps tel qu'on l'a expliqué au chapitre 2. L'algorithme Kalman MCR pseudo-linéaire est de complexité calculatoire légèrement plus grande que celle de l'algorithme MCR pseudo-linéaire. De plus, ces deux algorithmes présentent la même faible variance des paramètres estimés en comparaison à celle obtenue grâce à l'algorithme fondé sur le gradient.

Ces trois algorithmes ont été appliqués à la reconstruction de signaux audio (de parole et de musique), l'algorithme Kalman MCR pseudo-linéaire a montré à chaque fois le meilleur RSB.

3.5 Conclusions

L'extension de l'algorithme des moindres carré récursifs (MCR) à l'identification de signaux à échantillons manquants est étudiée dans ce chapitre. Ainsi, l'algorithme MCR pseudo-linéaire [1, 62, 63] est tout d'abord présenté et analysé. Cet algorithme converge typiquement vers une estimation biaisée des paramètres lorsque l'approche de prédiction utilisée est la régression linéaire modifiée. Nous avons alors utilisé conjointement avec cet algorithme un filtre de Kalman pour une prédiction récursive du processus $AR(L)$ optimale au sens des moindres carrés. L'utilisation de cette approche de prédiction permet d'obtenir une identification non biaisée des paramètres lorsqu'on ne considère pas un oubli exponentiel. Lorsqu'un oubli exponentiel est utilisé ($\lambda < 1$), une expression du biais ainsi que de la variance des paramètres sont établies. Ils sont de l'ordre de $(1 - \lambda)$. Les simulations montrent que grâce à l'utilisation du filtre de Kalman pour la prédiction, ce biais est réduit par rapport à celui obtenu grâce à l'algorithme MCR pseudo-linéaire. D'autre part, grâce à l'utilisation d'un algorithme de type MCR pour l'identification, cet algorithme offre une meilleure vitesse de convergence et un meilleur suivi des paramètres que les algorithmes de type gradient décrits au chapitre 2. Les simulations montrent qu'un bon choix du facteur d'oubli ($\lambda \rightarrow 1$) permet d'obtenir un compromis entre une bonne vitesse de convergence et un biais des paramètres estimés négligeable. De surcroît, grâce à la simplification du filtre de Kalman, cet algorithme, bien que plus complexe que les algorithmes de type gradient utilisant la prédiction par régression linéaire modifiée ou à k -pas, est simple et rapide. Cet algorithme est appliqué à la reconstruction de signaux réels de parole, il montre de bonnes performances en terme de RSB.

Les algorithmes étudiés dans les chapitres 2 et 3 considèrent une représentation directe du filtre. Ils peuvent donc conduire à l'identification de modèles instables à moins que les pôles du filtre identifié ne soient contraints à chaque pas de temps d'être à l'intérieur du cercle unité. Or ceci est d'une complexité calculatoire élevée. Pour cela, on s'intéresse dans le chapitre suivant à l'identification du filtre en utilisant la structure en treillis de celui-ci. Ainsi, les coefficients de réflexion du filtre en treillis, identifiés par certaines méthodes simples [19, 49, 50], garantissent la stabilité du filtre correspondant.

Chapitre 4

Algorithmes en treillis

4.1 Introduction

Dans les chapitres 2 et 3, nous avons décrit et proposé un ensemble d'algorithmes pour l'identification et la reconstruction adaptatives de processus AR à échantillons manquants. Les algorithmes décrits au chapitre 2 sont fondés sur le principe du gradient alors que ceux décrits au chapitre 3 sont basés sur le principe des moindres carrés récursifs. Or ces algorithmes, et à notre connaissance, tous les algorithmes déjà proposés dans la littérature pour le traitement en ligne des signaux à échantillons manquants utilisent une représentation directe du filtre. La stabilité du filtre tout pôle correspondant aux paramètres estimés n'est donc pas garantie. La stabilisation d'un filtre IIR en contraignant les pôles à être à chaque instant à l'intérieur du cercle unité (de module inférieur à 1) est d'une grande complexité calculatoire. D'autre part, lorsque la probabilité de perte des échantillons est élevée, les méthodes existantes peuvent converger très lentement voire même diverger. La divergence est principalement due à la grande variance des paramètres estimés et à l'identification de modèles correspondant à des filtres instables. Pour cela, on s'intéresse dans ce chapitre à l'identification du signal en utilisant la structure en treillis du filtre. Celle-ci est caractérisée par les coefficients de réflexion. On note $k^{(l)}$ le coefficient de réflexion correspondant à l'étage l du treillis. Les coefficients de réflexion estimés à partir de données grâce à des méthodes telles que celles proposées dans [19, 35, 49] garantissent la stabilité du filtre identifié. Ils déterminent de manière unique les paramètres AR grâce à l'algorithme de Durbin Levinson [23, 40].

Plusieurs algorithmes adaptatifs d'identification de filtres en treillis tels que les algorithmes des moindres carrés récursifs en treillis (MCRT) sont proposés dans la littérature [28, 38, 39]. Dans [28], Friedlander présente un article didacticiel à propos des filtres en treillis et leur application au traitement adaptatif de séries temporelles. Dans le but de l'analyse par prédiction linéaire de signaux non stationnaires, Makhoul *et al.* [49] proposent une méthode générale pour la mise à jour adaptative des coefficients de réflexion. Cette méthode est une version adaptative de l'algorithme de Burg. La valeur absolue des coefficients de réflexion, estimés en utilisant cette méthode, est toujours inférieure à 1. La stabilité du filtre tout pôle correspondant est alors garantie. De plus cette méthode est simple et bien adaptée à l'identification de processus non stationnaires. Nous nous intéressons alors à son extension à l'identification et la reconstruction de signaux non stationnaires à échantillons manquants en utilisant un filtre de Kalman. Nous proposons, à notre connaissance, le premier algorithme en treillis qui permet le traitement en ligne des signaux à échantillons manquants. Outre la stabilité du modèle identifié, cet algorithme

montre une bonne dynamique de convergence même lorsque la probabilité de perte d'échantillons est élevée. Il est donc adapté pour la reconstruction de signaux de parole.

Ainsi, on commence dans ce qui suit par quelques rappels sur les filtres en treillis, en s'intéressant particulièrement à l'estimation adaptative des coefficients de réflexion.

4.2 Identification adaptative de filtres en treillis

4.2.1 La structure en treillis du filtre

Les filtres en treillis sont décrits dans de nombreux textes tels que [28]. La structure en treillis du filtre est intéressante particulièrement lorsqu'on souhaite contrôler la stabilité de celui-ci. Elle est directement induite de l'algorithme de Levinson [40]. Une représentation en treillis d'un filtre tout pôle est donnée par la figure 4.1. Elle fait intervenir les coefficients de réflexion ainsi que les erreurs de prédiction directe et rétrograde notés respectivement $k_n^{(l)}$, $f_n^{(l)}$ et $b_n^{(l)}$ à l'instant n et à l'étage l du treillis.

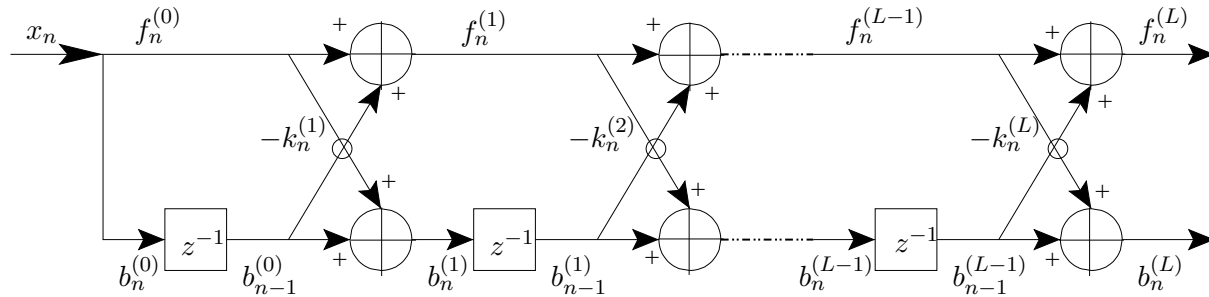


FIG. 4.1 – Schéma de base d'un filtre tout pôle en treillis.

Soit x_n le processus d'entrée au filtre en treillis. On note $\{a_i^{(L)}\}$ les coefficients du prédicteur linéaire optimal directe d'ordre L (paramètres du modèle AR(L)) et $\{\alpha_i^{(L)}\}$ ceux du prédicteur linéaire optimal rétrograde. Dans le cas de signaux scalaires stationnaires, les coefficients du prédicteur linéaire rétrograde sont obtenus en inversant l'ordre de ceux du prédicteur linéaire directe. Ainsi pour un processus AR(L), les coefficients des prédicteurs directe et rétrograde vérifient l'égalité suivante : $\alpha_i^{(L)} = a_{L-i}^{(L)}$, pour $0 \leq i \leq L$ sachant que $\alpha_L^{(L)} = a_0^{(L)} = 1$. A un instant n , les erreurs de prédictions directe et rétrograde pour un prédicteur d'ordre L sont alors données par les équations suivantes :

$$f_n^{(0)} = b_n^{(0)} = x_n, \quad (4.1)$$

$$f_n^{(L)} = x_n - \sum_{i=1}^L a_i^{(L)} x_{n-i}, \quad (4.2)$$

$$b_n^{(L)} = x_{n-L} - \sum_{i=1}^L a_i^{(L)} x_{n-L+i} \quad (4.3)$$

En se référant à la figure 4.1, les différentes grandeurs se propageant dans le filtre sont reliées, à un étage l correspondant à un modèle d'ordre l , par les équations de récurrence sur l'ordre

suivantes :

$$f_n^{(l)} = f_n^{(l-1)} - k_n^{(l)} b_{n-1}^{(l-1)}, \quad (4.4a)$$

$$b_n^{(l)} = b_{n-1}^{(l-1)} - k_n^{(l)} f_n^{(l-1)}. \quad (4.4b)$$

sachant que :

$$f_n^{(0)} = b_n^{(0)} = x_n. \quad (4.5)$$

Pour un prédicteur d'ordre L , les équations de récurrence (4.4a), and (4.4b) sont exécutées pour $1 \leq l \leq L$. Une structure en treillis telle que celle représentée à la figure 4.1 correspond à un filtre tout pôle stable lorsque tous les coefficients de réflexion vérifient :

$$|k_n^{(l)}| < 1, \quad l \geq 1 \quad (4.6)$$

Les paramètres du modèle $AR(L)$ sont déterminés de manière unique à partir des coefficients de réflexion grâce aux équations de récurrence suivantes :

- De $l = 0$ jusqu'à L

$$a_{l,n}^{(l)} = k_n^{(l)}, \quad (4.7)$$

- Pour $i = 1$ jusqu'à $l - 1$

$$a_{i,n}^{(l)} = a_{i,n}^{(l-1)} + k_n^{(l)} a_{l-i,n}^{(l-1)}, \quad (4.8)$$

- fin,
- fin.

Il est à noter qu'en exécutant les équations de récurrence précédentes de $l = 0$ jusqu'à L , les coefficients $(a_{i,n}^{(l)})_{1 \leq i \leq l}$, obtenus à chaque récurrence, correspondent à ceux du prédicteur optimal d'ordre l . Ceci découle d'un avantage de la représentation en treillis qui est la décorrélation des étages successifs. Ceci est une propriété importante des filtres en treillis selon laquelle le filtre en treillis correspondant à un prédicteur d'ordre L contient en lui tous les filtres correspondant aux prédicteurs d'ordres inférieurs. Ainsi, les l premiers étages d'un filtre en treillis forment le filtre en treillis du prédicteur d'ordre l .

Dans le paragraphe suivant, nous présentons les principales méthodes d'estimation des coefficients de réflexion à partir d'un bloc de données scalaires.

4.2.2 Calcul des coefficients de réflexion : estimation par bloc

Plusieurs méthodes existent pour le calcul des coefficients de réflexion. Elles peuvent correspondre à la minimisation d'un critère qui peut être une fonction de l'erreur de prédiction directe, rétrograde ou même une fonction des deux. Grâce à la propriété de décorrélation des étages successifs d'un treillis, l'optimisation d'un critère global de dimensions L est alors remplacée par une suite de L optimisations qui se font étage par étage. Les méthodes présentées ici sont des méthodes par bloc, elles ne prennent pas en considération l'évolution au cours du temps des coefficients de réflexion. Pour cela, on ne tient pas compte de l'indice temporel de ces derniers dans les équations suivantes.

Méthode directe Elle consiste à calculer le coefficient de réflexion à l'étage l en minimisant l'erreur quadratique moyenne de prédiction directe à cet étage [47, 48]. Ainsi, en remplaçant l'erreur de prédiction directe par son expression donnée par l'équation (4.4a), et en minimisant par rapport au coefficient de réflexion, on obtient :

$$k^{(l)} = k^{f(l)} = \frac{E \left\{ f_n^{(l-1)} b_{n-1}^{(l-1)} \right\}}{E \left\{ b_{n-1}^{(l-1)2} \right\}} \quad (4.9)$$

Le critère minimisé étant l'erreur quadratique moyenne de prédiction directe, cette méthode est similaire à la méthode d'autocorrélation ou d'autocovariance décrites au paragraphe 1.2.2 du chapitre 1.

Méthode rétrograde Elle consiste à calculer le coefficient de réflexion à l'étage l en minimisant l'erreur quadratique moyenne de prédiction rétrograde à cet étage [47, 48].

$$k^{(l)} = k^{b(l)} = \frac{E \left\{ f_n^{(l-1)} b_{n-1}^{(l-1)} \right\}}{E \left\{ f_n^{(l-1)2} \right\}} \quad (4.10)$$

Il est à noter ici que, les dénominateurs des équations (4.9) et (4.10) étant toujours positif, les coefficients de réflexion calculés par ces équations sont de même signe S

$$S = \text{signe } k^{f(l)} = \text{signe } k^{b(l)}. \quad (4.11)$$

Lorsque le signal identifié est stationnaire, les coefficients de réflexion calculés par (4.9) et (4.10) sont égaux, $k^{f(l)} = k^{b(l)}$.

L'inconvénient majeur de ces deux méthodes est que la valeur absolue des coefficients de réflexion n'est pas garantie de valeur inférieure à 1. Par conséquent, la stabilité du filtre tout pôle correspondant n'est pas garantie. Itakura *et al.* [35] proposent, comme solution à ce problème, la méthode de la moyenne géométrique.

Méthode de la moyenne géométrique Itakura *et al.* [35] proposent de calculer à chaque étage du filtre un seul coefficient de réflexion qui est égal à la moyenne géométrique des coefficients calculés par les méthodes directe et rétrograde. Les coefficients de réflexion ainsi calculés sont appelés PARCOR.

$$k^{(l)} = S \sqrt{k^{f(l)} k^{b(l)}} = \frac{E \left\{ f_n^{(l-1)} b_{n-1}^{(l-1)} \right\}}{\sqrt{E \left\{ f_n^{(l-1)2} \right\} E \left\{ b_{n-1}^{(l-1)2} \right\}}} \quad (4.12)$$

Les coefficients PARCOR ont toujours une valeur absolue inférieure à 1, ce qui permet de garantir la stabilité du filtre tout pôle correspondant.

Méthode du minimum On déduit des propriétés de la moyenne géométrique la double inégalité suivante :

$$\min \left\{ \left| k^{f(l)} \right|, \left| k^{b(l)} \right| \right\} \leq \sqrt{k^{f(l)} k^{b(l)}} \leq 1. \quad (4.13)$$

Cette propriété mène à un autre choix pour le coefficient de réflexion qui garantit la stabilité du filtre ; à savoir la méthode du minimum :

$$k^{(l)} = S \min \left\{ \left| k^{f(l)} \right|, \left| k^{b(l)} \right| \right\}. \quad (4.14)$$

Méthode générale Ici, le coefficient de réflexion d'ordre l est obtenu par la moyenne généralisée d'ordre r de $k^{f(l)}$ et $k^{b(l)}$. Il s'exprime par :

$$k^{(l)} = S \left[\frac{1}{2} \left(|k^{f(l)}|^r + |k^{b(l)}|^r \right) \right]^{1/r}. \quad (4.15)$$

Le coefficient de réflexion ainsi défini ne garantit la stabilité du filtre que lorsque $r \leq 0$. En particulier, lorsque $r \rightarrow 0$, on retrouve la méthode de la moyenne géométrique, alors que pour $r \rightarrow -\infty$, on retrouve la méthode du minimum. Le choix de $r = -1$ mène à la méthode de la moyenne harmonique.

Méthode de la moyenne harmonique aussi connue sous le nom de la méthode de Burg [19]. Le coefficient de réflexion à l'étage l du treillis est obtenu par la moyenne harmonique des coefficients calculés par les méthodes directe et rétrograde.

$$k^{(l)} = \frac{2E \left\{ f_n^{(l-1)} b_{n-1}^{(l-1)} \right\}}{E \left\{ f_n^{(l-1)2} \right\} + E \left\{ b_{n-1}^{(l-1)2} \right\}} \quad (4.16)$$

Le coefficient ainsi calculé vérifie toujours la condition (4.6). De plus, cette méthode possède une propriété importante : le coefficient de réflexion résulte de la minimisation de la somme des erreurs quadratiques moyennes de prédiction directe et rétrograde.

Dans le cas d'un signal stationnaire, toutes les méthodes précédentes aboutissent à un même coefficient de réflexion. Cependant, en pratique, les espérances mathématiques sont remplacées par des moyennes statistiques calculées à partir d'un nombre fini d'échantillons et le signal identifié n'est pas toujours stationnaire. Les différentes méthodes décrites ci-dessus donnent donc des coefficients de réflexion différents. Un choix entre ces différentes méthodes est donc à faire. On préfère en général utiliser le coefficient de Burg pour la stabilité qu'il garantit au filtre et pour la pertinence du critère qu'il minimise.

Les méthodes décrites ci-dessus permettent une estimation par bloc des coefficients de réflexion. Or nous nous intéressons dans le cadre de ce travail à l'estimation adaptative de processus non stationnaires. Ainsi, nous présentons dans ce qui suit l'identification adaptative de filtre en treillis. Les coefficients de réflexion estimés varient au cours du temps, il convient donc d'introduire l'indice temporel n dans la notation de ces derniers, $k_n^{(l)}$.

4.2.3 Identification adaptative du filtre en treillis

Alors que les algorithmes de MCR 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. Les récurrences sur le temps permettent la mise à jour des coefficients de réflexion $k_{n+1}^{(l)}$ pour $1 \leq l \leq L$, à l'instant $n+1$ à partir de leurs valeurs à l'instant précédent sachant toutes les erreurs de prédiction directes et rétrogrades jusqu'à l'instant n et la valeur du signal à l'instant courant. Les récurrences sur l'ordre l du prédicteur permettent de calculer à chaque instant les erreurs de prédiction directe et rétrograde.

Plusieurs algorithmes adaptatifs en treillis sont proposés dans la littérature [28]. Ils sont fondés sur le principe du gradient ou sur celui des moindres carrés. Chacun des algorithmes en treillis surpasse en vitesse et en qualité de convergence l'algorithme correspondant utilisant la structure transversale du filtre. On s'intéresse particulièrement aux algorithmes des moindres

carrés récursifs en treillis (MCRT) pour leur vitesse de convergence et leur capacité de poursuite. De plus, ils sont efficaces en termes de complexité calculatoire. Ils ne nécessitent que $O(L)$ opérations à chaque pas de temps en comparaison à $O(L^2)$ pour l'algorithme MCR qui utilise la structure directe du filtre transverse. Ainsi dans [38], une approche géométrique est utilisée afin d'établir les équations de récurrence temporelles d'un algorithme MCRT. Pour l'estimation d'un coefficient de réflexion, ils utilisent la propriété de décorrélation des étages successifs d'un treillis. Cette propriété est approximativement vraie dans le cas non stationnaire. Ainsi, les coefficients de réflexion à un étage du treillis sont calculés en minimisant simultanément les erreurs quadratiques moyennes de prédiction directe et rétrograde du même étage. A chaque instant, deux coefficients de réflexion différents sont alors estimés pour chaque étage du treillis, ils correspondent respectivement à la méthode directe et à la méthode rétrograde. Or la valeur absolue des coefficients ainsi estimés n'est pas garantie inférieure à 1.

Les algorithmes MCR en treillis normalisé [28, 38, 39] sont plus performants en terme de complexité calculatoire que ceux sans normalisation. Ils nécessitent trois relations de récurrence contre neuf dans le cas sans normalisation. Deux de ces relations de récurrence sont sur l'ordre et permettent la mise à jour des erreurs de prédiction directe et rétrograde normalisées. La troisième est une récurrence temporelle permettant la mise à jour des coefficients de réflexion qui vérifient tous la condition de stabilité donnée par l'équation (4.6). Ces algorithmes ont montré de plus le même comportement en terme de vitesse de convergence et de capacité de poursuite que ceux sans normalisation [38, 39].

L'algorithme MCR en treillis normalisé se résume aux étapes suivantes [28, 38, 39] :

- Pour $n = 1$ jusqu'à N (N étant la taille du signal)
 - Initialisation des récurrences sur l'ordre par les équations suivantes :

$$\hat{v}_x = \lambda \hat{v}_x + x_n^2, \quad (4.17a)$$

$$\tilde{f}_n^{(0)} = \tilde{b}_n^{(0)} = \hat{v}_x^{-1} x_n, \quad (4.17b)$$

$$k_n^{(0)} = 1, \quad (4.17c)$$

- Pour $l = 1$ jusqu'à L
 - Mise à jour du coefficient de réflexion à l'étage l grâce à l'équation de récurrence sur le temps suivante :

$$k_n^{(l)} = k_{n-1}^{(l)} \sqrt{\left(1 - \tilde{f}_n^{(l-1)2}\right) \left(1 - \tilde{b}_{n-1}^{(l-1)2}\right)} + \tilde{f}_n^{(l-1)} \tilde{b}_{n-1}^{(l-1)}, \quad (4.18)$$

- Mise à jour des erreurs de prédictions normalisées par les récurrences sur l'ordre suivantes :

$$\tilde{f}_n^{(l)} = \frac{\tilde{f}_n^{(l-1)} - k_n^{(l)} \tilde{b}_{n-1}^{(l-1)}}{\sqrt{\left(1 - \tilde{b}_{n-1}^{(l-1)2}\right) \left(1 - k_n^{(l)2}\right)}}, \quad (4.19a)$$

$$\tilde{b}_n^{(l)} = \frac{\tilde{b}_{n-1}^{(l-1)} - k_n^{(l)} \tilde{f}_n^{(l-1)}}{\sqrt{\left(1 - \tilde{f}_n^{(l-1)2}\right) \left(1 - k_n^{(l)2}\right)}}, \quad (4.19b)$$

- fin,
- fin.

où $\lambda \leq 1$ est le facteur d'oubli, $\tilde{f}_n^{(l)}$ et $\tilde{b}_n^{(l)}$ sont les erreurs de prédiction directe et rétrograde normalisées correspondant à l'étage l et à l'instant n , \hat{v}_x est l'estimée pondérée de la variance du processus centré x_n .

D'autre part, Makhoul *et al.* [49, 50] proposent une méthode générale d'estimation adaptative des coefficients de réflexion pour l'analyse par prédiction linéaire de processus non stationnaires. Ils utilisent aussi la propriété de décorrélation des étages successifs d'un treillis. Ainsi, ils calculent, à chaque étage l du treillis, un coefficient de réflexion $k_n^{(l)}$ qui minimise la fonction de coût suivante :

$$J_n^{(l)} = \sum_{i=-\infty}^n w_{n-i} \left(f_i^{(l)2} + b_i^{(l)2} \right). \quad (4.20)$$

où w_n est une fenêtre ou une suite de pondération des erreurs quadratiques de prédiction aux instants précédents. En pratique, la fenêtre doit vérifier la condition de causalité, $w_n = 0$ pour $n < 0$. De plus, ils montrent que lorsque $w_n \geq 0$ pour $n \geq 0$, le coefficient de réflexion qui minimise (4.20) vérifie toujours (4.6) [50]. Dans un contexte non stationnaire, il est convenable de pondérer la somme des erreurs quadratiques de prédiction directe et rétrograde par un facteur d'oubli exponentiel $0 < \lambda \leq 1$, donnant ainsi plus d'importance aux valeurs récentes. La suite de pondération est alors définie par :

$$w_n = 0, \quad n < 0 \quad (4.21)$$

$$w_n = \lambda^n, \quad n \geq 0. \quad (4.22)$$

Ainsi en remplaçant dans l'équation (4.20) les erreurs de prédiction par leurs expressions données par (4.4), et en minimisant par rapport à $k_n^{(l)}$, on obtient :

$$k_n^{(l)} = - \frac{2 \sum_{i=1}^n \lambda^{n-i} f_n^{(l-1)} b_{n-1}^{(l-1)}}{\sum_{i=1}^n \lambda^{n-i} \left[f_n^{(l)2} + b_n^{(l)2} \right]}, \quad (4.23a)$$

$$= - \frac{C_n^{(l)}}{D_n^{(l)}}. \quad (4.23b)$$

Le coefficient ainsi calculé vérifie toujours (4.6). Il est à noter que pour $\lambda = 1$, on retrouve le coefficient de Burg [19]. L'estimateur du coefficient de réflexion donné par (4.23) est une adaptation du coefficient de Burg au cas de signaux non stationnaires. Makhoul *et al.* [49, 50] proposent une estimation adaptative de $k_n^{(l)}$ grâce aux relations de récurrence temporelles suivantes :

$$C_{n+1}^{(l)} = \lambda C_n^{(l)} + 2 f_{n+1}^{(l-1)} b_n^{(l-1)}, \quad (4.24a)$$

$$D_{n+1}^{(l)} = \lambda D_n^{(l)} + f_{n+1}^{(l-1)2} + b_n^{(l-1)2}, \quad (4.24b)$$

$$k_{n+1}^{(l)} = - \frac{C_{n+1}^{(l)}}{D_{n+1}^{(l)}}, \quad (4.24c)$$

En utilisant ces relations, Makhoul *et al.* déduisent une expression de $k_{n+1}^{(l)}$ en fonction de $k_n^{(l)}$ et lui donnent une interprétation de type gradient ainsi que de type filtre de Kalman [49].

Un algorithme adaptatif pour l'estimation des coefficients de réflexion est alors obtenu en combinant les relations de récurrence temporelles données par (4.24) et les relations de récurrence sur l'ordre données par (4.4). Il se résume à l'instant $n + 1$ aux étapes suivantes : (On suppose ici qu'on a déjà gardé en mémoire $C_n^{(l)}$, $D_n^{(l)}$ et $b_n^{(l)}$ pour $1 \leq l \leq L$ calculés à l'instant précédent.)

- Initialisation pour $l = 0$

$$f_{n+1}^{(0)} = b_{n+1}^{(0)} = x_{n+1}, \quad (4.25a)$$

$$k_{n+1}^{(0)} = 1, \quad (4.25b)$$

- Pour $l = 1$ jusqu'à $\min(L, n)$
 - Estimation récursive de $C_{n+1}^{(l)}, D_{n+1}^{(l)}$ grâce aux équations (4.24a) et (4.24b).
 - En déduire $k_{n+1}^{(l)}$ grâce à l'équation (4.24c). Il est à noter que si le dénominateur $D_{n+1}^{(l)}$ est nul, $k_{n+1}^{(l)}$ est considéré également nul.
 - Estimation de $f_{n+1}^{(l)}$ et $b_{n+1}^{(l)}$ grâce aux équations de récurrence sur l'ordre (4.4a) et (4.4b).
- fin.

Cet algorithme, appelé l'algorithme de Burg adaptatif, permet d'estimer des coefficients de réflexion qui sont de valeur absolue inférieure à 1. De plus, les simulations montrent qu'il a un bon comportement de convergence similaire à celui des algorithmes MCR en treillis normalisé. De surcroît, il est d'une complexité calculatoire plus faible que ces derniers. On propose alors son extension à l'identification de processus AR à échantillons manquants en utilisant un filtre de Kalman tel que celui présenté au paragraphe 3.2 du chapitre 3 pour la prédiction des échantillons manquants.

4.3 Prédiction en utilisant un filtre de Kalman

On rappelle ici brièvement la prédiction d'un processus AR(L) à échantillons manquants en utilisant un filtre de Kalman. Cette approche de prédiction a été décrite au paragraphe 3.2 du chapitre 3.

4.3.1 Représentation d'état

Le processus AR, $\{x_n\}$, d'ordre L vérifie l'équation aux différences suivante : $x_n = a_1^{(L)}x_{n-1} + \dots + a_L^{(L)}x_{n-L} + \epsilon_n$, ϵ_n étant le processus d'innovation. La perte des échantillons est caractérisée par une variable binaire, $\{c_n\}$. L'observation est donc donnée par $y_n = c_n x_n$ où $y_n = x_n$ si l'échantillon est disponible, sinon $y_n = 0$. Le processus $\{x_n\}$ ayant subi une perte d'échantillons admet la représentation d'état suivante :

$$\begin{cases} \mathbf{x}_{n+1} &= A \mathbf{x}_n + [1 \ 0 \ \dots \ 0]^\top \epsilon_{n+1} \\ y_{n+1} &= \mathbf{c}_{n+1}^\top \mathbf{x}_{n+1} \end{cases} \quad (4.26)$$

$$\text{où } A = \left[\begin{array}{ccc|c} a_1^{(L)} & \dots & \dots & a_L^{(L)} \\ 1 & & & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \end{array} \right] \mathbf{x}_n = \begin{bmatrix} x_n \\ \vdots \\ x_{n-L+1} \end{bmatrix}, \mathbf{c}_n = \begin{bmatrix} c_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

L'échantillon qu'on souhaite prédire est à chaque instant le premier élément de l'état \mathbf{x}_{n+1} . Il est à noter ici qu'à chaque pas de temps, l'estimée des paramètres du modèle ainsi que la valeur de c_n (la disponibilité de l'échantillon) sont connues, la représentation d'état précédente correspond alors à un filtre de Kalman non stationnaire. On rappelle dans ce qui suit brièvement le filtre de Kalman qui permet l'estimation de l'état.

4.3.2 Filtre de Kalman

À l'instant $n + 1$, les estimées *a priori* et filtrée de l'état sont notées $\hat{\mathbf{x}}_{n+1|n}$ et $\hat{\mathbf{x}}_{n+1|n+1}$. Les matrices de covariance de l'erreur d'estimation correspondantes sont notées respectivement $P_{n+1|n}$ et $P_{n+1|n+1}$. K_{n+1} est le gain du filtre de Kalman.

Le filtre de Kalman permettant d'estimer l'état défini par l'équation (4.26) est décrit et analysé au paragraphe 3.2 du chapitre 3, nous rappelons ici brièvement les principales étapes exécutées à l'instant $n + 1$:

- L'estimée *a priori* de l'état, $\hat{\mathbf{x}}_{n+1|n}$, et celle de la matrice de covariance de l'erreur d'estimation correspondante, $P_{n+1|n}$, sont calculées grâce aux équations (3.17) et (3.16).
- Si l'échantillon x_{n+1} est disponible, $c_{n+1} = 1$, l'observation est donc égale à la valeur de celui-ci, $y_{n+1} = x_{n+1}$: le gain du filtre de Kalman, K_{n+1} , est calculée par l'équation (3.20). L'estimée de l'état est alors filtrée selon l'équation (3.18), et la matrice de covariance de l'erreur d'estimation de l'état est mise à jour grâce à l'équation (3.19).
- Si l'échantillon est manquant, $c_{n+1} = 0$, le gain du filtre de Kalman est nul $K_{n+1} = 0$. Il s'en suit que l'estimée de l'état ainsi que la matrice de covariance de l'erreur d'estimation sont conservées *a posteriori*, $\hat{\mathbf{x}}_{n+1|n+1} = \hat{\mathbf{x}}_{n+1|n}$ et $P_{n+1|n+1} = P_{n+1|n}$.

On rappelle aussi que si x_{n+1} est disponible, le premier élément du vecteur $\hat{\mathbf{x}}_{n+1|n+1}$, qu'on note dans la suite $\hat{x}_{n+1|n+1}$, est égal à x_{n+1} . Ainsi, un échantillon disponible est inchangé par le filtre de Kalman. Par contre, si x_{n+1} est manquant, $\hat{x}_{n+1|n+1}$ est sa prédiction optimale à l'instant $n + 1$. Ainsi, le signal reconstruit qui prend la valeur du signal lorsque l'échantillon est disponible et sa prédiction dans le cas contraire, est donné à tout instant $n + 1$ par : $z_{n+1} = \hat{x}_{n+1|n+1}$.

Il convient ici de rappeler que si x_{n+1} est manquant, son estimée mise à jour au cours des $L - 1$ pas de temps suivants est différente de sa prédiction fournie à l'instant $n + 1$, $\hat{x}_{n+1|n+1+t} \neq \hat{x}_{n+1|n+1}$ pour $0 < t \leq L - 1$. En effet, l'état contient en lui les $L - 1$ derniers échantillons ou leurs prédictions s'ils sont manquants. Si un échantillon est disponible, l'état est filtré par le filtre de Kalman selon l'équation (3.18). Par conséquent, des termes s'ajoutent à la prédiction des échantillons manquants existants dans cet état. On note ici que le filtrage de Kalman ajoute des termes uniquement à la prédiction des échantillons manquants, les échantillons disponibles restent inchangés. Ainsi, l'estimée d'un échantillon manquant mise à jour au cours des $L - 1$ pas de temps suivants s'exprime par l'équation (3.28).

4.4 Algorithme de Burg adaptatif combiné à un filtre de Kalman

Dans ce qui suit, on propose un algorithme adaptatif qui permet l'identification en treillis et la reconstruction d'un processus AR(L) à échantillons manquants. Il utilise conjointement l'algorithme de Burg adaptatif pour une identification en treillis du processus et le filtre de Kalman, évoqué ci-dessus au paragraphe 4.3, pour une reconstruction optimale adaptative des échantillons manquants. Le modèle ainsi identifié correspond à chaque pas de temps à un filtre stable.

En se référant aux équations (4.24), on constate que la mise à jour des coefficients de réflexion utilise les erreurs de prédiction rétrograde calculés à l'instant précédent. Il en est de même pour le calcul des erreurs de prédiction directe et rétrograde grâce aux équations de récurrence sur l'ordre (4.4). On en déduit alors par récurrence qu'il est nécessaire pour le calcul des coefficients de réflexion à un instant n d'avoir calculé toutes les variables du filtre en treillis aux instants précédents. Or les équations de récurrence sur l'ordre qui permettent de calculer les erreurs de prédiction sont initialisées à chaque pas de temps par la valeur du signal (équation (4.5)). On ne peut donc pas les utiliser pour calculer les erreurs de prédiction lorsqu'un échantillon

est manquant. De toutes façons, il n'est pas possible de calculer l'erreur de prédiction d'un échantillon manquant. Il est donc nécessaire d'estimer à cet instant là les erreurs de prédiction, elles seront utilisées pour la mise à jour des variables du treillis à l'instant suivant. Ainsi, dans le cas de signaux à échantillons manquants, la fonction de coût minimisée par l'algorithme proposé est une estimation de celle minimisée par l'algorithme de Burg adaptatif (équation (4.20)). Elle s'exprime à l'instant n par :

$$\hat{J}_n^{(l)} = \sum_{i=1}^n w_{n-i} \left(\hat{f}_n^{(l)2} + \hat{b}_n^{(l)2} \right). \quad (4.27)$$

où $\hat{f}_n^{(l)}$ et $\hat{b}_n^{(l)}$ désignent les valeurs estimées de $f_n^{(l)}$ et $b_n^{(l)}$ respectivement.

4.4.1 Estimation optimale des erreurs de prédiction directe et rétrograde

Les erreurs de prédiction directes et rétrogrades sont définies par les équations (4.1). En utilisant le théorème des échantillons manquants [65], une estimation optimale au sens des moindres carrés des erreurs de prédiction est alors donnée à l'instant n par les équations suivantes :

$$\hat{f}_n^{(0)} = \hat{b}_n^{(0)} = \hat{x}_{n|n}, \quad (4.28a)$$

$$\hat{f}_n^{(l)} = \hat{x}_{n|n} - \sum_{i=1}^l a_{i,n}^{(l)} \hat{x}_{n-i|n}, \quad (4.28b)$$

$$\hat{b}_n^{(l)} = \hat{x}_{n-l|n} - \sum_{i=1}^l a_{i,n}^{(l)} \hat{x}_{n-l+i|n}, \quad (4.28c)$$

où, $\hat{x}_{n-i|n}$ est l'estimée optimale au sens des moindres carrés de x_{n-i} connaissant les échantillons disponibles jusqu'à l'instant n .

Les équations précédentes (4.28) sont analogues à celles qui définissent les erreurs de prédiction (4.1). La différence est que les erreurs de prédiction ainsi que la valeur du signal sont remplacées par leurs estimées respectives. Par conséquent, en remplaçant dans les équations (4.5) et (4.4) les erreurs de prédiction et la valeur du signal par leurs estimées, on obtient des équations récursives sur l'ordre permettant d'estimer les erreurs de prédiction dans le cas de signaux à échantillons manquants. Ceci revient donc à remplacer un échantillon manquant par sa prédiction dans l'initialisation des équations sur l'ordre. Ainsi, l'algorithme de Burg adaptatif se résume dans le cas de signaux à échantillons manquants à un instant n aux équations suivantes :

– Initialisation pour $l = 0$

$$\hat{f}_n^{(0)} = \hat{b}_n^{(0)} = \hat{x}_{n|n}, \quad (4.29a)$$

$$\hat{k}_n^{(0)} = 1, \quad (4.29b)$$

– Pour $l = 1$ jusqu'à $\min(L, n)$

$$C_n^{(l)} = \lambda C_{n-1}^{(l)} + 2\hat{f}_n^{(l-1)}\hat{b}_{n-1}^{(l-1)}, \quad (4.30a)$$

$$D_n^{(l)} = \lambda D_{n-1}^{(l)} + \hat{f}_n^{(l-1)2} + \hat{b}_{n-1}^{(l-1)2}, \quad (4.30b)$$

$$\hat{k}_n^{(l)} = -\frac{C_n^{(l)}}{D_n^{(l)}}, \quad (4.30c)$$

$$\hat{f}_n^{(l)} = \hat{f}_n^{(l-1)} - \hat{k}_n^{(l)}\hat{b}_{n-1}^{(l-1)}, \quad (4.30d)$$

$$\hat{b}_n^{(l)} = \hat{b}_{n-1}^{(l-1)} - \hat{k}_n^{(l)}\hat{f}_n^{(l-1)}, \quad (4.30e)$$

– fin.

On propose d'utiliser un filtre de Kalman pour une prédiction optimale au sens des moindres carrés des échantillons manquants. Or le filtre de Kalman utilise une représentation directe du filtre transverse. Il est donc nécessaire de calculer à chaque instant les paramètres AR à partir des coefficients de réflexion grâce aux équations de récurrence (4.7).

On rappelle ici que des termes s'ajoutent à la prédiction d'un échantillon manquant grâce au filtrage de l'état lorsque des échantillons sont disponibles au cours des $L - 1$ pas de temps suivants. Ainsi, soit $\hat{x}_{n|n}$ la prédiction optimale d'un échantillon manquant à l'instant n , celle-ci ne sera plus optimale si des échantillons sont disponibles au cours des $L - 1$ pas de temps suivants. Dans un contexte temps réel, l'estimée actualisée d'un échantillon, $\hat{x}_{n|n+t}$, est uniquement utilisée pour la prédiction des échantillons suivants. Par contre, le signal est reconstruit en utilisant $\hat{x}_{n|n}$. Par ailleurs, les erreurs de prédiction directe et rétrograde, estimées en utilisant la prédiction d'un échantillon manquant, seront utilisées à l'instant suivant. Ainsi, comme pour la prédiction des échantillons manquants, leurs estimées devraient aussi être actualisées pour être optimales à l'instant suivant. Pour éviter de les recalculer plusieurs fois, on propose ce qui suit :

Si l'échantillon x_n est manquant, il est prédit en utilisant le filtre de Kalman, sa prédiction est utilisée pour reconstruire le signal. Par contre, les erreurs de prédiction, $\left(f_n^{(l)}\right)_{1 \leq l \leq L}$ et $\left(b_n^{(l)}\right)_{1 \leq l \leq L}$ et les coefficients de réflexion $\left(k_n^{(l)}\right)_{1 \leq l \leq L}$ ne seront estimés que lorsque la prédiction de l'échantillon est actualisée, c.-à-d. jusqu'à ce qu'un nouvel échantillon soit disponible. Quand un échantillon est disponible à l'instant $n + t$, l'état est alors mis à jour grâce au filtre de Kalman. Nous calculons alors toutes les variables du filtre en treillis correspondant aux échantillons manquants entre l'instant du dernier échantillon disponible, $n - h$ et l'instant courant, $n + t$. Il est à noter ici que lorsque l'intervalle de temps séparant deux échantillons disponibles est supérieur à L ($t - h + 1 > L$), les prédictions des $L - 1$ échantillons manquants précédents (ceux qui sont contenus dans l'état) sont mis à jour. Les prédictions des échantillons manquants qui précèdent ces derniers (aux instants $n - h + 1$ jusqu'à $n + t - L$) ne le sont pas. Malgré cela, il est nécessaire de calculer les variables du filtre en treillis à ces instants là même si aucune information nouvelle n'est disponible. Néanmoins, les paramètres du modèle AR ne sont pas mis à jour à ces instants là pour économiser du temps de calcul.

4.4.2 Algorithme proposé

L'algorithme proposé se résume donc à l'instant $n + 1$ aux étapes suivantes : La première ligne de la matrice A est remplacée par $\hat{\mathbf{a}}_n^{(L)\top}$, l'estimée du vecteur des paramètres à l'instant n . Elle est alors notée A_{n+1} .

$$A_{n+1} = \left[\begin{array}{ccc|c} \hat{a}_{1,n}^{(L)} & \dots & \dots & \hat{a}_{L,n}^{(L)} \\ 1 & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & 1 & 0 \end{array} \right], \quad (4.31)$$

$$P_{n+1|n} = A_{n+1}P_{n|n}A_{n+1}^\top + R_\epsilon,$$

$$\hat{\mathbf{x}}_{n+1|n} = A_{n+1}\hat{\mathbf{x}}_{n|n},$$

$$\hat{y}_{n+1|n} = c_{n+1}\hat{x}_{n+1|n},$$

Si x_{n+1} est disponible, i.e. $c_{n+1} = 1$

$$K_{n+1} = P_{n+1|n} \mathbf{c}_{n+1} (\mathbf{c}_{n+1}^\top P_{n+1|n} \mathbf{c}_{n+1})^{-1}, \quad (4.32a)$$

$$P_{n+1|n+1} = (I_d - K_{n+1} \mathbf{c}_{n+1}^\top) P_{n+1|n}, \quad (4.32b)$$

$$\hat{\mathbf{x}}_{n+1|n+1} = \hat{\mathbf{x}}_{n+1|n} + K_{n+1} (y_{n+1} - \hat{y}_{n+1|n}). \quad (4.32c)$$

Ainsi, la prédiction des échantillons manquants précédents jusqu'à l'instant $n-L+1$ (existants dans l'état) est alors mise à jour grâce au filtrage de l'état par l'équation (4.32c). Il convient maintenant de calculer les variables du filtre en treillis à partir de la dernière observation à l'instant $n-h$, où $h \geq 0$ dépend du schéma de perte. Ainsi à chaque instant t , pour $n-h+1 \leq t \leq n+1$, l'algorithme de Burg adaptatif dans le cas de signaux à échantillons manquants est utilisé pour calculer les coefficients de réflexion $\hat{k}_t^{(l)}$ et les erreurs de prédiction $\hat{f}_t^{(l)}, \hat{b}_t^{(l)}$ pour $1 \leq l \leq L$. Les erreurs de prédiction directe et rétrograde sont initialisées par la prédiction actualisée des échantillons manquants (celles-ci sont contenues dans l'état filtré $\hat{\mathbf{x}}_{n+1|n+1}$), i.e. $\hat{f}_t^{(0)} = \hat{b}_t^{(0)} = \hat{x}_{t|n+1}$. Ainsi,

- Pour $t = n-h+1$ jusqu'à $n+1$
 - Initialisation pour $l = 0$

$$\hat{f}_t^{(0)} = \hat{b}_t^{(0)} = \hat{x}_{t|n+1}, \quad (4.33a)$$

$$\hat{k}_t^{(0)} = 1, \quad (4.33b)$$

- Pour $l = 1$ jusqu'à $\min(L, n)$

$$C_t^{(l)} = \lambda C_{t-1}^{(l)} + 2 \hat{f}_t^{(l-1)} \hat{b}_{t-1}^{(l-1)}, \quad (4.34a)$$

$$D_t^{(l)} = \lambda D_{t-1}^{(l)} + \hat{f}_t^{(l-1)2} + \hat{b}_{t-1}^{(l-1)2}, \quad (4.34b)$$

$$\hat{k}_t^{(l)} = -\frac{C_t^{(l)}}{D_t^{(l)}}, \quad (4.34c)$$

$$\hat{f}_t^{(l)} = \hat{f}_t^{(l-1)} - \hat{k}_t^{(l)} \hat{b}_{t-1}^{(l-1)}, \quad (4.34d)$$

$$\hat{b}_t^{(l)} = \hat{b}_{t-1}^{(l-1)} - \hat{k}_t^{(l)} \hat{f}_t^{(l-1)}, \quad (4.34e)$$

- fin,
- fin.

Enfin, les paramètres $(\hat{a}_{i,n+1}^{(L)})_{1 \leq i \leq L}$ sont estimés à partir des coefficients de réflexion $(\hat{k}_{n+1}^{(l)})_{1 \leq l \leq L}$ grâce à l'algorithme de Durbin Levinson donné par les équations (4.7) et (4.8).

Par contre si x_{n+1} est manquant, $c_{n+1} = 0$ l'état estimé *a priori*, $\hat{\mathbf{x}}_{n+1|n}$, n'est pas filtré par le filtre de Kalman. Les coefficients de réflexion $(\hat{k}_{n+1}^{(l)})_{1 \leq l \leq L}$ n'étant pas encore estimés, les paramètres du modèle AR ne sont pas mis à jour.

$$K_{n+1} = 0, \quad (4.35a)$$

$$P_{n+1|n+1} = P_{n+1|n}, \quad (4.35b)$$

$$\hat{\mathbf{x}}_{n+1|n+1} = \hat{\mathbf{x}}_{n+1|n}, \quad (4.35c)$$

$$\hat{\mathbf{a}}_{n+1}^{(L)} = \hat{\mathbf{a}}_n^{(L)}. \quad (4.35d)$$

Un organigramme représentant les différentes étapes de cet algorithme est présenté aux figures 4.6 et 4.7. Cet algorithme permet d'identifier le modèle optimal stable au sens du critère donné par l'équation (4.27).

La fonction de coût minimisée par l'algorithme Kalman MCR pseudo-linéaire, décrit au chapitre 3, est la moyenne pondérée des erreurs quadratiques de prédiction aux instants où les échantillons sont disponibles. Ainsi, les paramètres estimés ne sont pas mis à jour par l'algorithme Kalman MCR pseudo-linéaire lorsqu'un échantillon est manquant. En effet, la mise à jour des paramètres grâce à l'algorithme MCR est proportionnelle à l'erreur de prédiction qui ne peut pas être calculée lorsqu'un échantillon est manquant. Par contre, la fonction de coût minimisée par l'algorithme proposé ici est la moyenne pondérée de toutes les erreurs quadratiques de prédiction. Lorsqu'un échantillon est manquant, les erreurs de prédictions ne peuvent pas être calculées, elles sont alors remplacées par leurs estimées. En effet, dans le cas des algorithmes en treillis, il est nécessaire pour estimer les coefficients de réflexion à un instant n d'avoir calculé toutes les variables du filtre aux instants précédents. Pour cela, les paramètres du filtre en treillis sont estimés à tous les instants par l'algorithme proposé même lorsqu'un échantillon est manquant. Ceci suggère qu'il a une bonne dynamique de convergence et de poursuite des paramètres même pour des probabilités élevées d'échantillons manquants. Cette hypothèse est confirmée grâce aux simulations.

L'algorithme proposé et l'algorithme Kalman MCR pseudo-linéaire utilisent le même filtre de Kalman pour la reconstruction, leurs complexités calculatoire ne diffèrent donc que par l'algorithme d'identification utilisé. Leurs complexités calculatoire ont été calculées, elles sont, pour les deux algorithmes, de l'ordre de $O((1-q)NL^2)$, où q est la probabilité de perte des échantillons, N la taille du signal et L l'ordre du modèle AR. Ces algorithmes ont été implémentés en langage Matlab, les simulations des deux algorithmes ont nécessité des temps de calcul voisins. Pour des modèles AR d'ordre faible, l'algorithme proposé est plus rapide que le Kalman MCR pseudo-linéaire. Par contre, il est plus lent pour des modèles AR d'ordre élevés, tels que ceux qui permettent de modéliser des signaux de parole.

4.5 Simulations

L'algorithme décrit dans ce chapitre est comparé à l'algorithme Kalman MCR pseudo-linéaire à l'aide de trois expériences. Le signal test utilisé pour la première simulation est une juxtaposition de processus AR(2) ayant subi une perte d'échantillons dont la probabilité est élevée. Dans les deux expériences suivantes, les deux algorithmes sont appliqués à la reconstruction d'un signal de parole (simulation 2) et de musique (simulation 3) pour différentes probabilités de perte d'échantillons. Leurs performances dans la reconstruction du signal est mesurée en termes du RSB (Rapport Signal sur Bruit de reconstruction).

Simulation 1 Le signal test utilisé dans cette expérience est une juxtaposition de processus AR(2), il est généré sur $15 \cdot 10^3$ échantillons. Les paramètres du signal sont $[1.5, -1]$ pour les premiers $5 \cdot 10^3$ échantillons, $[-0.5, -1]$ pour les $5 \cdot 10^3$ échantillons suivants et finalement $[0.5, -1]$ pour les échantillons restants. La probabilité de perte d'échantillon est de $q = 0.7$. Le facteur d'oubli est choisi de $\lambda = 0.99$. La figure 4.2 représente l'estimation instantanée du premier paramètre en utilisant les deux algorithmes. Elle montre que l'algorithme proposé offre une convergence plus rapide que l'algorithme Kalman MCR pseudo-linéaire même pour un nombre élevé d'échantillons absents. Pour des facteurs d'oubli plus faibles, les deux algorithmes convergent comme il se doit plus rapidement, ceci est aux dépens d'une plus grande variance des paramètres estimés.

D'autre part, les simulations montrent que la vitesse de convergence de l'algorithme Kalman MCR pseudo-linéaire dépend du schéma et de la probabilité de perte des échantillons. En effet, cette expérience a été reproduite pour 1000 générations du schéma de perte. Le tableau 4.1 montre alors la répartition des RSB fournis en utilisant chacun des deux algorithmes.

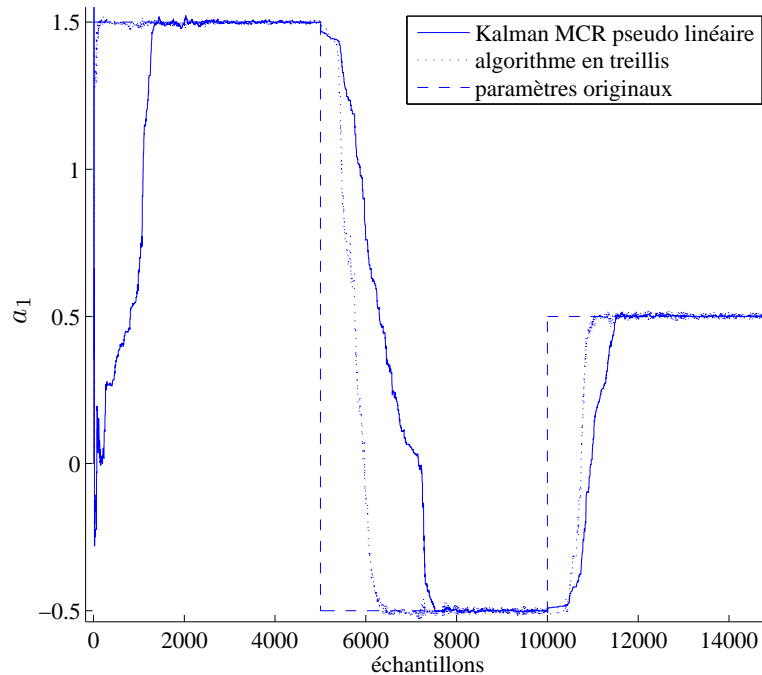


FIG. 4.2 – Estimation instantanée du paramètre a_1 en utilisant les deux algorithmes comparés.

TAB. 4.1 – Répartition des RSB fournis par chacun des deux algorithmes comparés pour 1000 générations différentes du schéma de perte.

Algorithme	RSB < 0 dB	0dB ≤ RSB < 15 dB	15 ≤ RSB
Kalman MCR pseudo-linéaire	8.1 %	9.5 %	82.4 %
algorithme en treillis	0 %	0.1 %	99.9 %

Le tableau 4.1 montre que le RSB fourni par l'algorithme Kalman MCR pseudo-linéaire est, pour 17.6 % des 1000 schémas de perte, inférieur à 15 dB alors qu'il l'est pour uniquement 0.1% des fois en utilisant l'algorithme proposé ici. Ceci est dû à un retard dans la convergence ou à une divergence. Il est à noter ici que l'algorithme Kalman MCR pseudo-linéaire a divergé pour 2% des schémas de perte. La divergence peut être due à une forte variance des paramètres estimés et à l'estimation de paramètres correspondant à des modèles instables.

Par ailleurs, la figure 4.2 semble montrer que l'algorithme proposé converge vers une estimation non biaisée des paramètres. Cependant, il existe un biais dû à la minimisation des erreurs de prédiction aux instants où les échantillons sont manquants. Ce biais est d'autant plus faible que a_2 est voisin de l'unité (à la limite de la zone de stabilité). De tels signaux correspondent à des filtres résonnants. Ces propriétés de l'algorithme permettent d'envisager une application à la reconstruction de signaux de parole ou de musique. Ceci est le cas de la deuxième série de simulations.

Simulation 2 Les deux algorithmes comparés sont appliqués à la reconstruction du signal test usuel *'Mary had a little lamb, its fleece was white as snow'* échantillonné à 8kHz. Ce signal est reconstruit en utilisant les deux algorithmes comparés pour différentes probabilités de perte d'échantillons. Pour chaque probabilité de perte, le schéma d'observation est régénéré 1000 fois. Ainsi pour chaque algorithme, la moyenne et la variance du RSB sont estimées pour chaque probabilité de perte sur l'ensemble des différents schémas d'observation. Le facteur d'oubli utilisé pour les deux algorithmes est de $\lambda = 0.99$. Pour cette valeur du facteur d'oubli, les deux algorithmes offrent de bonnes performances. Le signal de parole est modélisé par un processus AR d'ordre 12.

Les figures 4.3 et 4.4 montrent la moyenne et l'écart type du RSB pour chacun des algorithmes en fonction de la probabilité de perte d'échantillons.

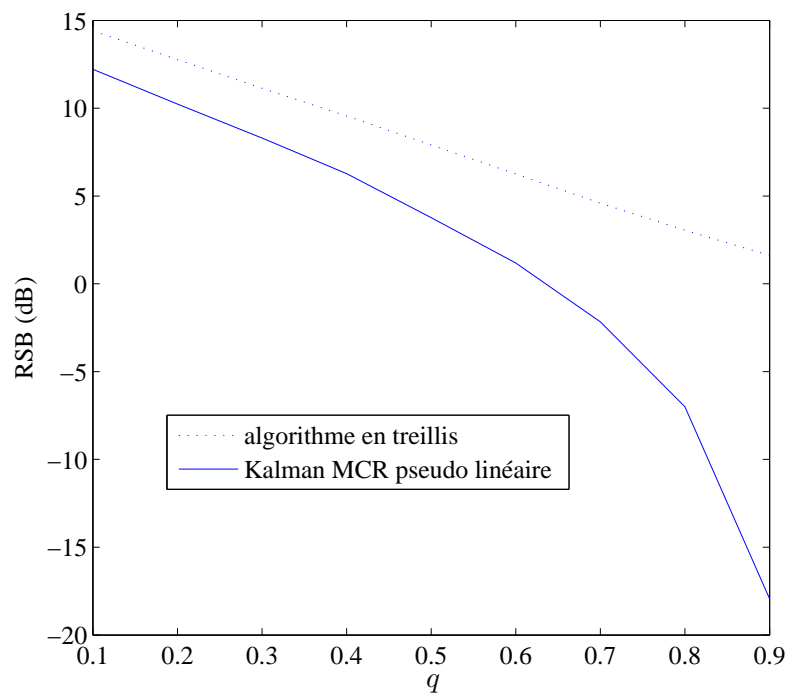


FIG. 4.3 – RSB moyen en fonction de la probabilité de perte d'échantillons pour l'expérience 2.

La figure 4.3 montre que l'algorithme proposé offre en moyenne de meilleures performances que celles obtenues grâce à l'autre algorithme en terme d'erreur de reconstruction, particulièrement pour les fortes probabilités de perte d'échantillons. Le RSB moyen obtenu grâce à l'algorithme proposé suit une décroissance linéaire lorsque la probabilité de perte croît. Par contre, en utilisant l'algorithme Kalman MCR pseudo-linéaire, la décroissance du RSB est exponentielle. Ainsi, pour une perte de 70% des échantillons, le RSB obtenu grâce à l'algorithme proposé est supérieur en moyenne de 4.5 dB de celui obtenu grâce à l'algorithme Kalman MCR pseudo-linéaire. Cet écart moyen est de 17 dB lorsque 85 % des échantillons sont manquants. Il est à noter ici que le signal reconstruit utilisant l'algorithme proposé est intelligible même lorsque 85 % des échantillons sont manquants d'une manière aléatoire.

De plus, la figure 4.4 montre que l'écart type du RSB obtenu grâce à l'algorithme proposé ici est négligeable devant celui obtenu utilisant l'autre algorithme. De surcroît, il est presque constant

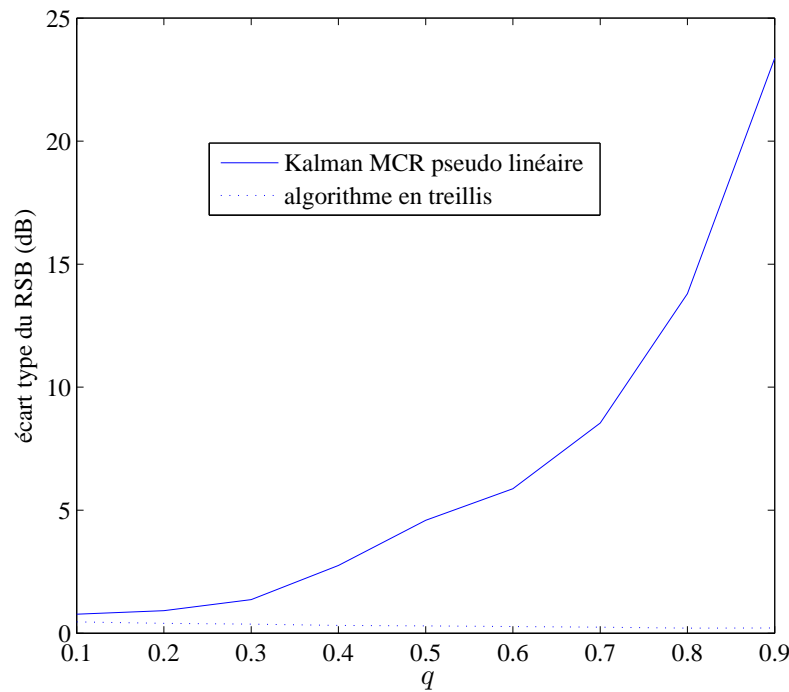


FIG. 4.4 – Écart-type du RSB en fonction de la probabilité de perte d'échantillons pour l'expérience 2.

pour toutes les probabilités de perte d'échantillons. Ceci montre la robustesse de l'algorithme proposé face aux fortes probabilités de perte d'échantillons ainsi qu'aux schémas d'observation du signal. D'autre part, l'écart type du RSB obtenu grâce à l'autre algorithme croît exponentiellement avec la probabilité de perte. On en conclut que la reconstruction du signal utilisant l'algorithme Kalman MCR pseudo-linéaire est fortement dépendante du schémas d'observation pour les probabilités de perte d'échantillons élevées. En effet, un signal de parole n'est pas stationnaire, les paramètres du modèle AR correspondant sont constants pour de faibles durées. Lorsque la probabilité de perte d'échantillons est élevée, l'algorithme Kalman MCR pseudo-linéaire n'arrive pas à suivre les variations des paramètres.

Simulation 3 Dans cette expérience, le signal test utilisé est un signal musical. C'est la toccata en Do min de J. Bach jouée par Glen Gould. Les signaux de musiques sont par essence non stationnaires. De plus, les sons correspondent le plus souvent à des filtres résonnants. Ceci suggère que l'algorithme proposé dans ce chapitre présente de bonnes performances dans la reconstruction de signaux de musique. Ainsi, les mêmes simulations que celles faites à l'expérience 2 sont répétées ici pour le signal musical.

Le tableau 4.2 montre la moyenne et l'écart type du RSB pour des probabilités de perte d'échantillons de 60%, 70%, 80% et 90%. M_1 et M_2 sont les valeurs moyennes du RSB obtenue en utilisant l'algorithme proposé dans ce chapitre ainsi que l'algorithme Kalman MCR pseudo-linéaire. S_1 et S_2 sont les écarts type correspondants.

En se référant au tableau 4.2, on constate que l'algorithme proposé offre, comme attendu, de bonnes performances dans la reconstruction de signaux musicaux. Tout comme dans l'expérience

TAB. 4.2 – Comparaison des deux algorithmes pour l'expérience 3

M_1 et S_1 sont le RSB moyen et l'écart type du RSB obtenus en utilisant l'algorithme proposé dans ce chapitre, M_2 et S_2 sont obtenus en utilisant l'algorithme Kalman MCR pseudo-linéaire.

q	0.6	0.7	0.8	0.9
$M_1 \pm S_1$ (dB)	20 ± 0.16	15 ± 0.14	10.27 ± 0.11	5.4 ± 0.10
$M_2 \pm S_2$ (dB)	16 ± 0.41	11 ± 2.46	-0.18 ± 12.36	-38 ± 31.8

2, M_1 suit une décroissance linéaire lorsque la probabilité de perte q augmente alors que la décroissance de M_2 est exponentielle.

Des tests d'écoute ont montré que les différences entre le signal original et le signal reconstruit en utilisant l'algorithme proposé, ne sont pas audibles en général. Par contre, pour des probabilités de perte élevées, les erreurs reconstruction de ces signaux en utilisant l'algorithme Kalman MCR pseudo-linéaire se traduisent par des impulsions ou même une déformation du son. Ceci est le cas lorsque les paramètres du modèle changent brusquement (début d'une nouvelle note).

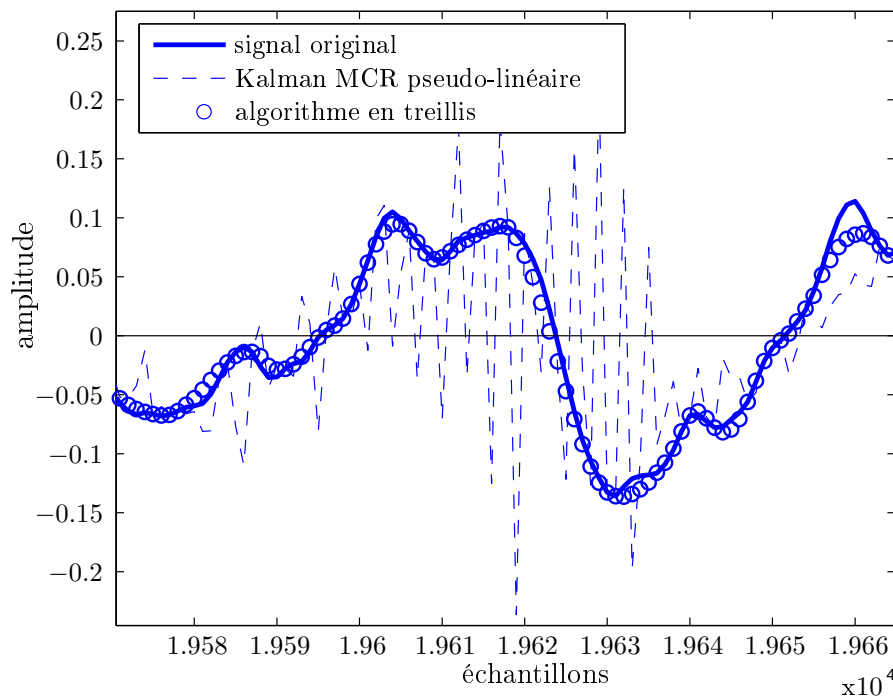


FIG. 4.5 – Reconstruction du signal au début d'une nouvelle note en utilisant les deux algorithmes comparés pour l'expérience 3.

La figure 4.5 montre la reconstruction du signal test en utilisant les deux algorithmes comparés au début d'une nouvelle note. Le début d'un nouvelle note correspond à un saut dans les paramètres AR. Lorsque la probabilité de perte est élevée, l'algorithme Kalman MCR pseudo-linéaire ne s'adapte pas rapidement aux variations des paramètres. Ceci peut se traduire par une phase transitoire d'instabilité caractérisée par de fortes oscillations dans le signal reconstruit, tel qu'on le voit sur la figure 4.5. Par contre, l'algorithme proposé s'adapte rapidement aux variations de paramètres en garantissant la stabilité. Ainsi, le signal reconstruit en utilisant l'algorithme proposé suit correctement le signal original.

4.6 Conclusion

On s'est intéressé dans ce chapitre à l'identification de signaux à échantillons manquants en utilisant la structure en treillis du filtre. Un nouvel algorithme adaptatif permettant une identification stable et une reconstruction robuste d'un processus AR à échantillons manquants est proposé. À notre connaissance, c'est le premier algorithme qui permet un traitement en ligne de signaux à échantillons manquants en utilisant un filtre en treillis. Il utilise un algorithme de Burg adaptatif adapté au cas de signaux à échantillons manquants en utilisant un filtre de Kalman pour la prédiction. La stabilité du filtre ainsi identifié est garantie à chaque instant. En outre, cet algorithme est robuste à de larges probabilités de pertes. De plus, les simulations montrent qu'il offre une convergence rapide et un bon suivi des paramètres même pour des probabilités de perte d'échantillons élevées. Ceci pourrait s'expliquer par la mise à jour des coefficients de réflexion à tous les instants, même lorsqu'un échantillon est manquant. Cependant, ceci est réalisé aux dépens d'un biais d'identification dû à la fonction de coût minimisée par l'algorithme. Cet algorithme a été comparé à l'algorithme Kalman MCR pseudo-linéaire proposé au chapitre 3. Ce dernier utilise une représentation directe du filtre transverse, par conséquent la stabilité du filtre identifié n'est pas garantie. En outre, les simulations montrent que ses performances dépendent fortement du schéma de perte en particulier pour des probabilités de perte élevées. L'application des deux algorithmes à la reconstruction de signaux de parole montre l'avantage de l'algorithme proposé en termes d'erreur de reconstruction et en particulier pour des probabilités de perte d'échantillons élevées. Des tests d'écoute confirment ces résultats. En effet, un signal de parole reconstruit en utilisant l'algorithme proposé reste intelligible même lorsque 85 % des échantillons sont manquants.

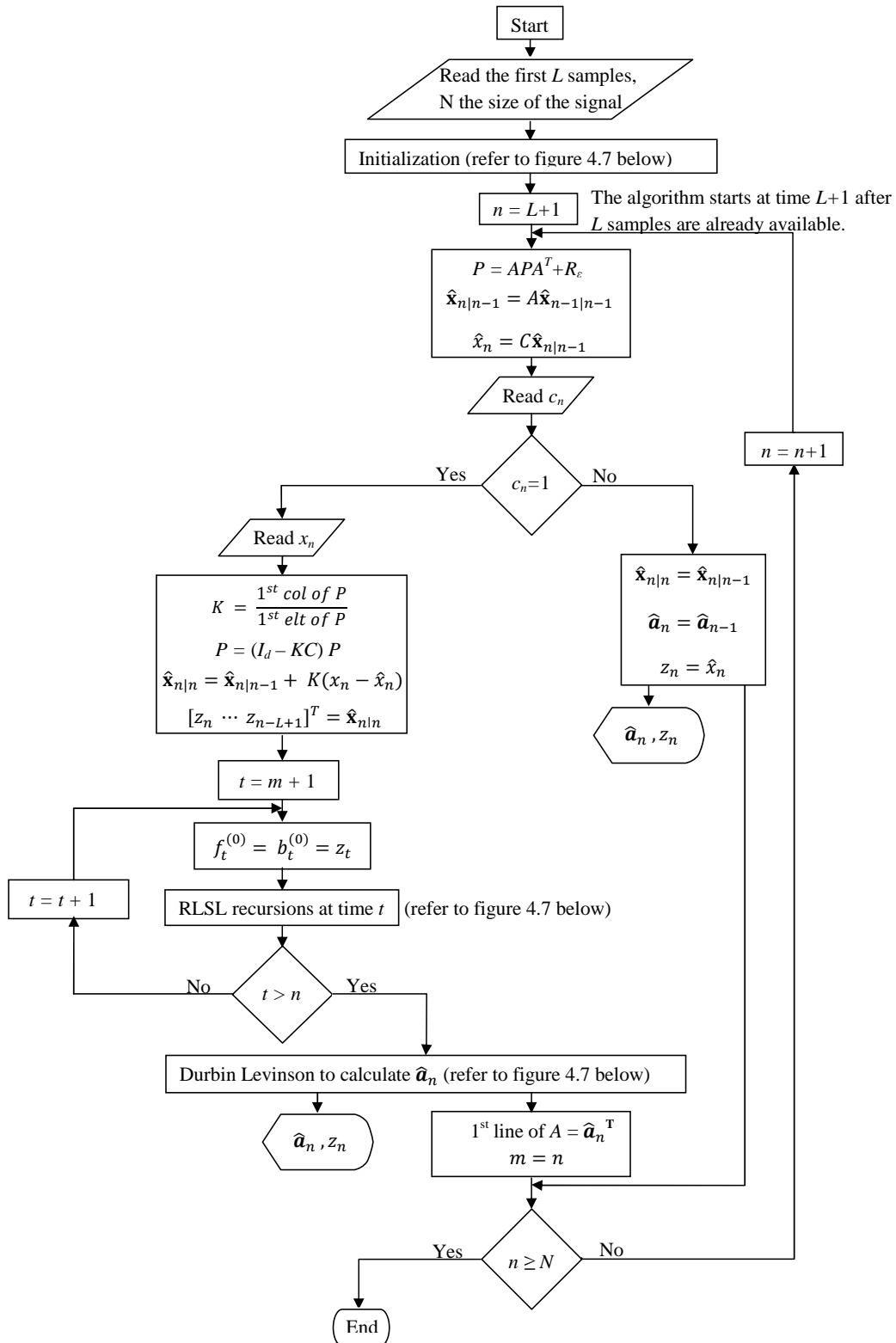


FIG. 4.6 – Organigramme de l'algorithme en treillis pour l'identification et la reconstruction de signaux à échantillons manquants.

Initialization: is at time L since the algorithm starts at time $L+1$.

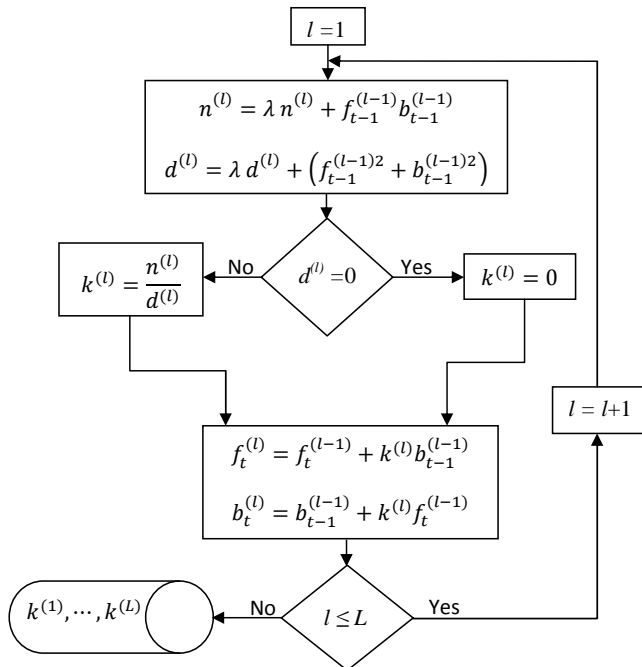
$$\begin{aligned} \hat{\mathbf{x}}_{L|L} &= [x_L \cdots x_1]^T, \\ [z_L \cdots z_1]^T &= \hat{\mathbf{x}}_{L|L}, \\ A &= \begin{bmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}, \\ R_e &= [1 \ 0 \ \dots \ 0]^T [1 \ 0 \ \dots \ 0], \\ P &= 10^3 \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}, \\ [k^{(1)} \ \dots \ k^{(L)}] &= [0 \ \dots \ 0], \\ [b_L^{(0)} \ \dots \ b_L^{(L)}] &= [0 \ \dots \ 0], \\ [f_L^{(0)} \ \dots \ f_L^{(L)}] &= [0 \ \dots \ 0], \\ m &= L, \lambda = 0.99, \\ [d^{(1)} \ \dots \ d^{(L)}] &= [0 \ \dots \ 0], \\ [n^{(1)} \ \dots \ n^{(L)}] &= [0 \ \dots \ 0] \end{aligned}$$

z_n is the signal reconstruction process

m is for each time, the instant of the last previous available sample

$n^{(l)}, d^{(l)}$ are the numerator and the denominator of the reflection coefficient $k^{(l)}$. They are updated at each time recursively.

RLSL recursions at time t :



Durbin Levinson at time n :

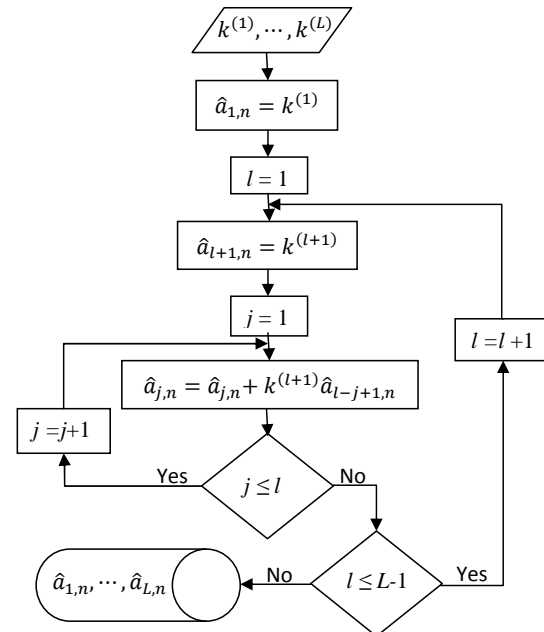


FIG. 4.7 – Organigramme de l'algorithme en treillis pour l'identification et la reconstruction de signaux à échantillons manquants.

Chapitre 5

Echantillonnage adaptatif, systèmes à transmission non uniforme

Nous avons décrit aux chapitres 2, 3 et 4 un ensemble d'algorithmes qui permettent l'identification et la reconstruction adaptatives d'un processus AR à échantillons manquants. Les algorithmes décrits aux chapitres précédents ont été appliqués à la reconstruction de signaux de parole qui ont subi une perte aléatoire d'échantillons. Ils ont montré de bonnes performances même pour un taux élevé d'échantillons manquants. Nous proposons alors de les utiliser dans des systèmes de transmission non uniforme dans un but de compression des signaux. Nous nous intéressons, dans un premier temps, à l'échantillonnage adaptatif et à la transmission non uniforme à partir de signaux codés en MIC (PCM en anglais). Nous décrivons plusieurs méthodes de transmission non uniforme. Ainsi, en combinant les méthodes d'échantillonnage adaptatif et les algorithmes adaptatifs de reconstruction de signaux à échantillons manquants proposés aux chapitres précédents, nous proposons des systèmes de codage/décodage à transmission non uniforme. Nous nous intéressons par la suite à la transmission non uniforme dans un contexte de codage différentiel. Une méthode de transmission non uniforme des erreurs de prédiction est alors proposée. En combinant cette méthode et les méthodes paramétriques de reconstruction des signaux, un système de codage MICDA à transmission non uniforme (MICDA-TNU) est conçu.

Dans ce qui suit, nous présentons le principe de l'échantillonnage adaptatif ainsi que quelques méthodes de transmission non uniforme.

5.1 Echantillonnage adaptatif

On se propose ici de ne transmettre qu'un nombre minimal d'échantillons choisis de telle sorte qu'il soit possible de reconstruire le signal avec une erreur tolérée. Une méthode paramétrique adaptative de prédiction telle que celles proposées aux chapitres précédents est utilisée. Ainsi, à chaque instant n , l'échantillon x_n est prédit grâce à la méthode paramétrique connaissant les échantillons déjà transmis et une estimation actualisée des paramètres du modèle, $\hat{\mathbf{a}}_n$. On note $e_{n,P}$ l'erreur de prédiction de l'échantillon x_n . Une certaine fonction des erreurs de prédiction jusqu'à l'instant n , notée \mathcal{F}_n , est alors comparée à une erreur tolérée, notée \mathcal{E}_n . Si $\mathcal{F}_n < \mathcal{E}_n$, l'échantillon peut donc être prédit avec une erreur tolérée, on évite alors de le transmettre. Le signal résultant est à échantillons manquants, mais la perte des échantillons n'est pas aléatoire. Cette idée peut être vue comme une compression en temps réel avec perte. Ainsi si la méthode paramétrique de prédiction utilisée est efficace, le nombre d'échantillons transmis est considérablement réduit alors que le signal est reconstruit par le récepteur avec une erreur contrôlée.

D'autre part, en augmentant l'erreur de reconstruction tolérée, le nombre d'échantillons transmis est réduit. Dans [55], Mirsaidi *et al.* proposent deux méthodes de transmission fondées sur deux couples différents de valeurs pour les fonctions \mathcal{F}_n et \mathcal{E}_n . Ils s'intéressent aussi à la réduction de l'erreur de quantification par le principe de transmission non uniforme des échantillons [54], ce qui offre un nouveau critère pour le choix des fonctions \mathcal{F}_n et \mathcal{E}_n . Le principe de transmission non uniforme adaptative est aussi utilisé par Asswad *et al.* [5] pour la réduction simultanée de l'erreur de quantification et du débit de transmission des images. Dans ce qui suit, nous présentons plusieurs méthodes de transmission non uniforme adaptative en se basant sur différents choix des fonctions \mathcal{F}_n et \mathcal{E}_n . Pour cela, on distingue deux principaux groupes : les méthodes de transmission adaptative basées sur un seuil de l'erreur de reconstruction et celles basées sur la réduction de l'erreur de quantification.

5.1.1 Critère fondé sur l'erreur de reconstruction

Méthode de transmission 1 Ici, les fonctions \mathcal{F}_n et \mathcal{E}_n sont choisies d'une manière simple, selon [55] :

$$\mathcal{F}_n = g_n |e_{n,P}| \quad (5.1)$$

$$\mathcal{E}_n = c^{te}. \quad (5.2)$$

Ainsi, l'erreur de prédiction instantanée est comparée à une erreur tolérée constante choisie à partir de connaissances *a priori* sur le signal. Le pourcentage d'échantillons transmis ainsi que l'erreur de reconstruction du signal dépendent donc de la valeur de \mathcal{E}_n . Cette dernière permet alors de calculer une borne supérieure de l'erreur quadratique moyenne de reconstruction du signal et par conséquent une borne inférieure du rapport signal sur bruit de reconstruction (RSB). Le coefficient g_n est un facteur de normalisation. Il est choisi selon $g_n = 1/\hat{\sigma}_{x,n}$ où $\hat{\sigma}_{x,n}^2$ est l'estimée de la puissance du signal à l'instant n . Cette estimation étant effectuée à l'émission, tous les échantillons du signal peuvent être utilisés pour estimer la puissance du signal. La normalisation de l'erreur de prédiction $e_{n,P}$ permet de choisir une valeur de \mathcal{E}_n indépendamment de la puissance du signal. Cette méthode est efficace dans le cas de signaux stationnaires, par contre, elle l'est moins dans le cas non stationnaire. Pour cela, Mirsaidi *et al.* [55] proposent une autre méthode plus adaptée au cas de signaux non stationnaires.

Méthode de transmission 2 Dans le cas de signaux non stationnaires, Mirsaidi *et al.* [55] proposent de faire évoluer la valeur de \mathcal{E}_n en fonction des propriétés statistiques du signal en question. Ils utilisent l'idée que les performances de la prédiction sont meilleures lorsque les échantillons consécutifs sont fortement corrélés. Par conséquent, lorsque les échantillons sont fortement corrélés, ils sont moins fréquemment transmis. Ainsi, à chaque instant, les coefficients d'autocorrélation jusqu'à l'ordre L du modèle sont estimés de manière récursive au niveau de l'émetteur. Ils définissent alors les termes \mathcal{F}_n et \mathcal{E}_n selon :

$$\mathcal{F}_n = \frac{1}{\hat{\sigma}_{x,n}} e_{n,P} \quad (5.3)$$

$$\mathcal{E}_n = \rho \left(1 - \frac{\hat{\mathbf{r}}_{x,n}^\top \hat{\mathbf{a}}_n}{\hat{\sigma}_{x,n}^2} \right). \quad (5.4)$$

où $\hat{\mathbf{r}}_{x,n}$ représente l'estimée du vecteur $\mathbf{r}_x = [r_1, \dots, r_L]^\top$ à l'instant n , $\hat{\mathbf{a}}_n$ est l'estimée des paramètres à l'instant n obtenue grâce à un algorithme adaptatif d'identification et de reconstruction de signaux à échantillons manquants, le coefficient ρ permet de contrôler le pourcentage d'échantillons transmis.

Méthode de transmission 3 Nous proposons de transmettre un nombre minimal d'échantillons tout en garantissant un rapport signal sur bruit de reconstruction supérieur à un certain seuil que l'on fixe. Pour cela, on propose de vérifier à chaque instant que le rapport signal sur bruit de reconstruction au niveau du récepteur est supérieur au seuil fixé. Ainsi à chaque instant n , l'échantillon est prédit en utilisant l'une des méthodes décrites aux chapitres précédents connaissant les échantillons déjà transmis et une estimation des paramètres du modèle à l'instant n . Ayant $e_{n,P}$ et x_n , le rapport signal sur bruit de reconstruction à l'instant n est alors estimé en dB par :

$$\mathcal{F}_n = 10 \log \left(\frac{\hat{\sigma}_{x,n}^2}{\text{EQMR}_n} \right) = 10 \log \left(\frac{n}{\sum_{i=1}^n e_{i,P}^2} \hat{\sigma}_{x,n}^2 \right). \quad (5.5)$$

où $\text{EQMR}_n = \frac{1}{n} \sum_{i=1}^n e_{i,P}^2$ est l'erreur quadratique moyenne de reconstruction à l'instant n avec $e_{i,P} = 0$ si l'échantillon x_i est transmis. Il est à noter que cette estimation est calculée récursivement sans avoir à garder en mémoire toutes les erreurs de prédictions précédentes. Ceci est obtenu en calculant de manière récursive séparément l'erreur quadratique moyenne de reconstruction (EQMR_n) et la puissance du signal. L'estimée du rapport signal sur bruit ainsi obtenue est comparée au seuil fixé \mathcal{E}_n . Si $\mathcal{F}_n \geq \mathcal{E}_n$, l'échantillon n'est pas transmis, il sera prédit auprès du récepteur avec une erreur tolérée. Dans le cas contraire, l'échantillon est transmis, EQMR_n devrait alors être corrigée en remplaçant $e_{n,P}$ par zéro. Il est à noter qu'on n'a pas tenu compte ici de l'erreur de quantification lorsqu'un échantillon est transmis. On note $x_{n,Q}$ la valeur quantifiée de l'échantillon x_n et $e_{n,Q} = x_n - x_{n,Q}$ l'erreur de quantification correspondante. Dans ce cas, $\text{EQMR}_n = \frac{1}{n} \sum_{i=1}^n e_{i,r}^2$ où $e_{i,r}$ est l'erreur de reconstruction à l'instant i donnée par,

$$e_{i,r} = \begin{cases} e_{i,Q} & \text{si l'échantillon est transmis,} \\ e_{i,P} & \text{sinon.} \end{cases} \quad (5.6)$$

La méthode ainsi décrite permet de diminuer le débit de transmission tout en garantissant un rapport signal sur bruit de reconstruction toléré. Nous proposons de vérifier à chaque instant le rapport signal sur bruit toléré, ceci permet d'éviter qu'une certaine tranche du signal soit entièrement transmise et bien reconstruite aux dépens d'une autre qui serait alors mal reconstruite.

5.1.2 Critère fondé sur l'erreur de quantification

Dans un système de transmission numérique simple tel que celui représenté à la figure 5.1.2, en supposant que l'erreur de transmission est nulle ($\hat{x}_{n,Q} = x_{n,Q}$), l'erreur de reconstruction du signal est donc égale à l'erreur de quantification $e_{n,Q}$. Mirsaidi *et al.* [54] s'intéressent à la réduction de l'erreur de quantification par transmission non uniforme adaptative du signal. Grâce à cette technique, l'erreur de reconstruction du signal ainsi que le débit de transmission sont réduits en comparaison à la transmission uniforme des échantillons quantifiés. Avant de présenter les méthodes proposées dans ce but, un rappel sur la quantification est présenté dans ce qui suit.

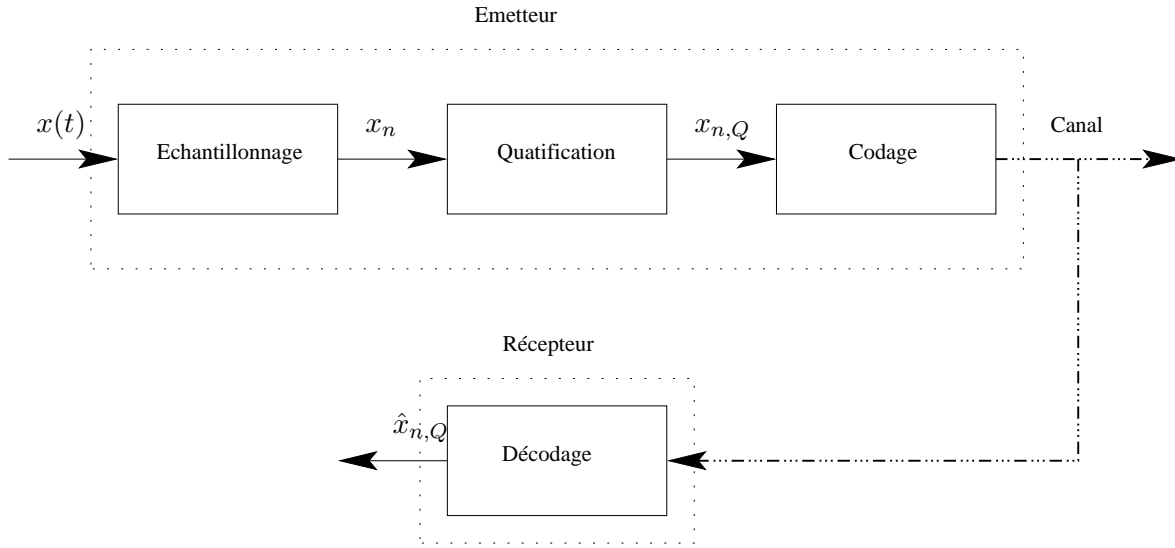


FIG. 5.1 – Schéma simplifié d'un système de transmission numérique.

Rappels sur la quantification

Une étape fondamentale dans la représentation numérique d'un signal analogique est la quantification. Cette étape suit l'étape d'échantillonnage. La quantification est une fonction non linéaire non injective. Elle associe au signal analogique (qui peut prendre une infinité de valeurs différentes) un nombre fini N_Q de valeurs. Ainsi, soit $x_{max} = \max_n |x_n|$ la valeur maximale en valeur absolue que peut prendre le signal. Cette valeur est supposée en pratique finie. L'intervalle $[-x_{max}, x_{max}]$, contenant une infinité de valeur, est alors subdivisé en N_Q intervalles juxtaposés. A chacun de ces N_Q intervalles est assignée une valeur qui peut être par exemple la borne inférieure de l'intervalle ou la valeur médiane de celui-ci.

La quantification remplace la valeur exacte du signal par une valeur approchée. Il y a donc une dégradation irréversible de l'information. On parle alors de bruit de quantification dont la valeur à chaque instant est égale à l'erreur de quantification $e_{n,Q}$.

Les différents intervalles et niveaux de quantifications peuvent être choisis de différentes façons en fonction de l'application envisagée. Lorsque les N_Q intervalles sont d'amplitudes égales, les différents niveaux de quantification sont donc équidistants (uniformément répartis), la quantification est dite uniforme. Deux niveaux de quantification consécutifs sont alors séparés d'un pas de quantification δ . Lorsque la quantification est uniforme celui-ci est donné par :

$$\delta = \frac{2x_{max}}{N_Q}. \quad (5.7)$$

L'amplitude crête à crête du bruit de quantification est égale à δ ($-\frac{\delta}{2} \leq e_{n,Q} \leq \frac{\delta}{2}$) et sa puissance est de

$$E \{e_{n,Q}^2\} = \frac{\delta^2}{12}. \quad (5.8)$$

Ce bruit est d'autant plus faible que le pas de quantification δ est faible et par conséquent le nombre de niveaux de quantification N_Q est élevé.

Les différents niveaux de quantification sont codés par des symboles binaires. En utilisant des mots binaires de B bits, on arrive à coder $N_Q = 2^B$ valeurs différentes. Le débit binaire est alors

donné par $d = BF_e$ où F_e est la fréquence d'échantillonnage. Ainsi, en augmentant le nombre de bits nécessaire pour coder un échantillon, l'erreur de quantification est réduite par contre ceci se fait au détriment d'une augmentation du débit binaire. En pratique, on essaye de maintenir un compromis entre un débit faible et une qualité minimale requise pour le signal reconstruit. Ainsi, en général 11 bits par échantillons sont requis pour une bonne représentation d'un signal de parole quantifié uniformément [59, 67].

Or, ce n'est pas avec la quantification uniforme que l'on obtient la meilleure caractéristique quand la densité de probabilité de l'amplitude du signal n'est pas uniforme tel qu'il est le cas des signaux sonores [67]. Dans ce cas, on peut obtenir de meilleures performances en utilisant une quantification non uniforme logarithmique de sorte que le pas de quantification augmente au fur et à mesure que l'amplitude du signal est importante. Une quantification logarithmique permet de maintenir un quotient erreur de quantification/niveau de signal sensiblement constant [59]. Une telle quantification est assimilée en pratique à une compression logarithmique du signal suivie d'une quantification uniforme. Pour le codage de la parole, deux lois de compression ont fait l'objet de normalisations mondiales. La loi μ est utilisée aux Etats-Unis, au Canada et au Japon, alors que la loi A est utilisée dans les systèmes européens et le reste du monde [59]. Nous rappelons ici l'expression de la loi de μ , celle-ci sera utilisée dans la partie consacrée aux simulations.

$$F(x) = \text{sign}(x) \frac{\ln(1 + \mu |x|)}{1 + \mu} \quad (5.9)$$

où $\mu = 255$ et $x \in [-1; 1]$.

Réduction de l'erreur de quantification par transmission adaptative

L'idée proposée par Mirsaidi *et al.* pour réduire l'erreur de quantification se résume comme suit : Supposons qu'il soit possible de prédire l'échantillon à l'instant n d'une manière efficace de sorte que l'erreur de prédiction soit inférieure à celle de quantification ($|e_{n,P}| < |e_{n,Q}|$). Dans ce cas, il est préférable de remplacer au niveau du récepteur l'échantillon par sa prédiction et d'éviter par conséquent sa transmission. Lorsque la méthode paramétrique de prédiction utilisée est efficace, un grand nombre d'échantillons ne sera pas transmis. Ainsi, l'erreur totale de reconstruction sera inférieure à celle obtenue pour une transmission uniforme du signal quantifié. De plus, le débit de transmission (ou le volume de données nécessaire) est réduit en raison de la transmission non uniforme. Les fonctions \mathcal{F}_n et \mathcal{E}_n de la méthode à transmission non uniforme ainsi décrite sont alors définies par :

$$\mathcal{F}_n = |e_{n,P}|, \quad (5.10)$$

$$\mathcal{E}_n = |e_{n,Q}|. \quad (5.11)$$

Or, le récepteur ne dispose que d'échantillons quantifiés, la prédiction des échantillons non transmis se fait donc à partir de ceux-ci. Les performances de la prédiction peuvent donc être mauvaises à cause de l'erreur de quantification. Pour pallier ce problème, considérons les deux systèmes suivants :

- Système 1 : la transmission est périodique (tous les échantillons quantifiés sont transmis). Le nombre de bits utilisés pour coder un échantillon est B_1 . Le débit de transmission est donc donné par $d_1 = F_e B_1$ où F_e est la fréquence d'échantillonnage.
- Système 2 : la transmission est non périodique, les échantillons manquants sont alors remplacés par leurs prédictions. Le nombre de bits utilisés pour coder un échantillon transmis est B_2 . Dans ce cas, la transmission étant non uniforme, on considère le débit de transmission moyen donné par $\bar{d}_2 = p F_e B_2$ où p désigne le pourcentage des échantillons transmis.

Il est clair que pour obtenir de meilleures performances de reconstruction dans le deuxième système, on choisit $B_2 > B_1$. Ainsi, si la méthode de prédiction utilisée est performante, un nombre considérable d'échantillons ne sera pas transmis. Si le pourcentage d'échantillons transmis est suffisamment faible de sorte que :

$$p < \frac{B_1}{B_2} \quad (5.12)$$

le second système offre un débit de transmission moyen inférieur à celui obtenu par le premier système ($\bar{d}_2 < d_1$). Ainsi, grâce à la transmission non uniforme, si une méthode de prédiction performante est utilisée, il est possible d'améliorer le rapport signal sur bruit de reconstruction tout en diminuant le débit de transmission.

Dans ce qui suit, un système de transmission numérique utilisant les méthodes de transmission non-uniforme décrites ci-dessus est proposé. Ce système utilise pour la reconstruction du signal l'une des méthodes d'identification et de reconstruction de signaux à échantillons manquants proposées aux chapitres précédents.

5.2 Système à transmission non uniforme

Nous proposons ici un algorithme de compression fondé sur le principe de transmission non uniforme. Il utilise une méthode paramétrique adaptative pour la prédiction de signaux à échantillons manquants. Les méthodes paramétriques décrites aux chapitres 2, 3 et 4 sont bien adaptées à cette fin. Ainsi le signal transmis est caractérisé par un processus AR non stationnaire. L'objectif principal est d'obtenir de meilleures performances en termes de RSB et de débit de transmission que celles obtenues par les méthodes classiques de transmission périodique.

Ainsi, à l'émission, un algorithme adaptatif permettant de choisir les instants appropriés de transmission est utilisé. Il permet donc d'éviter de transmettre un échantillon si celui-ci peut être prédit à la réception avec une erreur tolérée selon un critère fixé à l'avance. Le rôle du récepteur est donc de reconstruire en temps réel un signal à échantillons manquants en utilisant une méthode paramétrique. Un algorithme d'identification et de reconstruction en temps réel de processus AR non stationnaires à échantillons manquants (Cf. les chapitres 2, 3 et 4) est alors utilisé.

Ainsi, à l'instant n , connaissant tous les échantillons déjà transmis, l'émetteur utilise le même algorithme de reconstruction utilisé par le récepteur pour évaluer la prédiction de l'échantillon x_n par ce dernier. Ayant l'erreur de prédiction $e_{n,P}$, la fonction \mathcal{F}_n des erreurs de prédiction du récepteur jusqu'à l'instant n est mise à jour récursivement. Celle-ci est comparée à une limite spécifique tolérable \mathcal{E}_n . Tel qu'il a été mentionné ci-dessus, le choix de ces deux fonctions dépend de l'objectif et des performances attendues. Ainsi si le but est de réduire le débit de transmission tout en conservant un rapport signal sur bruit de reconstruction minimal donné, les fonctions \mathcal{F}_n et \mathcal{E}_n sont définies par la méthode de transmission 3 (Cf. le paragraphe 5.1.1). Par ailleurs, pour minimiser l'erreur de quantification, ces fonctions sont choisies à chaque instant selon $\mathcal{F}_n = |e_{n,P}|$ et $\mathcal{E}_n = |e_{n,Q}|$. On rappelle que le système de transmission non uniforme peut utiliser un nombre de bits par code B_2 différent (en général supérieur) de B_1 , celui utilisé par le système de transmission uniforme auquel il est comparé. Les erreurs de quantification respectives sont à chaque instant $e_{n,Q1}$ et $e_{n,Q2}$. Si l'objectif est de réduire l'erreur de reconstruction du système à transmission uniforme (due à la quantification), on choisit $\mathcal{E}_n = e_{n,Q1}$. Dans ce cas on suppose bien évidemment que la quantification a lieu à l'émission ainsi la vraie valeur du signal est connue. Ceci permet de pouvoir quantifier en utilisant des nombres différents de bits par code et de pouvoir calculer dans chaque cas l'erreur de quantification correspondante.

Il est à noter que dans le système à transmission non uniforme, un bit indicateur (drapeau) de la transmission ou non d'un échantillon est transmis. Pour des signaux localement stationnaires et lorsque la méthode de reconstruction utilisée est performante, il est possible que plusieurs échantillons consécutifs ne soient pas transmis. Ceci est particulièrement le cas des silences dans un signal sonore (de musique ou de parole) tel qu'on le constatera dans la partie consacrée aux simulations. Dans ce cas, un seul bit nul est transmis correspondant au premier échantillon manquant. Ainsi, au cours des instants suivants, le récepteur considère les échantillons manquants (ils sont alors remplacés par leurs prédictions) jusqu'à ce qu'un drapeau 1 soit reçu. Ceci permet d'éviter la transmission inutile de plusieurs drapeaux nuls consécutifs. Il en est de même pour le cas de plusieurs échantillons consécutifs transmis. Ainsi, le nombre de bits transmis par le système à transmission non uniforme en utilisant ce principe de codage est de :

$$B_{tot} = pNB_2 + N_d \quad (5.13)$$

où B_{tot} est le nombre total de bits transmis, N_d le nombre de bits drapeaux transmis et N la taille totale du signal. Le débit de transmission moyen est alors obtenu par :

$$d = \left(pB_2 + \frac{N_d}{N} \right) F_e, \quad (5.14)$$

Ainsi, pour avoir un meilleur débit que celui obtenu par un système de transmission uniforme (codage PCM par exemple) qui utilise B_1 bits/échantillon, le système de transmission non uniforme devrait vérifier :

$$pB_2 + \frac{N_d}{N} < B_1. \quad (5.15)$$

L'algorithme de codage/décodage décrit ci-dessus se résume à l'instant n aux étapes suivantes :

Au niveau de l'émetteur

- L'échantillon x_n est quantifié, si nécessaire en utilisant deux nombres de bits différents, les deux versions quantifiées sont notées $x_{n,Q1}$ et $x_{n,Q2}$ respectivement.
- L'échantillon x_n est prédit en fonction de l'estimée des paramètres à l'instant précédent $\hat{\mathbf{a}}_{n-1}$ et des versions quantifiées des échantillons déjà transmis (l'algorithme évalue la performance du récepteur à la prédiction de cet échantillon). La prédiction de l'échantillon est notée $\hat{x}_{n|n-1}$. Il est à noter ici que toutes les étapes de l'algorithme de reconstruction nécessaires à la prédiction de l'échantillon sont aussi exécutées.
- Ayant la vraie valeur de l'échantillon, l'erreur de prédiction $e_{n,P}$ ainsi que les erreurs de quantification $e_{n,Q1}$ et $e_{n,Q2}$ de celui-ci sont calculées, ces valeurs sont utilisées pour calculer les fonctions \mathcal{F}_n et \mathcal{E}_n . Dans le cas de la méthode de transmission 3 (Cf. paragraphe 5.1.1), EQMR_n et $\hat{\sigma}_{x,n}$ sont calculées récursivement en utilisant respectivement $e_{n,P}$ et x_n . Tandis que pour la méthode de transmission permettant de réduire l'erreur de quantification (Cf. paragraphe 5.1), $\mathcal{F}_n = e_{n,P}$ et $\mathcal{E}_n = e_{n,Q1}$.
- Comparer \mathcal{F}_n et \mathcal{E}_n (La transmission de l'échantillon x_n est-elle requise ?) :
 1. Si $\mathcal{F}_n < \mathcal{E}_n$, l'échantillon n'est pas transmis. Par conséquent les étapes de la méthode paramétrique de reconstruction dans le cas où l'échantillon est manquant sont exécutées (Cf. chapitres 2, 3 et 4). En particulier, l'estimée des paramètres du modèle AR n'est pas mise à jour, $\hat{\mathbf{a}}_n = \hat{\mathbf{a}}_{n-1}$. Si l'échantillon précédent est transmis, un bit de valeur 0 est transmis.

2. Si $\mathcal{F}_n > \mathcal{E}_n$, l'échantillon doit être transmis. Dans le cas où deux quantifications différentes de l'échantillon sont utilisées, la meilleure quantification $x_{n,Q2}$ est transmise. Les étapes de l'algorithme de reconstruction lorsqu'un échantillon est reçu sont exécutées. Notamment, les paramètres du modèle AR sont mis à jour. Il est à noter que la valeur quantifiée de l'échantillon et non sa vraie valeur est utilisée pour l'estimation des paramètres ou la prédiction des échantillons aux instants suivants. Ceci est dû au fait que l'émetteur évalue la prédiction effectuée au niveau du récepteur qui ne dispose que des échantillons quantifiés. Ainsi, la mise à jour des paramètres grâce aux algorithmes du gradient et MCR (Cf. chapitres 2 et 3) s'écrit sous la forme suivante :

$$\hat{\mathbf{a}}_n = \hat{\mathbf{a}}_{n-1} + \text{Gain}(x_{n,Q2} - \hat{x}). \quad (5.16)$$

Par ailleurs, si l'algorithme d'identification utilise la structure en treillis du filtre (Cf. chapitre 4), il est initialisé à l'instant n par l'équation suivante :

$$\hat{f}_n^{(0)} = \hat{b}_n^{(0)} = x_{n,Q2}. \quad (5.17)$$

Dans le cas de la méthode de transmission 3, la valeur de EMQR_n devrait être corrigée en remplaçant l'erreur de prédiction $e_{n,P}$ par l'erreur de quantification de l'échantillon transmis $e_{n,Q2}$. Si l'échantillon précédent est manquant, un bit indicateur de valeur 1 est transmis.

Au niveau du récepteur Une méthode paramétrique adaptative d'identification et de reconstruction de signaux à échantillons manquants est utilisée (la même doit être utilisée par l'émetteur). Différentes méthodes ont été décrites aux chapitres 2, 3 et 4, on rappelle ici brièvement leurs principales étapes.

- Si un échantillon est reçu, les paramètres du modèle AR sont mis à jour. Le signal reconstruit prend alors à l'instant n la valeur de l'échantillon reçu.
- Si l'échantillon est manquant, il est prédit en fonction des échantillons disponibles précédents et de l'estimée actuelle des paramètres. Sa prédiction est utilisée pour reconstruire le signal. Les paramètres du modèle AR sont conservés.

Le système à transmission non uniforme ainsi décrit est représenté sur la figure 5.2. On s'intéresse dans un premier temps à l'amélioration du MIC (ou en anglais PCM) puis du MICDA (ou en anglais ADPCM) grâce à l'utilisation de la transmission adaptative non uniforme. Ainsi, dans ce qui suit, les systèmes de transmission non uniforme décrits ci-dessus sont comparés par simulations aux codeurs à transmission uniforme. Une nette amélioration en termes de RSB et de débit de transmission moyen simultanément est mise en valeur.

5.3 Simulations

Le système de transmission non uniforme décrit ci-dessus est comparé ici à la transmission uniforme de tous les échantillons quantifiés (MIC) par des simulations sur des signaux de parole. On s'intéresse particulièrement à la méthode de transmission 3 et à celle permettant l'amélioration de l'erreur de quantification. Dans ce qui suit, nous appelons méthode MIC, méthode A et méthode B, respectivement, le système de transmission uniforme, le système de transmission non uniforme utilisant la méthode de transmission 3 et celui permettant de réduire l'erreur de quantification. Le signal de parole test utilisé est la phrase usuelle '*Mary had a little lamb, its fleece was white as snow*' enregistrée par la voix d'un homme. Le signal est échantillonné à la

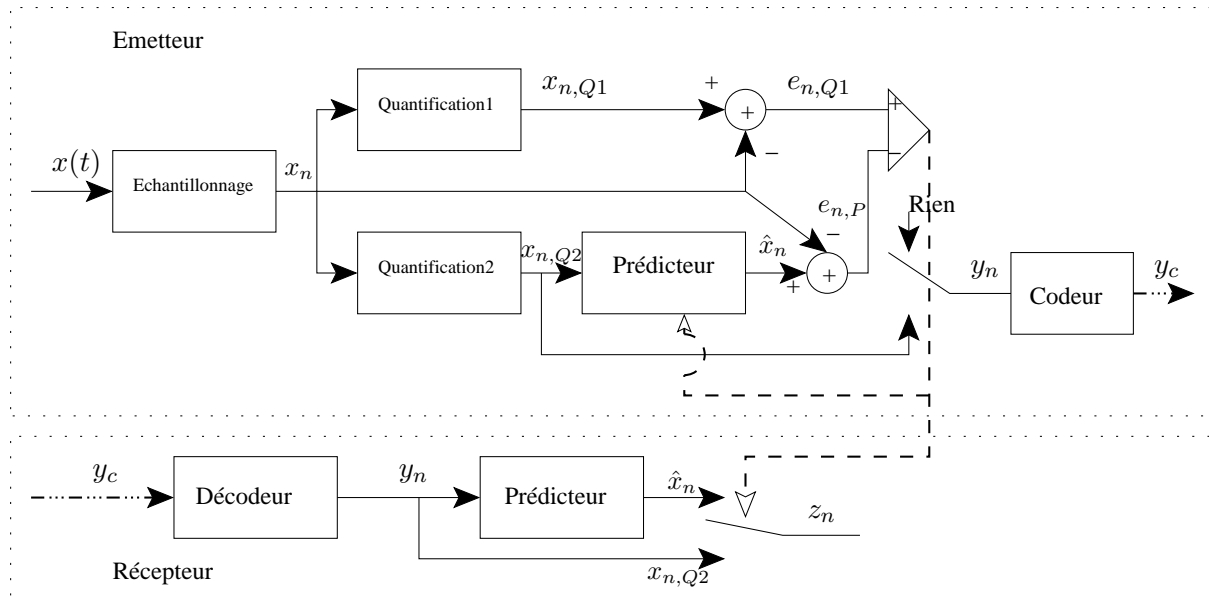


FIG. 5.2 – Schéma simplifié d'un système de transmission numérique.

fréquence $F_e = 8$ kHz. La figure 5.3 montre l'évolution du RSB en dB en fonction du débit de transmission pour un codage MIC du signal test en distinguant le cas de la quantification uniforme du signal et celle suivant la loi de μ . Cette figure servira de référence dans les simulations.

Comme il se doit, le RSB croît avec le débit de transmission (l'erreur de quantification est plus faible en utilisant un plus grand nombre de niveaux de quantification). De plus, on note l'intérêt d'une quantification suivant une loi de μ de la parole, elle offre un meilleur RSB que celui obtenu par une quantification uniforme pour un même débit de transmission. Pour cela, nous nous intéressons dans les simulations au cas de quantification suivant une loi de μ . Ainsi, dans ce qui suit, les méthodes MIC, A et B sont comparées par trois séries de simulations. Leurs performances sont mesurées en termes de RSB (dB) et de débit de transmission moyen. Pour la méthode MIC, le débit de transmission est donné par $d_1 = B_1 F_e$, alors que pour les méthodes A et B, il est donné par l'équation 5.14. La première série permet de comparer les méthodes MIC et A. Grâce à ces simulations, on montre qu'il est possible d'obtenir, en utilisant la méthode A, un RSB donné pour un débit de transmission moyen inférieur à celui requis par la méthode MIC. Grâce à la deuxième série de simulations, on montre qu'il est possible, en utilisant la méthode B, d'améliorer simultanément l'erreur de reconstruction et le débit de transmission obtenus par la méthode MIC. Finalement, les méthodes A et B sont comparées par une troisième série de simulations. On rappelle ici que les méthodes A et B utilisent des algorithmes adaptatifs d'identification et de reconstruction de signaux à échantillons manquants. Les algorithmes Kalman MCR pseudo-linéaire (décrit au chapitre 3) et celui en treillis (décrit au chapitre 4) étant bien adaptés à la reconstruction adaptative de signaux de parole, on propose de les utiliser dans ces méthodes. Dans ce qui suit, on appelle méthode A-KMCR, A-KMCRT, B-KMCR et B-KMCRT, les méthodes A et B utilisant respectivement les algorithmes Kalman MCR pseudo-linéaire et celui en treillis. Pour les différentes simulations suivantes, le facteur d'oubli utilisé dans les algorithmes KMCR et KMCRT est $\lambda = 0.99$. Ce choix est empirique, il est justifié aux chapitres 3 et 4. De plus, la parole est caractérisée par un processus AR non stationnaire d'ordre 12.

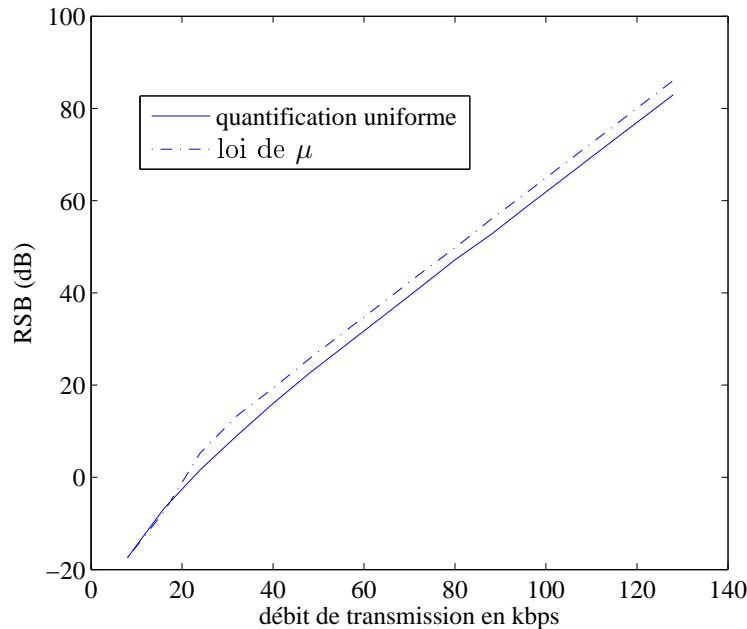


FIG. 5.3 – Evolution du RSB(dB) en fonction du débit de transmission pour un codage MIC du signal test.

Simulations 1 Les méthodes A-KMCR et A-KMCRT sont comparées ici à la méthode MIC. Il convient alors de rappeler que la méthode A (décrite au paragraphe 5.1.1) est une méthode de transmission non uniforme qui permet de garantir un rapport signal sur bruit de reconstruction RSB donné. Ainsi, pour un RSB fixé *a priori* (et bien évidemment pour un signal particulier), le débit de transmission moyen obtenu en utilisant les méthodes A-KMCR et A-KMCRT dépend du nombre de bits B_2 utilisé pour coder les échantillons transmis. Inversement, la méthode MIC fournit un RSB pour un débit de transmission (un nombre de bits par échantillon B_1) fixé *a priori*. Ainsi, en se référant à la figure 5.3, pour un débit de transmission $d_1 = 40$ kbps (correspondant à $B_1 = 5$ bits/échantillon), la méthode MIC offre un RSB de 19.33 dB. Dans le but de comparer les deux méthodes A et MIC, le RSB de la méthode A est fixé à 19.33 dB. Les deux méthodes sont alors comparées en termes du débit de transmission moyen. Il reste alors à choisir le nombre de bits B_2 utilisé par la méthode A pour coder un échantillon émis. Ainsi, la figure 5.4 montre le débit de transmission moyen obtenu par les méthodes A-KMCR et A-KMCRT en fonction du nombre de bits B_2 par échantillon émis pour un RSB = 19.33 dB.

Cette figure montre que pour un RSB = 19.33 dB, le plus faible débit de transmission moyen de chacune des méthodes A-KMCR et A-KMCRT est obtenu pour $B_2 = B_1 + 1 = 6$ bits. Pour ce choix, les débits de transmission moyens obtenus grâce aux méthodes A-KMCR et A-KMCRT sont respectivement de $d_2 = 26.14$ et $d_2 = 25.03$ kbps. Un gain de 14 kbps est alors obtenu par rapport à la méthode MIC ($d_1 = 40$ kbps). Pour $B_2 > 6$, le débit est croissant. Néanmoins, les simulations montrent que pour ces valeurs de B_2 , le pourcentage d'échantillons transmis est plus faible que celui obtenu pour $B_2 = 6$. Ceci s'explique de la façon suivante. L'erreur de quantification des échantillons transmis étant plus faible, les algorithmes de reconstruction peuvent fournir une meilleure prédiction entraînant donc une diminution du pourcentage p des échantillons transmis. Par contre, la diminution de p n'est pas suffisante pour compenser l'augmentation du nombre de bits B_2 par échantillon transmis et donc pour avoir un meilleur débit que celui obtenu pour

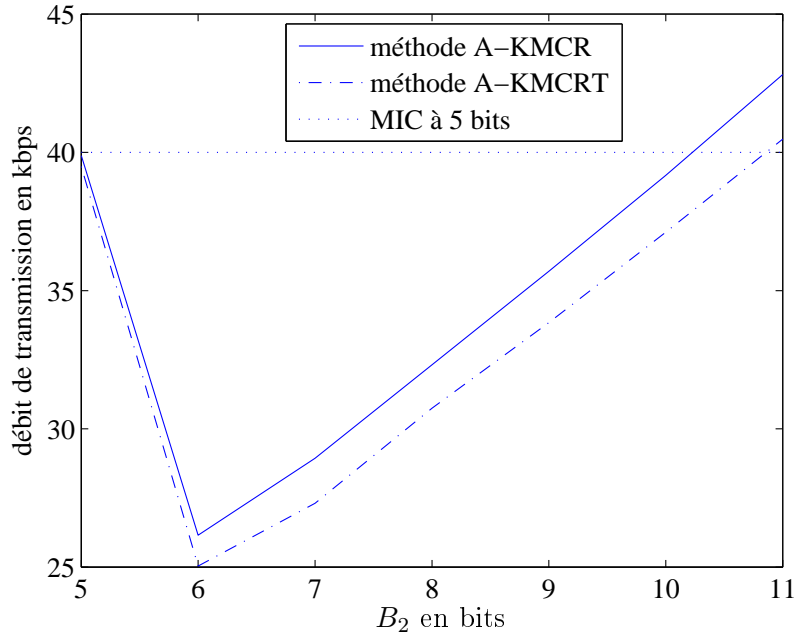


FIG. 5.4 – Débit de transmission moyen obtenu en utilisant la méthode A en fonction du nombre de bits B_2 pour $\text{RSB} = 19.33$ dB.

$B_2 = 6$. Le débit ainsi obtenu reste toutes fois inférieur à 40 jusqu'à $B_2 = 10$. A partir de cette valeur, $pB_2 + \frac{N_d}{N} > B_1$, par conséquent un débit supérieur à 40 est obtenu.

D'autre part, pour $B_2 < 5$, le RSB obtenu est inférieur à 19.33 dB. Pour ces valeurs du nombre de bits, l'erreur de quantification est élevée (correspondant à un $\text{RSB} < 19.33$ dB), par conséquent, l'erreur de prédiction l'est aussi. Ainsi, lorsque l'erreur de prédiction est élevée, l'échantillon est transmis par la méthode entraînant un $\text{RSB} < 19.33$ dB.

Ce même test est répété pour différentes valeurs du RSB. A chaque fois, le meilleur débit de transmission moyen des méthodes A-KMCR et A-KMCRT est obtenu pour $B_2 = B_1 + 1$ bits par échantillons transmis, où B_1 est le nombre de bits par échantillon pour lequel la méthode MIC fournit le même RSB. Ainsi, en se référant à la figure 5.3, pour $B_1 = 7$ ($d_1 = 56$ kbps), la méthode MIC fournit un $\text{RSB} = 31.78$ dB. Pour ce même RSB, le meilleur débit de transmission moyen de chacune des méthodes A-KMCR et A-KMCRT est obtenu pour $B_2 = 8$, ils sont respectivement de $d_2 = 48.58$ kbps et $d_2 = 48.38$ kbps. Un gain de 8 kbps est alors obtenu par rapport à la méthode MIC. Ainsi, ces tests ont permis de tracer la figure 5.5. Cette figure montre, pour le signal test quantifié suivant la loi de μ , le RSB obtenu en fonction du débit de transmission pour les méthodes A-KMCR, A-KMCRT et MIC.

Cette figure montre l'intérêt des méthodes A-KMCR et A-KMCRT à la réduction du débit de transmission. Ainsi, pour un RSB donné, le débit de transmission moyen obtenu en utilisant les méthodes A-KMCR et A-KMCRT est inférieur à celui requis par la méthode MIC pour fournir le même RSB. Réciproquement, pour un débit de transmission donné, il est possible d'obtenir grâce aux méthodes A-KMCR et A-KMCRT un meilleur RSB que celui fournis par la méthode MIC. Ce gain est d'autant plus important que le RSB (réciproquement le débit) est plus faible. Les deux méthodes de transmission non uniforme montrent des performances voisines, les performances de la méthode A-KMCRT dépassent néanmoins celles de la méthode A-KMCR pour les faibles valeurs du RSB. Ceci montre la robustesse aux faibles RSB des méthodes utilisant la structure

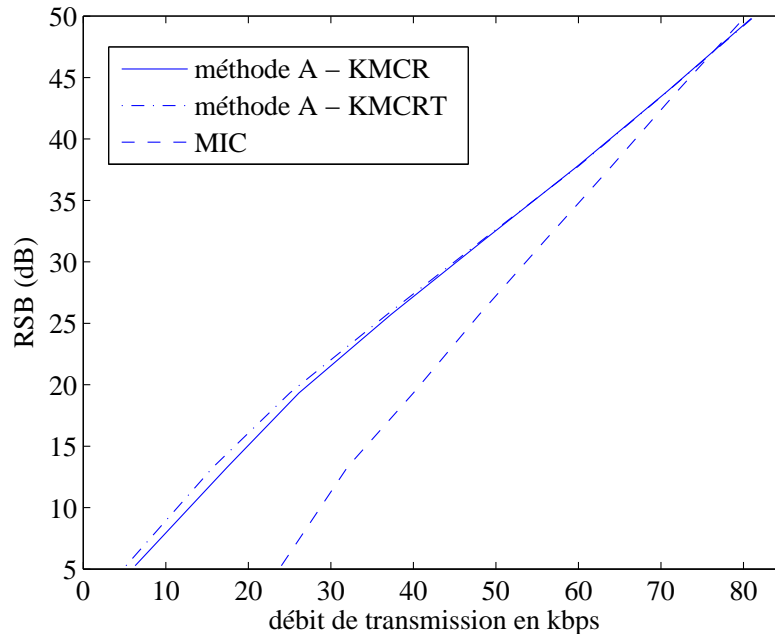


FIG. 5.5 – Comparaison des méthodes MIC, A-KMCR et A-KMCRT.

en treillis du filtre. Il est à noter ici que les méthodes A-KMCR et A-KMCRT ont même montré parfois de meilleures performances que celles obtenues par le codage ADPCM (donné par le standard G.726 [20]). Ainsi, par exemple, pour un débit de 40 kbps, le codage ADPCM fournit un RSB = 23.59 dB. Ce même RSB est obtenu grâce aux méthodes A-KMCR et A-KMCRT pour des débits de 34 et 34.5 kbps respectivement.

Simulations 2 Les méthodes B-KMCR, B-KMCRT et MIC sont comparées ici. Nous rappelons que la méthode B permet grâce à la transmission non uniforme la réduction de l'erreur de quantification, elle est décrite au paragraphe 5.1. Nous montrons ici, qu'il est possible d'obtenir une amélioration du RSB pour un débit de transmission plus faible. Ceci est illustré par les deux exemples suivants :

Exemple 1 : On se propose ici d'obtenir en utilisant les méthodes B-KMCR et B-KMCRT un meilleur RSB que celui fourni par la méthode MIC pour un débit $d_1 = 40$ kbps ($B_1 = 5$ bits). Tel qu'il a été expliqué au paragraphe 5.1, les méthodes B-KMCR et B-KMCRT utilisent un plus grand nombre de bits pour coder un échantillon que la méthode MIC à laquelle elles sont comparées. Dans cet exemple, les méthodes à transmission non uniforme utilisent un nombre de bits $B_2 = 6$ bits par échantillon transmis. L'erreur de prédiction fournie par les algorithmes de reconstruction est alors comparée à chaque instant à l'erreur de quantification obtenue par la méthode MIC, $\mathcal{F}_n = e_{n,P}$ et $\mathcal{E}_n = e_{n,Q_1}$. Le tableau 5.1 montre les résultats de cette simulation où B désigne le nombre de bits utilisé par chacune des méthodes pour coder un échantillon, et p est le pourcentage des échantillons transmis exprimé en %. Les performances de la méthode MIC sont rappelées à la première ligne à titre comparatif.

En se référant au tableau 5.1, on constate que les méthodes B-KMCR et B-KMCRT offrent presque un gain de 5dB par rapport au RSB fourni par la méthode MIC. Ceci est aussi

TAB. 5.1 – Comparaison des méthodes MIC, B-KMCR et B-KMCRT pour $B_1 = 5$ et $B_2 = 6$

Méthode	B	$p\%$	$(RSB)_{dB}$	débit de transmission
MIC	5	100	19.33	40
B-KMCR	6	70.89	24.47	37.04
B-KMCRT	6	70.73	24.44	37.01

accompagné d'un gain de 3 kbps par rapport au débit de transmission de la méthode MIC.

Exemple 2 On compare ici les méthodes B-KMCR et B-KMCRT à la méthode MIC pour un faible débit de transmission $d_1 = 24$ kbps ($B_1 = 3$ bits). Pour ce débit, la méthode MIC fournit un $RSB = 5.26$ dB. Les méthodes B-KMCR et B-KMCRT utilisent un nombre de bits $B_2 = 4$ bits par échantillons transmis. Les résultats de la simulation sont présentés au tableau 5.2.

TAB. 5.2 – Comparaison des méthodes MIC, B-KMCR et B-KMCRT pour $B_1 = 3$ et $B_2 = 4$

Méthode	B	$p\%$	$(RSB)_{dB}$	débit de transmission
MIC	3	100	5.26	24
B-KMCR	4	45.48	10.69	17.78
B-KMCRT	4	44.29	10.79	17.37

En se référant au tableau 5.2, on constate que, comme pour les méthodes A-KMCR et A-KMCRT, le gain fourni par les méthodes B-KMCR et B-KMCRT par rapport à la méthode MIC est plus important pour des RSB et des débits de transmission plus faibles. Les performances de la méthode B-KMCRT dépassent légèrement celles de la méthode B-KMCR. Ceci confirme l'intérêt des méthodes utilisant une structure en treillis du filtre pour des RSB et des débits de transmission plus faibles.

Il est à noter ici que pour des débits de transmission élevés, l'erreur de quantification est faible. Le nombre des échantillons non transmis (pour lesquels l'erreur de prédiction est plus faible que celle de quantification) est par conséquent faible de sorte que $pB_2 + \frac{N_d}{N} > B_1$. Dans ce cas, l'amélioration du RSB est alors réalisée au détriment d'une légère augmentation du débit de transmission.

Simulations 3 On se propose dans cette dernière série de simulations de comparer les différentes méthodes de transmission non uniforme utilisées dans les simulations précédentes (A-KMCR, A-KMCRT, B-KMCR et B-KMCRT). Pour comparer ces méthodes, nous utilisons les résultats obtenus grâce aux méthodes B-KMCR et B-KMCRT dans les simulations précédentes (Cf. tableaux 5.1 et 5.2). Ainsi, en se référant au tableau 5.1, les méthodes B-KMCR et B-KMCRT offrent également un RSB de 24.44 dB pour un débit de 37 kbps. Pour les comparer aux méthodes A-KMCR et A-KMCRT, nous fixons le RSB de ces dernières à 24.44 dB. Il reste alors à choisir de manière optimale le nombre de bits B_2 utilisés par ces méthodes pour coder les échantillons transmis. Ainsi, ce RSB est compris entre 19.33 dB et 26 dB. En se référant à la figure 5.3, ces valeurs du RSB sont fournies par la méthode MIC pour des débits de 40 kbps et de 48 kbps respectivement (correspondant à $B_1 = 5$ bits et $B_1 = 6$ bits respectivement). Pour cela nous choisissons, pour les méthodes A-KMCR et A-KMCRT, $B_2 = 6$ bits. Les résultats de cette simulation sont présentés au tableau 5.3. Les résultats obtenus par les méthodes B-KMCR et B-KMCRT sont présentés à titre de comparaison.

TAB. 5.3 – Comparaison des méthodes A-KMCR, A-KMCRT, B-KMCR et B-KMCRT pour le RSB obtenu en 5.1.

Méthode	B_2	$p\%$	$(RSB)_{dB}$	débit de transmission
B-KMCR	6	70.89	24.47	37.04
B-KMCRT	6	70.73	24.44	37.01
A-KMCR	6	72	24.44	37.2
A-KMCRT	6	71	24.44	37.1

En se référant au tableau 5.3, on constate que les quatre méthodes présentent des performances voisines, l'algorithme B-KMCRT étant légèrement meilleur que les autres.

Dans ce qui suit les quatre méthodes sont comparées pour un RSB de 10.79 dB, obtenu par les méthodes B-KMCR et B-KMCRT dans la simulation présentée au tableau 5.2. Ce RSB est compris entre 5.29 dB et 16 dB qui correspondent à des RSB fournis par la méthode MIC pour des débits de 24 kbps et de 32 kbps respectivement ($B_1 = 3$ bits et $B_1 = 4$ bits respectivement). Pour cela, au cours de cette simulation, les méthodes A-KMCR et A-KMCRT utilisent $B_2 = 4$ bits pour coder un échantillon transmis. Le tableau 5.4 montre les résultats obtenus pour les quatres méthodes.

TAB. 5.4 – Comparaison des méthodes A-KMCR, A-KMCRT, B-KMCR et B-KMCRT pour le RSB obtenu en 5.2.

Méthode	B_2	$p\%$	$(RSB)_{dB}$	débit de transmission
B-KMCR	4	45.48	10.69	17.78
B-KMCRT	4	44.29	10.79	17.37
A-KMCR	4	43.40	10.79	16.20
A-KMCRT	4	40.28	10.79	15.14

Le tableau 5.4 montre que les quatres méthodes ont encore une fois des performances voisines. Les méthodes A-KMCR et A-KMCRT présentent néanmoins des débits de transmission plus faibles que ceux obtenus en utilisant les méthodes B-KMCR et B-KMCRT. La méthode A-KMCRT présente pour cette simulation le meilleur débit de transmission moyen. Ceci confirme l'observation faite à la figure 5.5 selon laquelle la méthode A-KMCRT offre de meilleures performances que A-KMCR pour de faibles RSB (ou réciproquement de faibles débits de transmission).

En conclusion, les simulations présentées ci-dessus permettent d'illustrer l'intérêt de la transmission non uniforme par rapport à la méthode MIC. Les méthodes de transmission non uniforme utilisées ont montré des performances voisines. Les méthodes A-KMCR et A-KMCRT sont particulièrement intéressantes lorsqu'on souhaite obtenir un certain RSB pour des débits de transmission plus faibles. D'autre part, les méthodes B-KMCR et B-KMCRT permettent d'obtenir un meilleur RSB que celui obtenu par la méthode MIC (en d'autres termes permettent de réduire l'erreur de quantification) pour un débit de transmission moyen plus faible. Il est à noter ici que les méthodes simulées ci-dessus ont été aussi appliquées à d'autre signaux de parole tests, elles ont montré le même comportement que pour le signal test utilisé ici.

Dans ce qui suit, nous rappelons brièvement le principe du codage MICDA (en anglais ADPCM) normalisé par le CCITT sous le standard G.726 [20]. Dans le but de réduire le débit de transmission obtenu par ce principe de codage, nous proposons un codeur/décodeur utilisant le principe de transmission non uniforme dans un contexte de codage différentiel. Celui-ci est comparé au codage MICDA au paragraphe 5.6.

5.4 Le codage MICDA

Le codage différentiel (MICD) consiste à coder et à transmettre l'erreur de prédiction d'un échantillon sachant que la même approche de prédiction est utilisée à l'émission et à la réception. Si l'approche de prédiction utilisée est performante (ou si les échantillons du signal sont suffisamment corrélés), les erreurs de prédiction à coder sont de faibles amplitudes. Ceci permet de réduire la plage de quantification. Par conséquent, pour une erreur de quantification donnée, le nombre de bits requis pour coder l'erreur de prédiction est inférieur à celui requis pour coder l'échantillon (codage MIC). Ainsi, en comparaison au codage MIC, le codage MICD permet une augmentation du RSB pour un débit donné, ou inversement, une réduction du débit à RSB donné. Or, cette méthode présente des faiblesses lorsqu'elle est appliquée à un signal dont les caractéristiques peuvent varier subitement (par exemple, le passage d'une conversation téléphonique aux données numériques d'un fax). Le codage MICDA a alors été conçu comme solution à de tels problèmes.

Le codage MICDA est fondé sur le principe du codage différentiel. Il utilise une quantification adaptative des erreurs de prédiction. Ainsi, pour un débit de transmission donné, les seuils de quantification sont fixés dynamiquement selon les variations du signal à coder. L'organe qui permet cette quantification adaptative est utilisé à l'émission et à la réception. La figure 5.6 montre un schéma-bloc simplifié d'un système de codage/décodage MICDA. Dans ce qui suit nous rappelons brièvement ses principales fonctions. Ce principe de codage fait l'objet de la norme G.726, il est décrit en détail dans [20].

Quantification adaptative On utilise un quantificateur adaptatif non uniforme à 31, 15, 7 ou 4 niveaux pour quantifier le signal de différence (l'erreur de prédiction $e_{n,P}$) en vue d'un fonctionnement à 40, 32, 24 ou 16 kbit/s respectivement. Avant la quantification, $e_{n,P}$ est convertie en une représentation logarithmique en base 2 avec le facteur d'échelle v_n , qui est calculé au moyen du bloc d'adaptation du facteur d'échelle. La grandeur à quantifier, $e_{n,P}^l$ s'exprime donc par :

$$e_{n,P}^l = \log_2(|e_{n,P}|) - v_n. \quad (5.18)$$

La valeur ainsi calculée permet de déterminer la valeur du code binaire à transmettre I_n en fonction de l'intervalle à laquelle elle appartient. Les recommandations G.726 fournissent, pour chaque débit de transmission, une partition différente de l'ensemble des réels, et associe à chaque intervalle de la partition une valeur I_n . La quantification ainsi obtenue est logarithmique et adaptative grâce à l'utilisation du facteur d'échelle v_n . Le principe de base appliqué pour déterminer le facteur d'échelle du quantificateur est l'adaptation bimodale :

- adaptation "rapide" pour les signaux qui produisent des signaux de différence à grandes fluctuations (par exemple, signaux vocaux) ;
- adaptation "lente" pour les signaux qui produisent des signaux de différence à fluctuations faibles (par exemple, données transmises dans la bande vocale, tonalités).

La vitesse d'adaptation est déterminée par une combinaison de facteurs d'échelle à adaptation lente et rapide. Les facteurs d'échelle à adaptation lente et rapide v_n^r et v_n^l , calculés récursivement grâce aux équations suivantes :

$$v_n^r = (1 - 2^{-5}) v_n + 2^{-5} W[I_n], \quad (5.19)$$

$$v_n^l = (1 - 2^{-6}) v_{n-1}^l + 2^{-6} v_n^r, \quad (5.20)$$

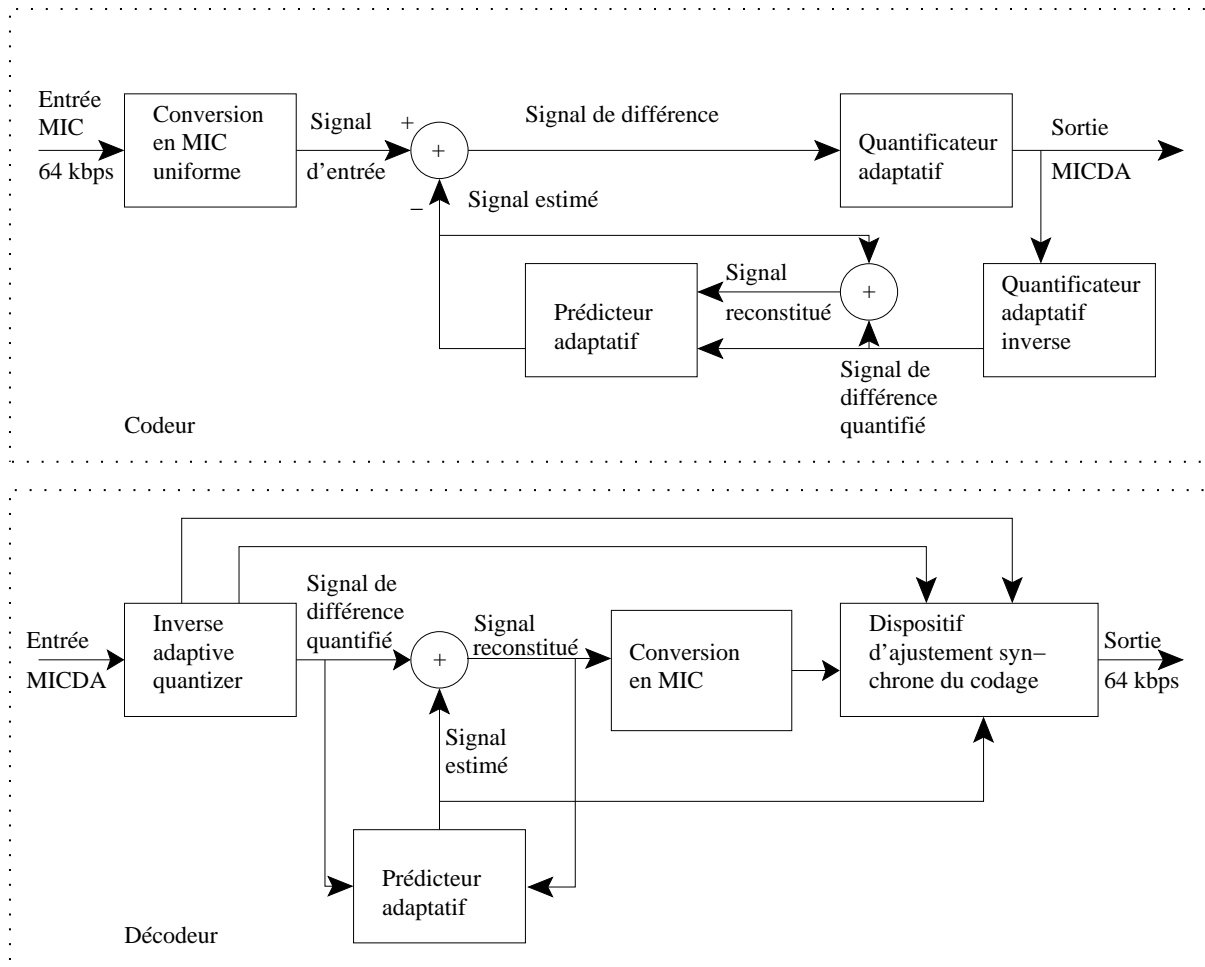


FIG. 5.6 – Schéma simplifié d'un codeur décodeur MICDA.

où $W [I_n]$ est une fonction discrète croissante de I_n , dont les valeurs sont fournies dans les recommandations G.726. Le facteur d'échelle est alors obtenu en combinant v_n^r et v_n^l selon l'équation suivante :

$$v_n = \rho_n v_{n-1}^r + (1 - \rho_n) v_{n-1}^l, \quad (5.21)$$

où $0 \leq \rho_n \leq 1$ est un paramètre de contrôle. Il est compris entre 0 et 1. Sa valeur tend vers l'unité pour les signaux vocaux et vers zéro pour les signaux de données transmis dans la bande vocale. Il se déduit de la mesure du taux de variation du signal de différence. Le calcul de ce paramètre est détaillé dans [20].

Il est à noter ici que le calcul du facteur d'échelle est également utilisé dans le quantificateur inverse. Un autre organe essentiel du codage MICDA est le prédicteur adaptatif.

Prédicteur adaptatif a pour fonction essentielle de calculer la valeur estimée de l'échantillon $\hat{x}_{n|n-1}$. Le signal est caractérisé par un modèle ARMA(2,6). Ainsi, soit $\hat{x}_{n|n-1}$ la valeur estimée de l'échantillon x_n et $\hat{x}_{n|n}$ l'échantillon reconstitué en utilisant l'erreur de prédiction quantifiée $e_{n,P}^Q$. Ainsi, l'échantillon x_n est estimé en fonction des échantillons reconstitués et des erreurs de prédiction quantifiées précédents via l'équation suivante :

$$\hat{x}_{n|n-1} = \sum_{i=1}^2 a_{i,n-1} \hat{x}_{n|n} + \sum_{i=1}^6 b_{i,n-1} e_{n,P}^Q, \quad (5.22)$$

où les coefficients a_i et b_i sont estimés d'une manière adaptative en fonction de la dynamique du signal et du signal de différence (l'erreur de prédiction quantifiée) (pour plus de détails sur l'estimation des paramètres se référer à [20]). Ainsi, l'erreur de prédiction est donnée par :

$$e_{n,P} = x_n - \hat{x}_{n|n-1}. \quad (5.23)$$

Cette valeur est alors quantifiée et transmise. A la réception, le même prédicteur adaptatif que celui utilisé à l'émission permet de calculer $\hat{x}_{n|n-1}$. L'échantillon est alors reconstitué en utilisant la valeur quantifiée de l'erreur de prédiction $e_{n,P}^Q$ via l'équation suivante :

$$\hat{x}_{n|n} = \hat{x}_{n|n-1} + e_{n,P}^Q. \quad (5.24)$$

Détecteur de tonalité et de transition il permet la détection de la transition d'un signal à bande étroite (tonalité par exemple) à un signal à large bande et vice versa. Ceci permet donc l'adaptation du codage. Cet organe est utile pour améliorer le fonctionnement dans le cas de signaux provenant de modems à modulation par déplacement de fréquence (MDF) en mode caractère. Nous nous intéressons ici uniquement aux signaux de voix.

Dans ce qui suit nous proposons d'utiliser le principe de transmission non uniforme dans un système de codage MICDA dans le but d'améliorer le débit de transmission de ce dernier.

5.5 Codage MICDA à transmission non uniforme

5.5.1 Transmission non uniforme dans un contexte de codage différentiel

On rappelle que les méthodes de codage différentiel sont intéressantes lorsque l'approche de prédiction utilisée est performante de sorte que les erreurs de prédiction soit de faibles amplitudes. Ainsi, il est possible que l'erreur de prédiction de certains échantillons soit inférieure au pas minimal de quantification fourni par le quantificateur adaptatif. Dans ce cas, l'erreur de prédiction

est codée par la valeur $I_n = 0$. Nous parlerons de "prédiction parfaite", lorsque l'erreur de prédiction est quantifiée à zéro. Si l'approche de prédiction utilisée est suffisamment performante, un nombre considérable d'échantillons sera parfaitement prédits. Nous proposons alors de ne pas transmettre dans ce cas l'erreur de prédiction quantifiée espérant ainsi réduire le nombre de bits transmis (donc réduire le débit de transmission moyen) sans toutefois perdre en termes de qualité de reconstitution du signal. Pour cela, le prédicteur adaptatif utilisé à l'émission et à la réception doit être adapté à l'estimation de signaux à échantillons manquants. Par ailleurs, la transmission d'un bit drapeau indicateur de la disponibilité ou non d'une erreur de prédiction quantifiée est requise.

5.5.2 Description du système à transmission non uniforme

Nous proposons ici d'utiliser le principe de transmission non uniforme décrit ci-dessus dans un système de codage différentiel dans le but de réduire son débit de transmission. L'utilisation d'une quantification adaptative telle que celle utilisée dans le codage MICDA (Cf. paragraphe 5.4) permet également d'améliorer les performances. Nous proposons alors de modifier le principe de codage/décodage MICDA décrit au paragraphe 5.4 pour l'adapter à la transmission non uniforme des erreurs de prédiction quantifiées. Le système proposé diffère donc du MICDA principalement par le prédicteur adaptatif utilisé. Il contient en plus un organe qui décide de la transmission de l'erreur de prédiction quantifiée ainsi que du bit indicateur.

Prédicteur adaptatif Il doit être adapté à l'estimation de signaux à échantillons manquants. Nous proposons alors d'utiliser l'un des algorithmes décrits aux chapitres 2, 3 et 4. L'algorithme de reconstruction adaptatif qui utilise la structure en treillis du filtre (Cf. chapitre 4) a montré de meilleures performances que les deux autres particulièrement dans le cas de faible débit de transmission tel qu'il est le cas du codage MICDA. Nous proposons alors de l'utiliser comme prédicteur adaptatif dans ce système. Ainsi, le signal est caractérisé par un processus AR(L) non stationnaire au lieu que par un processus ARMA(2,6). Dans le cas de transmission de la parole, nous utilisons un modèle AR d'ordre $L = 10$ à 12 . Nous rappelons ici que le prédicteur adaptatif ainsi choisi utilise un filtre de Kalman pour la prédiction des échantillons manquants et l'algorithme de Burg adaptatif pour l'estimation des coefficients de réflexion et par conséquent les paramètres du modèle AR. Lorsqu'un échantillon est disponible, l'état constitué des L derniers échantillons est filtré par le filtre de Kalman grâce à l'équation (3.18). La mise à jour de l'état est alors proportionnelle à l'erreur de prédiction de l'échantillon. Or, dans le cas du système à transmission non uniforme, le récepteur ne dispose que d'une version quantifiée de l'erreur de prédiction. Pour cela, même si la vraie valeur de l'échantillon est connue à l'émission, nous utilisons l'erreur de prédiction quantifiée pour filtrer l'état grâce au filtre de Kalman. Ceci permet de reproduire fidèlement, à l'émission, la prédiction telle qu'elle est effectuée à la réception. Ainsi, lorsqu'une erreur de prédiction quantifiée est transmise, l'échantillon reconstitué est donné par l'équation (5.24). L'échantillon ainsi reconstitué est utilisé tant à la réception qu'à l'émission pour initialiser l'algorithme de Burg adaptatif.

Organe de décision Ayant la valeur réelle de l'échantillon et sa prédiction fournie par le prédicteur adaptatif, l'erreur de prédiction est calculée. Celle-ci est quantifiée par un quantificateur adaptatif tel que celui proposé par les recommandations G.726. Si l'échantillon est parfaitement prédit (erreur de prédiction quantifiée à zéro), le code correspondant à l'erreur quantifiée n'est pas transmis. Si nécessaire un bit drapeau de valeur zéro est transmis. Dans le but de réduire le nombre de bits drapeaux transmis, le principe de transmission de ces bits décrit au paragraphe

5.2 est utilisé. Ainsi, on transmet un bit 1 (resp. 0) lorsqu'un échantillon est transmis alors que le précédent ne l'est pas (resp. lorsqu'un échantillon n'est pas transmis alors que le précédent l'est). A l'émission, le prédictif adaptatif devrait être informé de la transmission ou non d'un échantillon pour exécuter les étapes convenables qui seront également exécutées à la réception.

D'autre part, le but étant de réduire le nombre de bits transmis, le nombre d'échantillons non transmis devrait compenser la transmission du bit drapeau supplémentaire. Ainsi, soit un signal de N échantillons codé en MICDA sur B bits (dans le cas du MICDA, B peut prendre les valeurs 5, 4, 3 ou 2), le nombre total de bits transmis est alors : $B_{MICDA} = NB$ bits. Le système MICDA à transmission non uniforme, contrairement au système décrit au paragraphe 5.2, utilise pour coder une erreur de prédiction transmise le même nombre de bits B que le MICDA classique auquel il est comparé. Le nombre total de bits transmis par le système MICDA à transmission non uniforme est alors de $B_{NU} = pNB + N_d$, où p est le taux d'échantillons transmis et N_d le nombre de bits drapeaux transmis. La condition de rentabilité du système MICDA à transmission non uniforme s'écrit alors :

$$pNB + N_d < NB. \quad (5.25)$$

La figure 5.7 présente un schéma en bloc simplifié du codeur décodeur MICDA à transmission non uniforme (MICDA-TNU) décrit ci-dessus.

Dans ce qui suit, le système ainsi proposé est comparé par des simulations au codage MICDA.

5.6 Simulations

Le système MICDA à transmission non uniforme décrit ci-dessus est comparé ici au système de codage MICDA classique par des simulations sur des signaux de paroles. Dans le but de comparer aussi par rapport aux systèmes A-KMCR, A-KMCRT, B-KMCR et B-KMCRT, nous utilisons ici le même signal de parole test que celui utilisé au paragraphe 5.3. Nous rappelons que c'est la phrase usuelle '*Mary had a little lamb, its fleece was white as snow*' enregistrée par la voix d'un homme. Le signal est échantillonné à la fréquence $F_e = 8$ kHz. Dans ce qui suit, nous appelons respectivement MICDA et MICDA-TNU, le codeur décodeur MICDA classique et celui à transmission non uniforme. Les méthodes MICDA et MICDA-TNU sont alors appliqués à la compression du signal test pour les différentes valeurs de B possibles ($B = 2, 3, 4$ et 5). Nous rappelons que B est le nombre de bits utilisé pour coder une erreur de prédiction (appelé signal différence dans les recommandations G.726). Ces méthodes sont programmées en langage MATLAB et simulées sur un ordinateur Pentium 4 dont la vitesse du processeur est de 3GHz. Les résultats de ces tests sont présentés dans les tableaux 5.5, 5.6, 5.7 et 5.8, où le CPU est le temps de calcul en secondes requis pour chacune des méthodes.

TAB. 5.5 – Comparaison des méthodes MICDA-TNU et MICDA pour un nombre de bits $B = 2$ par échantillon.

Méthode	B	$p\%$	débit (kbps)	$(RSB)_{dB}$	CPU (s)
MICDA	2	100	16	4.86	5.5
MICDA-TNU	2	23	7	11,67	6.6

En se référant aux tableaux 5.5, 5.6, 5.7 et 5.8, on constate que la méthode MICDA-TNU offre à chaque fois un meilleur débit de transmission moyen que la méthode MICDA à laquelle elle est comparée. Ce gain en débit de transmission est aussi accompagné d'une augmentation du RSB grâce à l'utilisation d'un prédictif adaptatif performant. Il est à noter ici que des tests d'écoute ont aussi été effectués. Ils montrent que la qualité du signal codé par la méthode MICDA-TNU

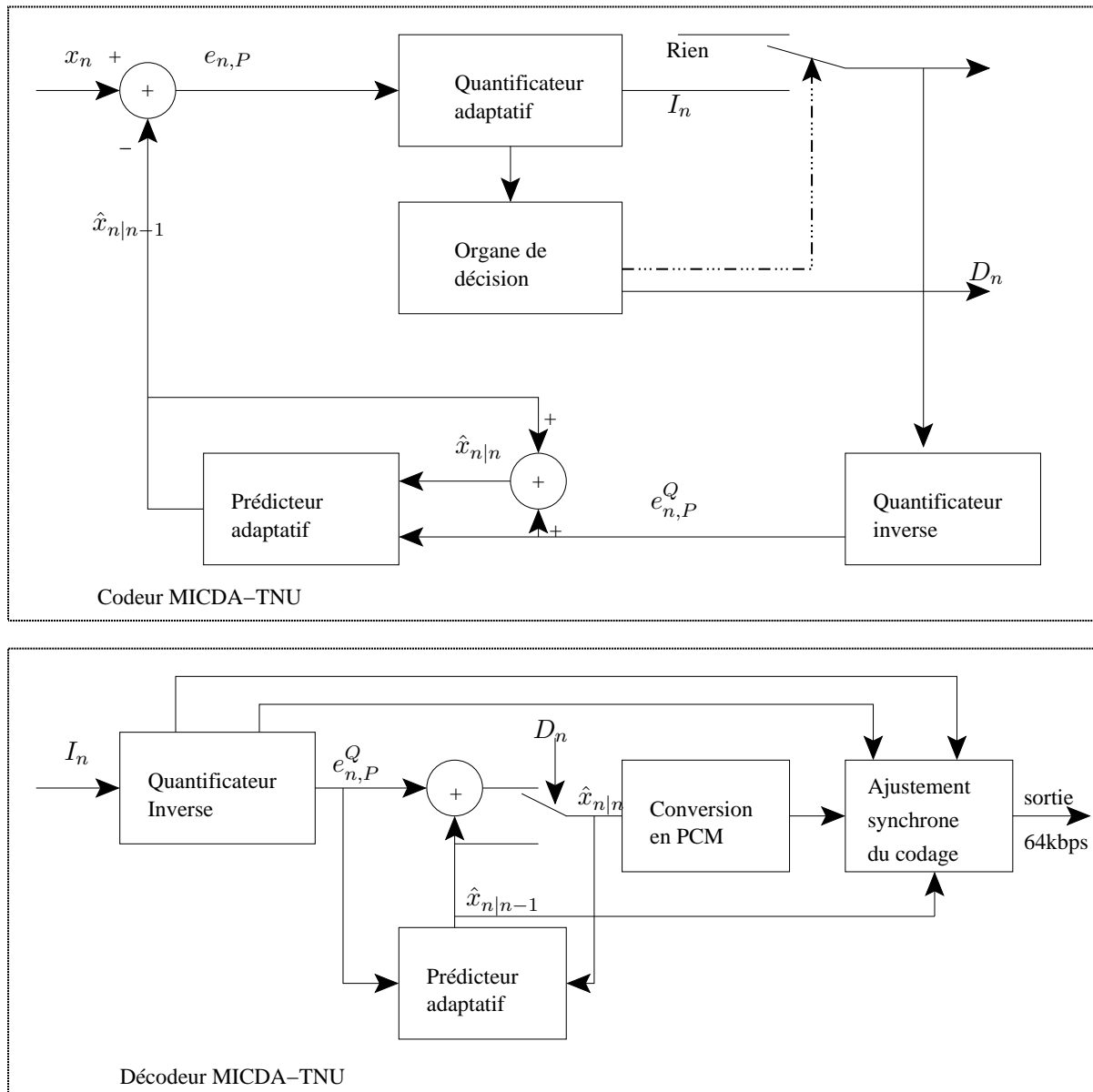


FIG. 5.7 – Schéma simplifié d'un codeur décodeur MICDA à transmission non uniforme.

TAB. 5.6 – Comparaison des méthodes MICDA-TNU et MICDA pour un nombre de bits $B = 3$ par échantillon.

Méthode	B	$p\%$	débit (kbps)	$(RSB)_{dB}$	CPU (s)
MICDA	3	100	24	17.25	5.5
MICDA-TNU	3	70	20	20.23	7.89

TAB. 5.7 – Comparaison des méthodes MICDA-TNU et MICDA pour un nombre de bits $B = 4$ par échantillon.

Méthode	B	$p\%$	débit (kbps)	$(RSB)_{dB}$	CPU (s)
MICDA	4	100	32	23.5	5.9
MICDA-TNU	4	85	29	29.78	8.6

TAB. 5.8 – Comparaison des méthodes MICDA-TNU et MICDA pour un nombre de bits $B = 5$ par échantillon.

Méthode	B	$p\%$	débit (kbps)	$(RSB)_{dB}$	CPU (s)
MICDA	5	100	40	29.5	6.1
MICDA-TNU	5	88	37	33.02	8.9

est meilleure que celle du signal codé par la méthode MICDA lorsque les deux méthodes utilisent le même nombre de bits pour coder un échantillon. L'amélioration des performances du MICDA grâce à la transmission non uniforme se fait néanmoins aux dépens d'une légère augmentation du temps de calcul. En se référant à la figure 5.5, nous constatons que les performances de la méthode MICDA-TNU dépassent aussi celles des méthodes A-KMCR et A-KMCRT (et par conséquent celles des méthodes B-KMCR et B-KMCRT).

Conclusions et perspectives

Conclusions

Les travaux présentés dans le cadre de cette thèse portent sur le traitement des signaux échantillonnés non uniformément. Nous nous sommes particulièrement intéressés à l'identification et à la reconstruction adaptatives de séries chronologiques à échantillons manquants. Les signaux traités sont modélisés par un processus AR non stationnaire.

Nous avons étudié premièrement l'extension des algorithmes de gradient au cas de signaux à échantillons manquants pour leur simplicité. Mirsaidi *et al.* [51, 52, 53] ont proposé dans cette optique un ensemble d'algorithmes simples du gradient utilisant la prédiction par régression linéaire modifiée. Ces algorithmes ont été décrits et analysés notamment en termes de biais d'identification. Pour résoudre ce problème, nous avons proposé deux alternatives à ces algorithmes qui utilisent deux autres approches de prédiction, le prédicteur à k -pas et celui à passé infini incomplet proposé par Bondon [13]. Les deux algorithmes ainsi obtenus convergent en moyenne vers une estimation non biaisée des paramètres. L'algorithme qui utilise la prédiction à k -pas est simple mais souffre d'une erreur de reconstruction importante. Alors que l'algorithme basé sur le prédicteur à passé infini incomplet permet une reconstruction optimale au sens des moindres carrés du signal aux dépens d'une plus grande complexité calculatoire.

Ayant noté la faiblesse des algorithmes de gradient en vitesse de convergence, nous nous sommes intéressés par la suite à l'extension de l'algorithme des moindres carrés récursifs (MCR) à l'identification de signaux à échantillons manquants. L'algorithme MCR pseudo-linéaire [1, 62, 63] est alors présenté et analysé. Cet algorithme converge typiquement vers une estimation biaisée des paramètres lorsque l'approche de prédiction utilisée est la régression linéaire modifiée [69]. Il fut généralisé dans [63] par l'introduction de plusieurs prédicteurs possibles à la place de la prédiction par régression linéaire modifiée. Nous avons alors utilisé conjointement avec cet algorithme un filtre de Kalman qui permet une prédiction optimale au sens des moindres carrés de processus AR(L) à échantillons manquants. L'algorithme résultant permet une identification non biaisée des paramètres lorsqu'on ne considère pas de facteur d'oubli. Lorsqu'un facteur d'oubli est utilisé ($\lambda < 1$), les expressions asymptotiques du biais ainsi que de la variance des paramètres estimés ont été établies. Ils sont de l'ordre de $(1 - \lambda)$. Les simulations montrent qu'un bon choix du facteur d'oubli ($\lambda \rightarrow 1$) permet d'obtenir un compromis entre une bonne vitesse de convergence et des biais des paramètres estimés négligeables. Cet algorithme est rapide, il a été appliqué à la reconstruction de signaux de parole à échantillons manquants, il a montré de bonnes performances en termes de RSB.

Souhaitant contrôler la stabilité du filtre identifié, nous nous sommes intéressés ensuite à l'identification des signaux à échantillons manquants en utilisant la structure en treillis du filtre. Ainsi, nous avons proposé une extension de l'algorithme de Burg adaptatif à l'identification de processus AR à échantillons manquants en utilisant pour la prédiction un filtre de Kalman (le même que celui utilisé avec l'algorithme MCR pseudo-linéaire). À notre connaissance, c'est

le premier algorithme qui permet un traitement en ligne de signaux à échantillons manquants en utilisant un filtre en treillis. L'algorithme résultant permet de garantir la stabilité du filtre correspondant aux paramètres identifiés. En outre, il est robuste à de larges probabilités de perte. De plus, les simulations montrent qu'il offre une convergence rapide et un bon suivi des paramètres, même pour des probabilités de perte d'échantillons élevées. L'application de cet algorithme à la reconstruction de signaux de parole montre que ses performances dépassent en termes d'erreur de reconstruction celles des méthodes qui utilisent la représentation directe du filtre et ce d'autant plus que la probabilité de perte est élevée. Des tests d'écoute ont permis de confirmer ces résultats. En effet, un signal de parole reconstruit en utilisant l'algorithme proposé reste intelligible même lorsque 85 % des échantillons sont manquants.

Dans la dernière partie de cette thèse, les algorithmes adaptatifs d'identification et de reconstruction des signaux à échantillons manquants ont été appliqués à l'échantillonnage adaptatif et à la transmission non uniforme de signaux. Ainsi, plusieurs méthodes de transmission non uniforme des signaux ont été présentées. Elles permettent de ne transmettre qu'un nombre minimal d'échantillons choisis de telle sorte qu'il soit possible de reconstruire le signal avec une erreur tolérée. En combinant les méthodes de transmission non uniforme et les algorithmes adaptatifs de reconstruction, des systèmes de codage à transmission non uniforme des signaux ont été conçus. Dans un premier temps, nous avons étudié la transmission non uniforme de signaux codés en MIC. Deux systèmes fondés sur deux méthodes de transmission non uniforme différentes ont été conçus. La première méthode est particulièrement intéressante lorsqu'on souhaite obtenir un RSB donné pour un débit de transmission plus faible que celui requis par le codage MIC. La deuxième méthode permet d'obtenir un meilleur RSB que celui obtenu par la méthode MIC (en d'autres termes elle permet de réduire l'erreur de quantification) pour un débit de transmission moyen plus faible. Ces méthodes ont été comparées au codage MIC par simulation sur des signaux de parole et ont montré une amélioration simultanée en termes de RSB et de débit de transmission. Nous nous sommes intéressés par la suite à la transmission non uniforme dans un contexte de codage différentiel. Une méthode de transmission non uniforme des erreurs de prédiction est alors proposée. En combinant cette méthode et les méthodes paramétriques de reconstruction des signaux, un système de codage MICDA à transmission non uniforme (MICDA-TNU) a été conçu. Il diffère du codage MICDA décrit dans les recommandations G.726 par le prédicteur adaptatif utilisé et par l'introduction d'un organe de décision sur la transmission ou non des erreurs de prédiction quantifiées. Ce principe de codage a été comparé au codage MICDA du standard G.726 par des simulations sur des signaux de parole, il a montré une amélioration simultanée du débit de transmission ainsi que du RSB des signaux reconstruits. Ainsi, lorsque les erreurs de prédiction sont codées sur 4 bits, le codage MICDA-TNU offre un débit de transmission moyen plus faible de 10% que celui obtenu par le codage MICDA. Ceci est accompagné d'une amélioration de 24% du RSB (en dB) par rapport à celui fourni par le codage MICDA.

Perspectives

Les travaux effectués dans le cadre de cette thèse offrent de nombreuses perspectives. Nous en résumons ici quelques unes.

Nous suggérons qu'une étude théorique approfondie de la convergence des algorithmes proposés soit menée dans des travaux ultérieurs, en particulier, l'analyse de la convergence de l'algorithme en treillis. Il serait également intéressant de trouver une expression analytique du temps de convergence des algorithmes proposés telle que celles établies pour les algorithmes de gradient et MCR sans échantillons manquants. De plus, une étude théorique sur le choix du facteur d'oubli ainsi que du pas de convergence (dans le cas des algorithmes du gradient) est requise. Ceci

permet, par exemple, de trouver un seuil pour le choix du facteur d'oubli permettant la stabilité numérique de l'algorithme MCR pseudo-linéaire.

D'autre part, l'extension des algorithmes proposés à l'identification et la reconstruction de processus ARMA à échantillons manquants pourrait faire l'objet d'un travail futur. Ceci est le cas particulièrement des algorithmes de gradient. En outre, il serait intéressant d'étudier le comportement de ces algorithmes dans le cas de signaux bruités. Pour cela, ces algorithmes doivent être modifiés pour traiter ce problème. Ainsi, par exemple, un bruit d'observation peut être considéré dans la représentation d'état de processus AR. Ceci implique une modification des équations du filtre de Kalman correspondant. De plus, une extension au cas des signaux bidimensionnels est également envisageable. Ceux-ci seront alors appliqués à la reconstruction et à la transmission non uniforme d'images par exemple.

Les signaux à échantillons manquants étant une forme de signaux échantillonnés non uniformément, l'extension des algorithmes proposés à l'analyse spectrale de signaux à échantillonnage aléatoire est alors envisageable. Ceci implique l'identification de processus continus échantillonnés de manière aléatoire. De tels processus peuvent être modélisés par des processus CAR, CARMA, etc.

Par ailleurs, nous avons observé, en appliquant les algorithmes proposés à la transmission non uniforme, une nette amélioration en termes de débit de transmission moyen et de RSB par rapport aux systèmes de codage à transmission périodique. La qualité des signaux compressés en utilisant ces méthodes peuvent être évalués par des tests psycho-acoustiques. Ceci permet d'envisager des applications de compression de signaux de parole et de musique.

Annexe A

Représentation autorégressive du prédicteur à k -pas pour un AR(2)

On se propose ici de trouver une représentation autorégressive du prédicteur à k -pas en fonction de L échantillons consécutifs disponibles dans le cas d'un processus AR(2). Ainsi, soit à prédire x_{n+k} à partir des échantillons disponibles x_n et x_{n-1} . La représentation autorégressive du prédicteur à k -pas \hat{x}_{n+k} en fonction de x_n et x_{n-1} s'écrit alors sous la forme :

$$\hat{x}_{n+k} = g_1^k x_n + g_2^k x_{n-1}. \quad (\text{A.1})$$

Il s'agit ici de trouver les expressions analytiques de g_1^k et de g_2^k en fonction des paramètres du modèle AR, a_1 et a_2 . Or, en se référant à l'équation (2.19), \hat{x}_{n+k} s'écrit sous la forme suivante :

$$\hat{x}_{n+k} = a_1 \hat{x}_{n+k-1} + a_2 \hat{x}_{n+k-2}, \quad (\text{A.2})$$

$$= a_1 (g_1^{k-1} x_n + g_2^{k-1} x_{n-1}) + a_2 (g_1^{k-2} x_n + g_2^{k-2} x_{n-1}). \quad (\text{A.3})$$

On en déduit que les coefficients autorégressifs $(g_i^k)_{i=1,2}$ vérifient les équations de récurrence sur le pas k suivantes :

$$g_i^k = a_1 g_i^{k-1} + a_2 g_i^{k-2} \quad i = 1, 2. \quad (\text{A.4})$$

Or, $\hat{x}_{n+1} = a_1 x_n + a_2 x_{n-1}$ on a donc $g_1^1 = a_1$ et $g_2^1 = a_2$. De plus, $\hat{x}_n = x_n$, on en déduit que $g_1^0 = 1$ et que $g_2^0 = 0$. Ceci permet d'initialiser les équations de récurrence (A.4). On a alors,

$$\begin{array}{l} k = 0 \\ k = 1 \\ k = 2 \\ k = 3 \end{array} \left| \begin{array}{ll} g_1^0 = 1 & g_2^0 \\ g_1^1 = a_1 & g_2^1 = a_2 \\ g_1^2 = a_1^2 + a_2 & g_2^2 = a_1 a_2 \\ g_1^3 = a_1^3 + 2a_1 a_2 & g_2^3 = a_1^2 a_2 + a_2^2 \end{array} \right. \quad (\text{A.5})$$

Ainsi, nous trouvons alors par récurrence que :

$$\begin{aligned} g_1^k &= \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} C_{k-i}^i a_1^{h-2i} a_2^i \\ g_2^k &= \sum_{i=0}^{\lfloor \frac{k-1}{2} \rfloor} C_{k-i-1}^i a_1^{h-2i-1} a_2^{i+1}. \end{aligned} \quad (\text{A.6})$$

Or l'erreur quadratique moyenne du prédiction qui est aussi dans ce cas la variance du prédicteur à k -pas est donnée par,

$$Var = \mathbf{E} \{ (x_{n+k} - \hat{x}_{n+k})^2 \} \quad (\text{A.7})$$

$$= \mathbf{E} \{ (x_{n+k} - g_1^k x_n + g_2^k x_{n-1})^2 \}. \quad (\text{A.8})$$

Ainsi en remplaçant g_1^k et g_2^k par leurs expressions données par l'équations (A.6), nous obtenons une expression de la variance du prédicteur en fonction des paramètres et du pas k donnée par :

$$Var = \mathbf{E} \{ (x_{n+k} - \hat{x}_{n+k})^2 \} = \left(\sum_{i=0}^{\lfloor \frac{k-2}{2} \rfloor} C_{k-i}^i a_1^{h-2i} a_2^i \right)^2 \sigma_\epsilon^2. \quad (\text{A.9})$$

Annexe B

Représentation autorégressive du prédicteur à passé incomplet pour un AR(L)

Nous nous intéressons ici à la représentation autorégressive du prédicteur à passé infini incomplet dans le cas d'un processus AR d'ordre fini L à échantillons manquants. Nous montrons, en utilisant les expressions de ce prédicteur fournie dans [13], que la prédiction optimale d'un échantillon est une combinaison linéaire de toutes les observations précédentes jusqu'à ce que L échantillons consécutifs disponibles soient rencontrés.

Ainsi soit à prédire l'échantillon x_n en fonction de son passé incomplet, supposons que les données $x_{n-n_1}, \dots, x_{n-n_M}$ pour $0 < n_1 < \dots < n_m < \dots < n_M$ sont manquantes, M étant leur nombre. La représentation autorégressive du prédicteur à passé infini incomplet est donnée par [13, 11] :

$$\hat{x}_n = \sum_{i \in \mathbb{N} - \mathcal{M}} g_i x_{n-i}, \quad (\text{B.1})$$

où $\mathcal{M} = \{n_1, \dots, n_M\}$ et

$$g_i = \delta_{i,0} - \sum_{s=0}^M \psi_s \sum_{j=0}^{n_s \wedge i} a_{n_s-j} a_{i-j}. \quad (\text{B.2})$$

$\delta_{i,0}$ étant le symbole de Kronecker.

Les coefficients (ψ_s) vérifient l'équation matricielle suivante :

$$U [\psi_0, \psi_1, \dots, \psi_M]^\top = [1, 0, \dots, 0]^\top \quad (\text{B.3})$$

U est la matrice non singulière de dimensions $(M+1) \times (M+1)$, dont les éléments sont donnés par :

$$u_{v,w} = \sum_{j=0}^{n_v \wedge n_w} a_{n_v-j} a_{n_w-j} \quad v, w = 0, \dots, M, \quad (\text{B.4})$$

où $n_v \wedge n_w$ est la valeur minimale entre n_v et n_w .

Nous montrons tout d'abord que les éléments $u_{v,w}$ de la matrice U sont nuls lorsque $|n_v - n_w| > L$. Ainsi, considérons le cas où $v < w$ ($n_v < n_w$ avec $n_w - n_v > L$), en se référant à l'équation (B.4), le terme $u_{v,w}$ est alors donné par :

$$u_{v,w} = \sum_{j=0}^{n_v} a_{n_v-j} a_{n_w-j}. \tag{B.5}$$

Or pour tout $j \leq n_v$, on a $n_w - j > L$. Par conséquent, pour un modèle AR(L), tous les termes a_{n_w-j} dans l'équation (B.5) sont nuls. Il s'en suit que $u_{v,w} = 0$ lorsque $n_w - n_v > L$. Il en est de même pour le cas où $n_v - n_w > L$.

Soit $x_{n-k-L+1}, \dots, x_{n-k}$ la séquence de L échantillons consécutifs disponibles la plus proche de x_n . Il s'agit alors de montrer que les coefficients g_i de la représentation autorégressive de \hat{x}_n donnés par (1.59) s'annulent pour $i > n - k - L + 1$.

Ainsi, soit $n - n_m = n - k + 1$ l'instant auquel un échantillon est manquant qui suit directement les L échantillons consécutifs disponibles. L'échantillon manquant précédent qui est à l'instant $n - n_{m+1}$, est alors séparé de celui qui suit (à l'instant $n - n_m$) de plus de L échantillons ($n_m - n_{m+1} > L$). Par conséquent, les termes $u_{v,w}$ de la matrice U sont tous nuls pour $v \leq m$ et $w \geq m$. Ainsi, lorsque L échantillons consécutifs sont disponibles dans le passé, la matrice U prend la forme suivante :

$$U = \begin{bmatrix} u_{0,0} & \dots & u_{0,m} & & & \\ \vdots & \ddots & \vdots & & & \\ u_{m,0} & \dots & u_{m,m} & & & \\ & & & u_{m+1,m+1} & \dots & u_{m+1,M} \\ & & & \vdots & \ddots & \vdots \\ & & & & & & \\ & & & & & & & u_{M,m+1} & \dots & u_{M,M} \end{bmatrix}. \tag{B.6}$$

où m est le nombre des échantillons manquants qui suivent les L échantillons consécutifs disponibles les plus proches de n . En remplaçant la matrice U ainsi obtenue dans le système matriciel donné par l'équation (B.3), on obtient que $\psi_s = 0$ pour $s > m$. Par conséquent, les coefficients g_i de la représentation autorégressive de \hat{x}_n s'écrivent dans ce cas sous la forme suivante :

$$g_i = \delta_{i,0} - \sum_{s=0}^m \psi_s \sum_{j=0}^{n_s \wedge i} a_{n_s-j} a_{i-j}. \tag{B.7}$$

Or lorsque $i \geq n - k - L$, on a $i - n_s > L$ pour tout $s \leq m$, les termes a_{i-j} de l'équation (B.7) sont alors tous nuls. On en déduit que les coefficients g_i de la représentation autorégressive de \hat{x}_n sont nuls pour $i > n - k - L + 1$.

Bibliographie

- [1] G.J. Adams, P. Albertos and G.C. Goodwin and A. Isaksson, "Parameter estimation for systems with missing data in the presence of white measurement noise," *IFAC Symposium on System Identification*, Copenhagen, 1994.
 - [2] P. Albertos, R. Sanchis and A. Sala, "Scarce data operating conditions : output prediction", *IFAC Symposium on System Identification*, pp. 469–474, Japan, 1997.
 - [3] P. Albertos, R. Sanchis and A. Sala, "Output prediction under scarce data operation : control application", *Automatica*, 35, pp. 1671–1681, 1999.
 - [4] L. Arvaston, H. Olsson and J. Holst, "Asymptotic bias in parameter estimation of AR-processes using recursive least squares with exponential forgetting", *Scandinavian journal of statistics*, vol. 27, pp. 177–192, 2000.
 - [5] K. Asswad, E. Lahalle and J. Oksman, "Reducing quantization error and bit rate transmission using prediction/quantization," *IEEE Nordic Signal Processing Symposium*, pp. 133–136, Kolmarden, Sweden, Juin 2000.
 - [6] K. Asswad, E. Lahalle and J. Oksman, "Real-time reconstruction of 2D signals with missing observations," *European Signal Processing Conference EUSIPCO*, vol. IV, pp. 2073–2076, Tampere (Finlande), September 2000.
 - [7] K. Asswad, E. Lahalle and J. Oksman, "Reconstruction en temps réel de signaux bidimensionnels à échantillons manquants," *Traitement du Signal*, vol. 18, No. 2, 2001.
 - [8] K. Asswad, "Echantillonnage et identification paramétriques adaptatifs pour des signaux bidimensionnels à échantillons manquants," *Thèse de doctorat en sciences de l'Université Paris XI*, 2002.
 - [9] M. Bellanger, "Adaptive digital filters : Second edition, Revised and expanded" *Signal processing and communications series, Marcel Dekker*, 2001.
 - [10] P. Bondon, "Prediction of a stationary signal with missing observations," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pp. 332–335, Istanbul, Turkey, 2000.
 - [11] P. Bondon, "Représentation autorégressive du prédicteur à passé infini incomplet d'une série chronologique stationnaire," *Comptes Rendus de l'Académie des Sciences. Série I. Mathématique 330*, pp. 915–920, 2000.
 - [12] P. Bondon, "Recursive relations for multistep prediction of a stationary time series," *Journal of Time Series Analysis*, vol. 22, No. 4, pp. 399–410, 2001.
 - [13] P. Bondon, "Prediction with incomplete past of a stationary process," *Elsevier, Stochastic processes and their applications*, pp. 67–76, 2002.
 - [14] P. Brockwell and R. Davis, *Time Series : Theory and methods*. Springer-Verlag, 1991.
-

-
- [15] P. Broersen, S. Waele and R. Bos, "Estimation of autoregressive spectra with randomly missing data," *IEEE Instrumentation and Measurement Technology Conference*, CO, USA, May 2003.
- [16] P. Broersen, "Accurate estimation of the autocovariance function of random observations," *IEEE Instrumentation and Measurement Technology Conference*, Como, Italy, May 2004.
- [17] P. Broersen and R. Bos, "Estimation of time series spectra with randomly missing data," *IEEE Instrumentation and Measurement Technology Conference*, Como, Italy, May 2004.
- [18] P. Broersen, S. Waele and R. Bos, "Autoregressive spectral analysis when observations are missing," *Automatica*, 40, pp. 1495–1504, 2004.
- [19] J.P. Burg, "Maximum entropy spectral analysis," *Ph.D. dissertation, Geophysics Dept., Stanford University*, Stanford, CA, USA, May 1975.
- [20] "Modulation par impulsion et codage différentiel (MICDA) à 40, 32, 24, 16 kbit/s," *CCITT : Recommendation G.726*, Genève, 1990.
- [21] M.C.Chevalier-Omnes, G.Chollet and Y.Grenier, "Speech analysis and restitution using time-dependent autoregressive models," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp 501-504, Tampa, Florida, 1985.
- [22] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp 1–38, 1977.
- [23] J. Durbin, "The fitting of time series models," *Review of the International Statistical Institute*, vol. 28, pp. 233–243, 1960.
- [24] E. Ewada and O. Macchi, "Convergence of RLS and LMS adaptive filters," *IEEE Trans. on Circuits and Systems*, 34, no. 7, pp 799–803, July 1987.
- [25] E. Ewada, "Transient performance degradation of the LMS, RLS, sign, signed regressor, and sign-sign algorithms with data correlation," *IEEE Trans. on Circuits and Systems*, 46, no. 8, August 1999.
- [26] A. Ferrari and G. Alengrin, "AR Spectral analysis with randomly missing observations," *Proc. IEEE Workshop on Statistical Signal and Array Processing*, pp. 320–323, September 1998.
- [27] A. Fletcher, Sundeep Rangan and Vivek Goyal, "Estimation from Lossy Sensor Data : Jump Linear Modeling and Kalman Filtering," *The 3rd International Symposium on Information Processing in Sensor Networks*, California, USA, 2004.
- [28] B. Friedlander, "Lattice filters for adaptive processing," in *Proceedings of the IEEE*, vol.70, No. 8, pp. 829–867, 1982.
- [29] A. A. Giordano and F. M. Hsu, "Least squares estimation with applications to digital signal processing," *Wiley*, New York, 1985.
- [30] Y. Grenier, "Rational nonstationary spectra and their estimation," *1st IEEE ASSP workshop on spectral estimation*, Hamilton, Ontario, Canada 1981.
- [31] Y. Grenier, "Time-dependent ARMA modeling of nonstationary signals," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 31, pp. 899–911, 1983.
- [32] Y. Grenier, "Time varying lattices and autoregressive models : parameter estimation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1337–1340, Paris, France, May 1982.
-

-
- [33] A. Isaksson, "Identification of ARX Models Subject to Missing Data," *IEEE Trans. on Automatic Control*, vol. 38, pp. 813–819, 1993.
- [34] A. Isaksson, "A Recursive EM Algorithm for identification subject to missing data," in *IFAC Symposium on System Identification*, vol. 2, pp. 679–684, Copenhagen, Denmark, 1994.
- [35] F. Itakura and S. Saito, "A statistical method for estimation of speech spectral density and formant frequencies," in *Transactions of the Institute of Electronics and Communication Engineers of Japan*, vol. 53-A, pp. 36–43, 1970.
- [36] A. Janssen, R. Veldhuis and L. Vries, "Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 45, No. 6, 1997.
- [37] R. H. Jones, "Maximum likelihood fitting of ARMA models to time series with missing observations," *Technometrics*, vol. 22, pp. 389–395, August 1980.
- [38] D. Lee and M. Morf, "Recursive square-root ladder estimation algorithms," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1005–1017, Denver, USA, April 1980.
- [39] D. Lee and M. Morf and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 29, pp. 627–641, 1981.
- [40] N. Levinson, "The Wiener (Root Mean Square) Error Criterion in Filter Design and Prediction," *Journal of Mathematical Physics*, vol. 25, pp. 261–278, 1947.
- [41] B. Lindoff and J. Holst, "Bias and covariance of the recursive least squares estimator with exponential forgetting factor in vector autoregressions," *Journal of time series analysis.*, vol. 17, pp. 553–571, 1996.
- [42] B. Lindoff and J. Holst, "Convergence analysis of the RLS identification algorithm with exponential forgetting factor in stationary ARX-structures," *International Journal of Adaptive Control and Signal Processing.*, vol. 13, pp. 1–22, 1998.
- [43] O. Macchi and E. Ewada, "Compared speed and accuracy of the RLS and LMS algorithms with constant forgetting factors," *RAIRO-APII*, 22, no. 3, pp. 255–267, July 1988.
- [44] O. Macchi, "Le filtrage adaptatif transverse," *Traitement du Signal*, vol. 5, No. 3, pp. 115–132, 1988.
- [45] O. Macchi, "Adaptive processing : The least mean squares approach with applications in transmission," *Wiley*, 1995.
- [46] J. Makhoul, "Linear prediction : A tutorial review," *Proceedings of the IEEE*, vol. 63, No. 4, pp. 561–580, 1975.
- [47] J. Makhoul, "New lattice methods for linear prediction," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 462–465, Philadelphia, USA, April 1976.
- [48] J. Makhoul, "Stable and efficient lattice methods for linear prediction," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 25, No. 5, pp. 423–428, 1977.
- [49] J. Makhoul and R. Viswanathan, "Adaptive lattice methods for linear prediction," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 83–86, Oklahoma, USA, April 1978.
- [50] J. Makhoul and L. Cosell, "Adaptive lattice analysis of speech," *IEEE Trans. on circuits and systems*, vol. 28, pp. 494–499, 1981.
-

-
- [51] S. Mirsaidi and J. Oksman, "A class of real-time AR identification algorithms in the case of missing observations," in *Proc. 8th European Signal Processing Conference, EUSIPCO*, pp. 803–806, Trieste, Italy, 1996.
- [52] S. Mirsaidi, "Méthodes d'échantillonnage non périodique adaptatif," *Thèse de doctorat de l'Université Paris XI*, 1997.
- [53] S. Mirsaidi, G. Fleury and J. Oksman, "LMS Like AR modeling in the case of missing observations," *IEEE Trans. on Signal Processing*, vol. 45, No. 6, 1997.
- [54] S. Mirsaidi, G. Fleury and J. Oksman, "Reducing quantization error using prediction/non uniform transmission," *Proc. International Workshop on Sampling Theory and Applications*, Aveiro, Portugal, 1997.
- [55] S. Mirsaidi, G. Fleury and J. Oksman, "An ADPCM-like system based on non uniform signal transmission," *Proc. International Workshop on Sampling Theory and Applications*, Aveiro, Portugal, 1997.
- [56] T.K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine* Vol. 13, Issue 6, pp. 47 – 60, Nov. 1996.
- [57] E. Parzen, "On spectral analysis with missing observations and amplitude modulation," *Sankhya*, ser. 4, vol. 25, pp. 383–392, 1963.
- [58] B. Porat and B. Friedlander, "ARMA spectral estimation of time series with missing observations," *IEEE Trans. on Information Theory*, vol. 30, No. 6, 1984.
- [59] L. Rabiner and R. Schafer, "Digital processing of speech signals," Prentice Hall, 1978.
- [60] Y. Rosen and B. Porat, "The second order moments of the sample covariances for time series with missing observations," *IEEE Trans. on Information Theory*, vol. 35, No. 2, 1989.
- [61] Y. Rosen and B. Porat, "Optimal ARMA parameter estimation based on the sample covariances for data with missing observations," *IEEE Trans. on Information Theory*, vol. 35, No. 2, 1989.
- [62] R. Sanchis, R. Sala and P. Albertos, "Scarce data operating conditions : Process model identification," *IFAC Symposium on System Identification*, 1997.
- [63] R. Sanchis and P. Albertos, "Recursive identification under scarce measurements - convergence analysis," *Automatica*, 38, pp. 535–544, 2002.
- [64] L. Scharf, "Statistical Signal Processing : Detection, Estimation and Time Series Analysis," *Addison-Wesley*, USA, 1991.
- [65] C. Therrien, "The missing observations theorem and a new proof of Levinson's recursion," in *Proc. IEEE International Symposium Circuits and Systems*, pp. 2307–2308, Kobe, Japan, 2005.
- [66] R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 23, No. 3, pp. 309–321, 1975.
- [67] J. G. Wade, "Codage et traitement du signal," Masson, 1991.
- [68] R. Wallin, A. Isaksson and L. Ljung, "An iterative method for identification of ARX models from incomplete data," in *Proc. 39th IEEE Conference on Decision and Control*, pp. 203–208, Sydney, Australia, December 2000.
- [69] R. Wallin and A. Isaksson, "Multiple Optima in Identification of ARX Models Subject to Missing Data," *EURASIP Journal on Applied Signal Processing*, vol. 1, pp. 30–37, 2002.
- [70] E. Walter and L. Pronzato, "Identification de modèles paramétriques à partir de données expérimentales," *MASSON*, Paris, 1994.
-