Modèles de séries chronologiques Bilinéaires : Estimations, Algorithmes et Applications

Khadija Bouzaâchane

N^o d'ordre :

Année : 2006

Université Mohammed V-Agdal

Ecole Mohammadia d'Ingénieurs

THESE

Présentée

Pour l'obtention du grade de

Docteur en Sciences Appliquées

U.F.R : Analyse, Modélisation et Simulation des Systèmes

Discipline : Mathématiques Appliquées

Spécialité : Statistique et Informatique

Par

Khadija BOUZAACHANE

Modèles de Séries Chronologiques Bilinéaires : Estimations, Algorithmes et Applications

Soutenue le 16 Décembre 2006 devant le jury

Pr. A. Cheddadi	EMI-Rabat	Président
Pr. Y. Benghabrit	Directeur-adjoint, ENSAM-Meknès	Directeur de thèse
Pr. M. Harti	FS Dhar-Mahrez-Fès	Rapporteur et co-encadrant
Pr. A. Bouamain	ENSEM-Casablanca	Rapporteur
Pr. A. Skalli	EMI-Rabat	Rapporteur
Pr. J. Allal	FS-Oujeda	Examinateur
Pr. A. Akharif	FST-Tanger	Examinateur

Je dédie ce travail

à ma mère,

à ma soeur Malika,

à mes frères,

à mon futur époux El Mahdi,

à Aimad, Samia et Ines.

Remerciements

Cette thèse est le fruit de quelques années de travail effectuées au sein du Laboratoire d'Etudes et de Recherche en Mathématiques Appliquées (LERMA) à l'Ecole Mohammadia d'Ingénieurs. Même si elle porte le nom de son auteur, elle est aussi le résultat de la combinaison presque magique de diverses contributions. Et je voudrais ici remercier les nombreuses personnes qui, à des titres divers, ont participé à son élaboration.

Le bon déroulement de cette thèse, jusqu'à son heureux dénouement, sont en grande partie imputables à mon directeur de thèse le Professeur Youssef Benghabrit. Il a suggéré ce travail, l'a dirigé et accompagné me laissant une certaine liberté d'action. J'ai d'ailleurs beaucoup apprécié la confiance qu'il m'a faite. Je le remercie vivement et chaleureusement pour sa disponibilité, son dynamisme et sa patience.

Ces remerciements sont à partager avec Monsieur Mostafa Harti, Professeur à la FS de Fès, qui s'est impliqué de manière remarquable dans cette thèse et lui a donné sa dimension actuelle, grâce à ses conseils experts et ses relectures approfondies.

Merci également à Monsieur Abdelkhalek Cheddadi, Professeur à l'EMI, pour m'avoir honoré par sa présidence du jury et pour son aide remarquable.

Je suis extrêmement reconnaissante à Monsieur Abdelhalim Skalli, Professeur à l'EMI, et à Monsieur Abdelhalim Bouamain, Professeur à l'ENSEM, d'avoir accepté d'être les rapporteurs de cette thèse. Leurs commentaires et suggestions sur la version

préliminaire m'ont été très précieux.

Je suis très sensible à l'honneur que m'ont fait, Monsieur Jelloul Allal, Professeur à la FS d'Oujeda et Monsieur Abdelhadi Akharif, Professeur à la FST de Tanger en s'intéressant à mon travail et en acceptant de faire partie de mon jury de thèse.

Je tiens à exprimer ma gratitude à Madame le professur Rajae Aboulaïch et Monsieur le professeur Ali Souissi pour leur aide précieuse et continue et à Monsieur Rachid Ellaia, Chef de l'UFR "Annalyse, Modélisation et Simulation de systèmes", de m'avoir fait part de sa documentation personnelle.

C'est aussi l'occasion pour moi de remercier du fond du coeur ma très chère mère, ma soeur Malika et mes frères Mohammed, Ahmed, Driss, El Mostapha et Hicham pour leur soutien et leur affection sans limite. Ils ont constamment été à mes côtés et m'ont toujours encouragée à faire ce qui me plaisait. Merci aussi à mon futur époux El Mahdi qui partage ma vie. Sa présence m'est tout simplement indispensable.

Enfin, je remercie la famille El Guarmah, en particulier Jemiaâ et ma belle mère, la famille Jaberi, mes très chères amies Soumia Gourari, Hassania et Amal, et aussi Ikram, Wafae, Samira, Amarti Mohammed Soufiane et Meskine Driss.

Tables des matières

Liste des figures

Liste des tableaux

Résumé

Plusieurs travaux sur l'estimation des paramètres de certains modèles bilinéaires ont été réalisés avec quelques applications et simulations.
Notre objectif était alors, d'édifier un nouvel algorithme d'estimation des paramètres de ces modèles. Notre travail consiste en deux contributions aux modèles bilinéaires :

- Nouveaux algorithmes d'estimation des paramètres, du modèle bilinéaire super-diagonal d'ordre un, du modèle bilinéaire diagonal d'ordre un et du modèle bilinéaire diagonal pur d'ordre p.
- Conception d'un logiciel scientifique qui traite la simulation et l'estimation de ces modèles.

Nous avons développé notre nouvelle technique d'estimation en se basant sur la méthode du maximum de vraisemblance et l'algorithme du filtre de Kalman, ce dernier requiert une représentation en espace d'état de ces modèles. Nous avons pour cela utilisé la représentation affine en état polynomiale en entrée introduite par Guégan et la représentation de Hamilton.
Pour exhiber la bonne performance de notre approche nous avons généré plusieurs simulations. Notre algorithme d'estimation a été implémenté dans un outil logiciel, que nous avons conçu au profit de certains modèles bilinéaires particuliers.

Abstract

Some works in estimate of the parameters of the bilinear models have been realized with few applications and simulations.
Our goal was, then, to construct a novel algorithm of estimating of the parameters of these models. Our work consists of two contributions for bilinear models :

- New algorithms of estimating of the parameters of first-order superdiagonal bilinear model, first-order diagonal bilinear model and p-order pure diagonal bilinear model.
- The conception of scientific software, devoted to simulation and estimating in these models.

This new technique of estimating is based on maximum likelihood method and Kalman filter algorithm.
The utilization of Kalman filter algorithm requires the state space representation of bilinear models. So, we have employed the polynomial state affine representation, introduced by Guegan, and Hamilton's space state representation, for bilinear models.
To demonstrate the good performance of our approach we conducted a series of simulations. Our algorithm was implemented in a scientific software, devoted to a particular bilinear model.

Notations et Rappels

Nous introduisons ici des rappels et des notations communes aux différentes parties de ce manuscrit.

Notations

- E : Espérance mathématique.
- $cov(X, Y)$: Covariance de X et Y.
- $V(X)$: Variance de X.
- $\varrho(A)$: Rayon spectral de la matrice A.
- $\otimes$: Produit de Kronecker.
- $\| A \| = \max(\sum_{i=1}^{p} \mid a_{ij} \mid)$: Norme d'une matrice A.
- 0_m : Matrice nulle d'ordre m.
- $X \sim \mathcal{L}$: X suit la loi $\mathcal{L}$.
- VarE : Variance des erreurs.
- VarD : Variance des données.
- A' : Transposé de A.

- $(X|Z)$: X sachant (conditionnellemnt à) Z.

Rappels

Loi de Pearson VI (Devroye 1986)

Les fonctions de Pearson ont été créées pour représenter des distributions unimodales. Il en existe douze. Elles ont été inventées par Karl Pearson à la fin du XIX^e siècle et au début du XX^e siècle. ci-dessous nous donnons la définition de la loi de Pearson VI. La densité de probabilité f de Pearson VI, pour $x \in [a, \infty)$, est définie par :

$$f(x) = C(x-a)^b x^{-d} \qquad d > b+1 > 0, a > 0$$

où C est la constante de normalisation.

La loi gamma (Giard 1995)

La loi Gamma est définie pour $0 \leq x \leq \infty$ et se caractérise par deux paramètres positifs, le paramètre de forme α et le paramètre d'échelle β. Sa densité est donnée par :

$$f(x) = \frac{\left(\frac{x}{\beta}\right)^{\alpha-1} \exp^{-x/\beta}}{\int_0^\infty \exp^{-t} t^{\alpha-1} dt}$$

Loi bêta (Giard 1995)

Cette loi est définie pour $a \leq x \leq b$ et se caractérise par deux paramètres de forme positifs υ et ω. Sa densité est la suivante :

$$f(x) = \frac{(x-a)^{\upsilon-1}(b-x)^{\omega-1}}{\left[\int_0^1 (1-t)^{\omega-1} dt\right](b-a)^{\upsilon+\omega-1}}$$

Rayon spectral

Le Rayon spectral d'une matrice A est égal à $\sup_{i=1,\ldots,k}(\lambda_i)$, où les λ_i pour $i = 1, \ldots, k$

sont les valeurs propres de la matrice A.

Produit de Kronecker

Le produit de Kronecker (ou tensoriel) de deux matrices $A = (a_{ij})_{1 \leq i \leq m, 1 \leq i \leq n}$ et B est

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{pmatrix}$$

Biais

Soit θ le paramètre à estimer et T un estimateur, c'est-à-dire une fonction des X_i, où les X_i sont des variables aléatoires, à valeurs dans un domaine acceptable pour θ. La quantité $E[T] - \theta$ s'appelle le biais.

Les critères usuels de validation d'une méthode de prévision(Mélard (1990))

Supposons qu'on dispose de n prévisions $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$, qui correspondent aux données $y_1, y_2, \dots, y_n$, et donc de n erreurs de prévisions $e_1, \dots, e_n$. Les critères suivants sont utilisés pour juger de la validité de la méthode de prévision.

Erreur moyenne

$$\overline{e} = \frac{1}{n} \sum_{i=1}^{n} e_i$$

Variance

$$var = \frac{1}{n} \sum_{i=1}^{n} (e_i - \overline{e})^2$$

Ecart type ("standard deviation: std")

$$std = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i - \overline{e})^2}$$

Carré moyen des erreurs ("Mean Square Error")

$$MSE = \frac{1}{n}\sum_{i=1}^{n} e_i^2$$

Erreur quadratique moyenne ("Root Mean Square Error")

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}$$

Erreur absolue moyenne en pourcentage ("Mean Absolute Percentage Error")

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \frac{\mid e_i \mid}{y_i}$$

Remarque : il faut que les y_i soient positives.

T-statistic

$$T - statistic = \frac{biais}{RMSE}$$

Introduction générale

Historique

Récemment, plusieurs travaux qui traitent les modèles non-linéaires de séries chronologiques ont vu le jour, démontrant ainsi la limitation du champ d'application des modèles linéaires de séries chronologiques et le gain qu'on peut gagner en s'intéressant aux modèles non-linéaires. En effet, dans plusieurs disciplines l'existence de la non-linéarité entre variables est reconnue.

Parmi les modèles non-linéaires, on peut citer les modèles bilinéaires (BL), les modèles autorégressifs à seuil (TAR), les modèles autorégressifs exponentiels (EXPAR), les modèles ARCH...etc (voir Guégan (1994)).

Dans cette thèse nous nous sommes intéressés aux modèles bilinéaires. Ces modèles, à l'origine, ont été développés d'un point de vue déterministe (voir Weiner (1958)), et leurs applications ont été essentiellement orientées en automatique et théorie du contrôle (voir Molher (1970))...

Et c'est grâce à Granger et Anderson (1978) que la représentation stochastique de ces modèles a été introduite pour modéliser des séries économiques. À la suite de leur travail et de celui de Subba Rao (1981), Guégan s'est intéressée à cette présentation stochastique et a étudié les propriétés probabilistes des cas particuliers des modèles bilinéaires à cause de leur complexité structurale.

En (1981), Guégan a étudié l'existence, l'inversibilité et l'expression de la fonction de vraisemblance d'un modèle bilinéaire superdiagonal d'ordre un. Ensuite, elle a abordé le problème de l'ergodicité pour des cas particuliers en (1983) et (1984).

À cette époque des théorèmes plus généraux sur l'existence des modèles bilinéaires, la fonction de covariance et les moments d'ordre supérieur ont vu le jour par Bahskara Rao, Subba Rao et walker en (1983). Cependant l'étude statistique de ces modèles reste peu développée, par manque de résultats constitutifs comme l'inversibilité ou l'existence de certains moments. Néanmoins, Pham et Tran en (1981) ont fourni des estimateurs consistants, pour les paramètres de certains modèles particuliers, par la méthode des moindres carrés, et Guégan en (1984) a estimé par la méthode des moments les paramètres de certains modèles.

L'année (1985) représente un tournant essentiel dans l'étude des modèles bilinéaires et une nouvelle orientation se dessine avec la représentation markovienne. En effet, Pham fut le pionnier de la représentation markovienne des modèles superdiagonaux. Il établit les conditions d'existence, d'unicité et de minimalité de cette représentation, ce qui lui a permis d'obtenir l'expression de la fonction de covariance et des moments d'ordre supérieur des modèles bilinéaires correspondants.

En (1986), Guégan a montré que cette représentation s'étend difficilement au cas des modèles sousdiagonaux et elle a introduit, en revanche, la forme stochastique des modèles polynomiaux en entrées affines en état définis par Sontag (1979).

C'est grâce à l'approche markovienne que l'étude statistique des modèles bilinéaires a connu un certain développement. En effet, Guégan et Pham (1987 a) ont obtenu l'inversibilié de la classe des modèles bilinéaires diagonaux, et en (1990) Liu a obtenu des conditions d'inversibilité pour les modèles bilinéaires généraux. Or, la notion d'inversibilié est indispensable pour espérer développer certaines propriétés statistiques des modèles bilinéaires, comme l'estimation qui n'a été abordée pertinemment, que pour des cas particuliers des modèles en question.

L'estimation par maximum de vraisemblance a été abordée par Subba Rao et Gabr en (1984), pour la classe des modèles superdiagonaux. En (1987 b) Guégan et Pham ont obtenu pour l'ensemble des paramètres des modèles diagonaux, des estimateurs consistants par la méthode des moindres carrés. En (1990), Kim et Billard ont

proposé des estimateurs des paramètres du modèle bilinéaire diagonal d'ordre un par la méthode de la fonction d'autocovariance.
Les tests, aussi, ont été considérés avec un grand intérêt. Cet outil statistique est le mieux adapté pour la discrimination entre les modèles linéaires et bilinéaires. Parmi les tests proposés, on cite : Guégan et Pham (1992), Wandji (1995), Benghabrit et Hallin (1992, 1996,1998), Houfaidi et Benghabrit (1998 b).

Malgré ce panorama de travaux, l'étude des modèles bilinéaires n'a pas encore atteint son apogée.

Objectif de cette thèse

Notre travail s'insère dans le cadre de l'estimation des paramètres des modèles bilinéaires, que leur complexité nous a poussé à nous intéresser à des cas particuliers de ces modèles.
Nous proposons une nouvelle technique d'estimation fondée sur la méthode du maximum de vraisemblance et l'algorithme du filtre de Kalman (Hamilton (1994)).
Cette technique a un double objectif : premièrement, réécrire les modèles bilinéaires, sous forme d'espace d'état, et deuxièmement, expliciter l'expression de leur fonction log-vraisemblance qui sera calculée par l'algorithme du filtre de Kalman.
L'esprit de cette technique s'inspire des travaux fondateurs de cette approche dans le cas des modèles ARMA. On cite, Pearlman (1980), Anshely et Kohn (1983), Shea (1987). D'autres travaux adoptent l'esprit de cette méthode en remplaçant le filtre de Kalman par les équations de Chandrasekhar, lorsque les données sont toutes disponibles, comme celui de Mélard (1983) , Shea (1989) et Harti (1996).
En se basant sur cette approche nous avons conçu des algorithmes d'estimations des paramètres des modèles bilinéaires, objet de cette étude, en adoptant différentes méthodes de maximisation de la fonction log-vraisemblance.

Les performances de ces algorithmes ont été validées à travers des simulations de Monte Carlo et des applications à des données réelles.
Le développement des logiciels statistiques est une tâche difficile, qui requiert la définition d'un ensemble d'utilitaires et d'interfaces graphiques pour faire le traitement souhaité. En pratique, il existe un ensemble de logiciels qui facilitent un certain nombre de traitements à caractère statistique et analyse des données. On peut citer SYSTAT, SPSS, SAS, S+, etc.
D'autres types de solutions plus spécialisées peuvent être envisagées, comme les applications indépendantes qui se résument à la programmation d'une application spécifique pour la résolution d'un problème défini.
Au cours de notre travail nous avons remarqué la rareté des logiciels qui traitent les modèles bilinéaires. D'où notre initiative de concevoir un logiciel spécifique aux modèles bilinéaires.
La première version de ce logiciel implémente l'algorithme d'estimation des paramètres des modèles bilinéaires en question et permet aussi de les simuler.

Présentation du mémoire

Ce manuscrit s'articule autour de quatre chapitres :
Le **premier chapitre** dresse un panorama de concepts utilisés tout au long de ce mémoire. Il débute par des généralités sur les processus stochastiques et la définition des modèles ARMA et leurs propriétés probabilistes constituées de l'inversibilité et de la stationnarité. L'introduction de ces modèles dans ce chapitre a comme objectif, de montrer un résultat de modélisation, par ces modèles, des données de fiabilité des logiciels exposées au chapitre deux.
Ensuite il présente la définition du modèle bilinéaire général, et ses propriétés d'existence, de stationnarité et d'inversibilité. L'objectif de cette thèse est de construire des algorithmes d'estimation des paramètres de modèles bilinéaires particuliers.

Cet algorithme utilise le filtre de Kalman pour évaluer la fonction de vraisemblance. Pour cela, nous présentons une description de l'algorithme du filtre de Kalman et avant d'évoquer l'esprit de notre approche d'estimation et l'expression de la log-vraisemblance d'un modèle bilinéaire, nous donnons un état d'art des méthodes d'estimation des paramètres des modèles bilinéaires. Lors de la procédure d'estimation nous avons besoin de maximiser la fonction de la log-vraisemblance par rapport aux paramètres à estimer. Comme notre fonction est compliquée à dériver, nous avons utilisé des méthodes d'optimisation sans dérivées. Dans ce chapitre nous décrivons succinctement et uniquement, les méthodes d'optimisation déterministes sans dérivées et les méthodes stochastiques, que nous avons employées dans ce mémoire.
Notre travail contient aussi une application d'un modèle bilinéaire particulier à la fiabilité des logiciels. Pour cette raison, nous donnons à la fin de ce chapitre un petit aperçu sur ce domaine.

Dans le **chapitre deux**, nous abordons le problème d'estimation pour le modèle bilinéaire superdiagonal d'ordre deux. Nous allons entamer la première partie de ce chapitre par la définition du modèle, la présentation des conditions de stationnarité et d'inversibilité, et nous construisons l'espace d'état de ce modèle à travers lequel nous donnons l'expression de la log-vraisemblance. Ensuite, nous décrivons l'algorithme d'estimation des paramètres de ce modèle.
Pour justifier la bonne performance de cet algorithme nous avons mené des séries de simulation dont les résultats seront exhibés à la fin de cette première partie.
La deuxième partie de ce chapitre est consacrée à la modélisation des données issues d'un projet réel de réalisation des logiciels. Tout d'abord, nous présentons les résultats de la modélisation par les modèles ARMA. Ensuite nous évoquons les résultats de l'application du modèle bilinéaire, objet de ce chapitre, aux données de fiabilité des logiciels. Avant d'ajuster le modèle à ces données, nous utilisons notre algorithme pour estimer ces paramètres. Et pour justifier la qualité prédictive de notre modèle,

nous le comparons avec celui de Chen et Singpurwalla (1994).

Le **chapitre trois** de ce mémoire, est consacré à l'estimation des paramètres d'un modèle bilinéaire diagonal d'ordre un. La même structure du chapitre deux sera adoptée dans ce chapitre, c'est-à-dire, la définition du modèle, la présentation de ces propriétés probabilistes et sa forme d'état. L'algorithme d'estimation de ce modèle est le même que celui du modèle défini dans le chapitre deux, nous nous contentons dans ce chapitre de dresser les résultats des simulations faites pour montrer la bonne performance de notre algorithme pour le modèle en question.
Une application du modèle bilinéaire diagonal d'ordre un aux données issues du tremblement de terre d'Al-Hoceima, a été réalisée, nous évoquons à la fin de ce chapitre les résultats numériques de cette étude.

Le **chapitre quatre**, aborde le problème d'estimation des paramètres du modèle bilinéaire diagonal pur d'ordre p. Pour ce modèle, nous avons adopté une autre présentation en espace d'état différente de celle des modèles objets des chapitres deux et trois. Nous avons aussi utilisé une méthode stochastique qui est le recuit simulé pour maximiser le log-vraisemblance de ce modèle au lieu de la méthode déterministe de Powell utilisée aux chapitres deux et trois.
Après la description de l'algorithme d'estimation des paramètres de ce modèle, nous exposons les résultats de simulations de cet algorithme, et nous présentons les différentes fonctionnalités du logiciel que nous avons conçu au profit des modèles bilinéaires objets de cette thèse. Ce logiciel se présente sous forme d'une application développée avec le langage Java. Actuellement, la plupart des logiciel existants abordent les modèles linéaires et les modèles non-linéaires ARCH et GARCH de séries chronologiques. Notre application présente l'avantage de permettre de simuler aisément certains modèles bilinéaires et d'estimer leurs paramètres.
Notre travail se termine par une **conclusion et des perspectives**.

Chapitre 1

Généralités

1.1 Introduction

Dans ce chapitre nous focalisons notre intérêt sur la présentation de l'aspect général des modèles et des méthodes que nous utiliserons dans les chapitres suivants. Ce préliminaire débutera par des généralités sur les processus stochastiques. Nous l'enchaînons ensuite par, la définition et quelques propriétés des modèles Auto-Régressifs Moyennes Mobiles (ARMA), et des généralités sur les modèles bilinéaires (BL). La présentation de certains cas particuliers de ces modèles se fera ultérieurement. La conception d'un algorithme d'estimation des paramètres des modèles bilinéaires est parmi les objectifs de ce travail. Cet algorithme se base sur la méthode du maximum de vraisemeblance et le filtre de Kalman. De ce fait, nous présenterons dans les paragraphes 4 et 5, respectivement, une description détaillée de l'algorithme du filtre de Kalman, et un état d'art sur les méthodes d'estimations des modèles bilinéaires. La puissance de notre algorithme d'estimation a été concrétisée via une étude par simulation et par des applications à des données réelles issues de la fiabilité des logiciels et des signaux séismiques. Nous donnerons dans le dernier paragraphe, une description succincte de la fiabilité des logiciels.

1.2 Généralités sur les processus stochastiques univariés : Rappels

Pour modéliser une collection d'observations, on cherche le modèle qui approxime adéquatement le vrai processus aléatoire sous-jacent à ces observations. Pour mener cette étude on a besoin de processus stochastiques. Ainsi, nous pensons indispensable de débuter ce chapitre par quelques définitions propres aux processus stochastiques univariés, que nous utiliserons dans la suite.

Définition 1 *Un processus stochastique est une famille de variables aléatoires définies sur un certain espace de probabilité et indexées par un certain ensemble $\mathcal{T}$ (ensemble de réelles ou des entiers relatifs) appelé ensemble de temps.*

Définition 2 *Un processus stochastique à temps discret est une suite de variables aléatoires réelles $(X_t, t \in \mathbb{Z})$. La loi d'un tel processus est caractérisée par les lois de toute sous famille finie $X_{t_1}, \ldots, X_{t_n}, n \in \mathbb{N}^*, t_1, \ldots, t_n \in \mathbb{Z}$ (théorème de Kolmogorov).*

Définition 3 *Le processus $(X_t, t \in \mathbb{Z})$ est dit strictement stationnaire, si pour toute suite finie d'instants $t_1, \ldots, t_k$, et pour tout entier t, la loi conjointe de $X_{t_1+t}, \ldots, X_{t_k+t}$, ne dépend pas de t.*

Définition 4 *Un processus $(X_t, t \in \mathbb{Z})$ est stationnaire du second ordre, si :*
$\diamondsuit$ $\forall t \in \mathbb{Z}$, $E[X_t^2] < \infty$,
$\diamondsuit$ $\forall t \in \mathbb{Z}$, $E[X_t] = \mu$ *(indépendant de t),*
$\diamondsuit$ $\forall t \in \mathbb{Z}, \forall h \in \mathbb{Z}$, $cov(X_t, X_{t+h}) = \gamma(h)$ *(indépendant de t).*

Exemple :

Un processus stationnaire fort utile est celui que l'on appelle bruit blanc; un tel processus est une suite de variable aléatoires $(e_t, t \in \mathbb{Z})$, d'espérances mathématiques nulle $(E[e_t] = 0)$, non corrélées ($\gamma(h) = 0$ si $h \neq 0$) et de même variance $(V[e_t] = \sigma^2 = \gamma(0))$.

Définition 5 *Un processus stationnaire est dit ergodique si toutes ses caractéristiques peuvent être déterminées à partir d'une seule trajectoire de ce processus : en pratique ceci veut dire qu'une trajectoire observée pendant une durée assez longue suffit à calculer l'espérance du processus X_t. Ainsi pour l'espérance : on calculera la moyenne temporelle(stochastique) sur un intervalle de durée h :*

$$\frac{1}{h}\int_{-h/2}^{h/2} X_t dt = M_h.$$

Si le processus est ergodique $M_h \to \mu = E[X_t]$ si $h \to \infty$.

Définition 6 *Un processus stochastique stationnaire $(X_t, t \in \mathbb{Z})$ est inversible si et seulement si il existe une suite $\tilde{e}_n$ fonction de $X_1, \ldots, X_n$ uniquement telle que $\tilde{e}_n - e_n \longrightarrow 0$ en probabilité quand $n \longrightarrow \infty$. $(e_t, t \in \mathbb{Z})$ est une suite de variables aléatoires indépendantes identiquement distribuées de moyenne nulle et de variance finie.*

Définition 7 *La fonction ρ définie sur $\mathbb{Z}$ qui à h fait correspondre :*

$$\rho(h) = \frac{cov(X_t, X_{t+h})}{\sqrt{V(X_t)}\sqrt{V(X_{t+h})}} = \frac{\gamma(h)}{\gamma(0)}$$

est appelée fonction d'autocorrélation du processus et son graphe est appelé corrélogramme.

Définition 8 *$(X_t, t \in \mathbb{Z})$ étant un processus stationnaire, on appelle fonction d'autocorrélation partielle la fonction :*

$$r(h) = \frac{cov(X_t - X_t^*, X_{t-h} - X_{t-h}^*)}{V(X_t - X_t^*)} \qquad \forall h > 0$$

où X_t^* (respectivement X_{t-h}^*) est la régression affine de X_t (respectivement X_{t-h}) sur $X_{t-1}, \ldots, X_{t-h+1}$

$r(h)$ est le coefficient de régression de $X_t - X_t^*$ sur $X_{t-h} - X_{t-h}^*$. D'après la propriété de Frisch et Waugh (Droesbeke et al 1989) il est égal au coefficient α_h de X_{t-h} dans la

régression de X_t sur $1, X_{t-1}, \ldots, X_{t-h}$:

$$X_t = \alpha_0 + \sum_{i=1}^{h} \alpha_i X_{t-i} + v_t$$

où v_t est tel que, $E[v_t] = 0$ et $E[v_t X_{t-i}] = 0$ pour $i = 1, \ldots, h$.

Dans plusieurs domaines on est confronté à des phénomènes évoluant dans le temps qu'on désire décrire, analyser, contrôler ou prévoir. L'économie, la finance, la gestion, la météorologie, la démographie, la fiabilité des systèmes, les réseaux de télécommunications...etc, constituent quelques exemples de ces domaines.
La suite de valeurs numériques correspondant aux réalisations d'un phénomène observé au cours du temps, s'appelle série chronologique, chronique ou temporelle. La façon la plus appropriée d'étudier une série chronologique est de la considérer comme la réalisation d'un processus stochastique.
La première étape de l'étude d'une série chronologique est la modélisation. Il s'agit de choisir le modèle qui s'ajuste adéquatement à la série qu'on veut étudier. Dans la pratique on trouve plusieurs modèles de séries chronologiques nous citons : modèles AutoRégressifs Moyennes Mobiles (ARMA), modèles Bilinéaires (BL), les modèles ARCH, les modèles GARCH,...
Nous consacrerons les deux paragraphes suivants aux modèles autorégressifs moyennes mobiles et aux modèles bilinéaires.

1.3 Modèles Autorégressifs Moyennes Mobiles : ARMA

La plus grande partie des travaux traitant de l'analyse des séries chronologiques concernait essentiellement les modèles linéaires. Le livre de Box et Jenkins (1970) et les travaux de Akaike (1969) suffisent à démontrer la maturité atteinte dans la modélisation

des séries chronologiques linéaires.

Ce chapitre se place dans une perspective assez générale et propose quelques concepts relatifs aux modèles Autorégressifs Moyennes Mobiles des séries chronologiques linéaires : définitions, stationnarité et inversibilité.

Définition 9 *L'opérateur retard B est un opérateur linéaire qui agit sur l'indice de temps en le décalant vers l'arrière, c'est-à-dire*

$$BY_t = Y_{t-1}, \qquad B^k Y_t = Y_{t-k}$$

Définition 10 *On appelle processus autorégressif moyenne mobile d'ordre p et q, noté ARMA(p,q) un processus stationnaire $(X_t; t \in \mathbb{Z})$ vérifiant une relation du type*

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \qquad \forall t \in \mathbb{Z}, \tag{1.1}$$

où les φ_i, θ_j sont des réels et $(e_t, t \in \mathbb{Z})$ est un bruit blanc de variance σ^2. La relation (1.1) peut aussi s'écrire

$$(1 - \sum_{i=1}^{p} \varphi_i B^i) X_t = (1 - \sum_{j=1}^{q} \theta_j B^j) e_t$$

ou

$$\Phi(B) X_t = \Theta(B) e_t$$

1.3.1 Stationnarité du Processus ARMA

Théorème 1 *(Gouriéroux et Monfort (1983))*
Le processus $(X_t, t \in \mathbb{Z})$ est stationnaire et admet la décomposition MA(∞)

$$X_t = \sum_{i=1}^{\infty} \zeta_i e_{t-i} \quad avec \quad \sum_{i=1}^{\infty} \mid \zeta_i \mid < +\infty$$

si et seulement si les racines de l'équation $\Phi(z) = 0$ *sont de modules supérieurs strictement à* 1 *c'est-à-dire*

$$\Phi(z) = 0 \Rightarrow \mid z \mid > 1$$

1.3.2 Inversibilité du Processus ARMA

Théorème 2 *(Gouriéroux et Monfort (1983))*
Le processus $(X_t, t \in \mathbb{Z})$ *est inversible, ce qui s'exprime par*

$$X_t = -\sum_{i=1}^{\infty} \pi_i X_{t-i} + e_t \quad avec \quad \sum_{i=1}^{\infty} \mid \pi_i \mid < +\infty$$

si et seulement si les racines de l'équation $\Theta(z) = 0$ *sont de modules supérieurs strictement à* 1, *c'est-à-dire*

$$\Theta(z) = 0 \Rightarrow \mid z \mid > 1$$

En pratique, on rencontre des séries chronologiques non-stationnaires. La non-stationnarité peut provenir de la variation de la moyenne, ou de la variance, ou bien des deux, dans le temps.
Lorsque la non-stationnarité est conséquente à la variation de la moyenne dans le temps, on procède à des différences successives de la série en question.

Définition 11 *Soit* $\bigtriangledown^d X_t = (1 - B)^d X_t$ *la différence d'ordre d de* X_t. *On appelle un processus ARMA intégré et on le note ARIMA*(p, d, q), *le processus* $(X_t, t \in \mathbb{Z})$ *tel que*

$$\Phi(B) \bigtriangledown^d X_t = \Theta(B) e_t$$

Les techniques standards d'analyse des séries chronologiques ont longtemps reposé sur la propriété fondamentale de linéarité. Cependant, de nombreuses recherches ont démontré que l'hypothèse de linéarité n'était qu'un pis-aller utopique apportant un confort appréciable dans l'étude probabiliste et statistique du modèle (Guégan (1993)). Pour pallier cette absence de linéarité, la classe des modèles bilinéaires a reçu une

attention particulière dans la littérature probabiliste et statistique depuis le travail précurseur de Granger et Andersen (1978a). Ces modèles, qui sont une extension des modèles ARMA, ont la particularité d'être linéaires par rapport à chacune des variables (X_t, e_t), lorsqu'on considère l'autre constante. Dans le paragraphe suivant, nous présentons la définition du modèle bilinéaire général et les propriétés probabilistes associées.

1.4 Modèles Bilinéaires

Les modèles bilinéaires ont vu le jour grâce à Granger et Anderson (1978a), qui à partir de la démarche des automaticiens, qui considèrent des modèles bilinéaires à entrée déterministe, ont pu par analogie, réalisés des modèles bilinéaires dont l'entrée est une suite de variables aléatoires indépendantes équidistribuées supposées généralement de lois gaussiennes. Ces modèles se définissent par :

Définition 12 *Un modèle bilinéaire général d'ordre (p,q,P,Q) noté BL(p,q,P,Q) est une équation aux différences stochastiques non linéaire de la forme*

$$X_t = \sum_{i=1}^{p} a_i X_{t-i} + \sum_{l=1}^{q} c_l e_{t-l} + \sum_{k=1}^{P}\sum_{j=1}^{Q} b_{kj} X_{t-k} e_{t-j} + e_t \tag{1.2}$$

où X_t est un processus stochastique, $(a_i, 1 \leq i \leq p)$, $(c_l, 1 \leq l \leq q)$ et $(b_{kj}, 1 \leq k \leq P, 1 \leq j \leq Q)$ sont des constantes et où $(e_t, t \in \mathbb{Z})$ est un bruit blanc.

Définition 13 *Un modèle bilinéaire est dit*

Superdiagonal si $b_{kj} = 0$ pour $k < j$,

Sousdiagonal si $b_{kj} = 0$ pour $k > j$,

Diagonal si $b_{kj} = 0$ pour $k \neq j$.

Depuis leur apparition et malgré leur complexité, les modèles bilinéaires ont été amplement développés en utilisant plusieurs approches : les choas de wiener, approche directe, représentation markovienne et la représentation affine d'état. Un grand intérêt a été concentré sur l'existence de tels modèles stationnaires, l'inversibilité, l'estimation des paramètres et les tests de linéarité. Cependant, malgré l'ampleur des travaux sur ces questions, elles n'ont été abordées qu'à partir des modèles particuliers, vu la dureté d'extension des résultats pour le modèle général (1.2).

Présentement, nous allons exposer le résultat de Liu et Brockwell (1988), concernant la propriété probabiliste existence et stationnarité du modèle bilinéaire(1.2). Ce résultat généralise celui de Akamanan, Bhaskara Rao et Subramanyian (1986) et de Bhaskara Rao, Subba Rao et Walker (1983), et il se base sur une représentation d'état du modèle (1.2).

On utilise les notations suivantes :
$\mathcal{X} = [X_t, X_{t-1}, \ldots, X_{t-p+1}]'$, $\tilde{e}_t = [e_t, e_{t-1}, \ldots, e_{t-q}]'$, avec $p = \max(p, P)$, A est la matrice d'ordre $(p \times p)$,

$$A = \begin{bmatrix} a_1 & a_2 & \ldots & a_p & 0 & \ldots & 0 \\ 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & 0 & 1 & 0 \end{bmatrix}$$

B_j, les matrices d'ordre $(p \times p)$, $j = 1, \ldots, Q$,

$$B_j = \begin{bmatrix} b_{1j} & b_{2j} & \ldots & b_{pj} & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \end{bmatrix}$$

et C la matrice d'ordre $(p \times (q+1))$,

$$C = \begin{bmatrix} c_0 & c_1 & \dots & c_q \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Alors (1.2) se réécrit

$$\mathcal{X}_t = A\mathcal{X}_{t-1} + C\tilde{e}_t \sum_{j=1}^{Q} B_j \mathcal{X}_{t-1} e_{t-j}. \tag{1.3}$$

1.4.1 Stationnarité du processus Bilinéaire

Théorème 3 *(liu et Brockwell, 1988)*
Soit le processus $(\mathcal{X}, t \in \mathbb{Z})$ défini par (1.3), où les matrices A, B_j et C ont été définies précédemment. On suppose de plus que $E(e_t^{2q}) < \infty$ et $E(e_t^s) = 0$ pour tout $s < q$. Posons :

$$\Gamma = \begin{bmatrix} A \otimes A + \sigma^2 B_1 \otimes B_1 & A \otimes B_2 + A \otimes B_2 & B_2 \otimes B_2 \\ \sigma^2(A \otimes B_1 + A \otimes B_1) & \sigma^2(B_2 \otimes B_1 + B_1 \otimes B_2) & 0 \\ \sigma^2(A \otimes A + \gamma^4 B_1 \otimes B_1) & \sigma^2(A \otimes B_2 + A \otimes B_2) & \sigma^2(B_2 \otimes B_2) \end{bmatrix}$$

Alors si

$$\lambda = \varrho(\Gamma) < 1, \tag{1.4}$$

(i) L'équation (1.3) a une solution strictement stationnaire $\mathcal{X}_t$, dont la première composante X_t est une solution stationnaire et ergodique de (1.2).
(ii) La solution unique de (1.2) est la première composante de

$$\mathcal{X}_t = C\tilde{e}_t + \sum_{n=1}^{\infty} \prod_{j=1}^{n} \Phi(t-j) C\tilde{e}_{t-n}$$

où

$$\Phi(t) = A + B_1 e_{t-1} + \ldots + B_Q e_{t-Q}, \quad t = \pm 1, \ldots .$$

Pour la démonstration on peut consulter le livre de Guégan ((1988), page 94). Dans le théorème $\otimes$ désigne le produit de Kronecker et ϱ le rayon spectral (voir Notations et Rappels). Ces résultats permettent de retrouver ceux établis par pham et tran (1981) et Gugan (1981) pour les modèles BL(1,0,1,1) et BL(0,0,2,1) respectivement, que nous allons voir en détail dans les chapitres suivants.

1.4.2 Inversibilité du processus Bilinéaire

Un concept crucial pour les modèles bilinéaires, est l'inversibilité, qui joue un rôle fondamental dans la prédiction et l'estimation.
Considérons le modèle général BL(p,q,P,Q) défini par (1.2), et soit les matrices $B(t)$, d'ordre $(p \times p)$, associées,

$$B(t) = \begin{bmatrix} a_1 + \sum_{j=1}^{p} b_{1j} X_{t-j} & a_2 + \sum_{j=1}^{p} b_{2j} X_{t-j} & \ldots & a_p + \sum_{j=1}^{p} b_{pj} X_{t-j} \\ 1 & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

À partir de la définition de l'inversibilité faible (Définition 6), Liu a obtenu des conditions d'inversibilité pour les processus bilinéaires généraux. Alors on déduit du théorème précédent le résultat suivant

Proposition 1 *(Liu, (1990 b))*
Supposons qu'un processus $(X_t, t \in \mathbb{Z})$ *vérifie les conditions du théorème* (3) *et la condition* $E(Log \mid X_1 \mid) < \infty$. *Alors si,*

$$E(\log \parallel \prod_{j=1}^{p} B(t-j) \parallel) < 0,$$

le processus vérifiant (1.2) *est inversible.*

Dans les chapitres suivants, nous donnerons les condtions d'inversibilité pour les modèles particuliers BL(0,0,2,1), BL(1,0,1,1) et BL(0,0,p,p).
Comme nous l'avons mentionné auparavant, l'étude des modèles bilinéaires a été abordée en utilisant maintes approches. La représentation Affine en état en fut une. Cette représentation a été introduite par Guégan (1986,1987), en s'inspirant du travail de Sontag (1979). L'existence de cette représentation a été établie par Guégan (1987). On l'appelle aussi modèles polynômiaux en les entrées, affines en l'état, car les coefficients se représentent soit par des matrices polynômiales en e_t, soit des polynômes en e_t.

Définition 14 *Le processus stochastique* $(\xi_t, t \in \mathbb{Z})$ *à temps discret gouverné par le modèle affine en l'état polynomial en l'entrée est défini par*

$$\begin{cases} \xi_{t+1} &= A(e_t)\xi_t + C(e_t) \\ X_t &= H(e_t)\xi_t + B(e_t), \end{cases} \tag{1.5}$$

où X_t *représente les observations du processus au temps* t*, et où :*

- *(H1)* ξ_t *est l'état du système.* ξ_1 *est donnée et est indépendante de* $A(e_t)$*,* $B(e_t)$ *pour* $t = 1, 2, \ldots$
- *(H2)* $(e_t, t \in \mathbb{Z})$*, est une séquence de variables aléatoires indépendantes identiquement distribuées, centrées de variance* σ^2 *finie.*
- *(H3)* A, C, H *sont, matrice carrée* $(n \times n)$*, vecteur colonne* $(n \times 1)$ *, vecteur ligne* $(1 \times n)$*. Ils sont polynomiaux en* e_t *de degré fini.*
- *(H4)* $H(e_t)$ *est un polynôme en* e_t *de degré fini.*

Cette représentation d'état va nous permettre d'utiliser le filtre de Kalman pour le calcul de la log-vraisemlance de modèles bilinéaires particuliers. Dans le paragraphe

suivant, nous allons tout d'abord décrire comment un système dynamique peut être exprimé en un espace d'état pour qu'il soit analysé en utilsant le filtre de Kalman. Ensuite nous décrivons les étapes de l'algorithme du filtre de Kalman pour l'espace d'état avec des paramètres variant stochastiquement.

1.5 Filtre de Kalman

Kalman (1960,1963) fut le pionnier de l'algorithme appelé filtre de Kalman. Son idée était d'exprimer un système dynamique par une représentation espace d'état et de mettre à jour d'une manière séquentielle la projection linéaire de ce système. Parmi les bénéfices que fournit l'algorithme, on cite le calcul de la log-vraisemblance du processus ARMA.

1.5.1 Présentation espace d'état

Soit X_t un $(N \times 1)$ vecteur de variables observées et ξ_t un $(r \times 1)$ vecteur non-observable appelé vecteur d'état. Hamilton (1994) a donné la représentation espace d'état de X_t, qui est définie par le système d'équations suivant :

$$\begin{cases} \xi_{t+1} &= F\xi_t + v_{t+1} & \text{: équation d'état} \\ X_t &= H\xi_t + A'y_t + w_t. \quad t \geq 0 & \text{: équation d'obseration} \end{cases} \tag{1.6}$$

où F, A' et H sont des matrices de paramètres de dimensions $(r \times r)$, $(N \times k)$ et $(N \times r)$ respectivement et y_t est $(k \times 1)$ vecteur de variables exogènes ou prédéterminées.
v_t et w_t sont deux vecteurs de bruit blanc de dimension $(r \times 1)$ et $(N \times 1)$ respectivement, non corrélés avec ξ_t, ils vérifient

$$E[v_t v'_\tau] = \begin{cases} Q & \text{pour} \quad t = \tau \\ 0_m & \text{ailleurs} \end{cases}$$

$$E[w_t w'_\tau] = \begin{cases} R & \text{pour} \quad t = \tau \\ 0_m & \text{ailleurs} \end{cases}$$

où Q et R sont des matrices de dimension $(r \times r)$ et $(N \times N)$ respectivement. v_t et w_t sont assumés non-corrélés

$$E[v_t w'_\tau] = 0 \quad \text{pour tout} \quad t \quad \text{et} \quad \tau.$$

y_t peut contenir des valeurs du vecteur X_t.

1.5.2 Présentation espace d'état avec paramètres stochastiques

Le modèle espace d'état avec paramètres stochastiques donné par Hamilton (1994), est défini par

$$\begin{cases} \xi_{t+1} &= F(y_t)\xi_t + v_{t+1} \\ X_t &= H(y_t)\xi_t + a(y_t) + w_t \end{cases} \tag{1.7}$$

où $F(y_t)$ désigne une matrice $(r \times r)$, $a(y_t)$ un vecteur $(N \times 1)$ et $H(y_t)$ une matrice $(N \times r)$, dont les éléments sont fonctions de y_t. Cette présentation espace d'état généralise celle donnée précédemment (1.6).

On assume que, conditionnellement à y_t et aux données $\mathcal{X}_{t-1} \equiv (X'_{t-1}, X'_{t-2}, \ldots, X'_1, y'_{t-1}, y'_{t-2}, \ldots, y'_1)$ observées jusqu'à $t-1$, le vecteur (v'_{t+1}, w'_{t+1}) a une distribution Gaussienne de moyenne nulle et de matrice variance-covariance suivante :

$$\begin{bmatrix} Q(y_t) & 0 \\ 0 & R(y_t) \end{bmatrix}.$$

On en déduit que $\begin{bmatrix} \xi_t \\ X_t \end{bmatrix} \Big| y_t, \mathcal{X}_t$ est gaussien. Maintenant, nous allons décrire l'algorithme du filtre de Kalman. On présume que $x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N$ ont été observées. L'algorithme débute par le calcul de $\xi_{1|0}$ qui dénote la prévision de $\hat{\xi}_1$, et qui n'est autre que l'espérance de ξ_1

$$\hat{\xi}_{1|0} = E(\xi_1),$$

le carré moyen des erreurs comises est

$$P_{1|0} = E([\xi_1 - \hat{\xi}_{1|0}][\xi_1 - \hat{\xi}_{1|0}]').$$

En partant de ces valeurs initiales, la deuxième étape est de calculer $\hat{\xi}_{2|1}$ et $P_{2|1}$. La procédure de calcul pour $t = 2, 3, \ldots, N$ a la même forme basique. C'est pourquoi nous restreingons la description de l'algorithme à l'étape t.
À l'étape t en donnant $\hat{\xi}_{t|t-1}$ et $P_{t|t-1}$, l'objectif sera de calculer $\hat{\xi}_{t+1|t}$ et $P_{t+1|t}$ à partir des deux équations suivantes

$$\hat{\xi}_{t+1|t} = F(y_t)\hat{\xi}_{t|t-1} + K_t H'(y_t)(X_t - a(y_t) - H(y_t)\hat{\xi}_{t|t-1}), \tag{1.8}$$

où

$$K_t = P_{t|t-1}H(y_t)(H'(y_t)P_{t|t-1}H(y_t) + R)^{-1},$$

est la matrice de gain de Kalman,

$$P_{t+1|t} = F(y_t)[P_{t|t-1} - K_t H'(y_t)P_{t|t-1}]F'(y_t) + Q. \tag{1.9}$$

Ensuite nous calculons la prévision de X_{t+1} donnée par

$$\hat{X}_{t+1|t} = a(y_t) + H'(y_t)\hat{\xi}_{t+1|t} \tag{1.10}$$

et le carré moyen des erreurs associées

$$\hat{M}_{t+1|t} = E([X_{t+1} - \hat{X}_{t+1|t}][X_{t+1} - \hat{X}_{t+1|t}]') = H'(y_t)P_{t+1|t}H(y_t) + R. \qquad (1.11)$$

Nous avons présenté un descriptif succinct du filtre de Kalman, dont l'utilité sera explicitée dans le paragraphe suivant et pour plus de détail de cet algorithme, nous invitons le lecteur à consulter le livre de Hamilton (1994).
Dans le paragraphe suivant, avant de donner notre nouvelle approche d'estimation par la méthode du maximum de vraisemblance, nous allons faire un tour d'horizon sur les méthodes d'estimation des paramètres des modèles bilinéaires.

1.6 Les méthodes d'estimation des paramètres des modèles bilinéaires

Parmi les problèmes de l'inférence statistique qui se posent dans les modèles bilinéaires est l'estimation des paramètres inconnus. La difficulté et la complexité de ces modèles ont poussé l'intérêt des chercheurs vers l'estimation des paramètres des modèles particuliers.
Dans ce paragraphe nous allons présenter les méthodes d'estimation des paramètres des modèles bilinéaires, nous évoquerons aussi notre nouvelle approche qui se base sur la méthode du maximum de vraisemblance et nous donnerons l'expression de la fonction de la log-vraisemblance. Dans la suite $\theta = (a_1, \ldots, a_p, c_1, \ldots, c_q, b_{11}, \ldots, b_{PQ}, \sigma^2)$ représente les paramètres à estimer.

1.6.1 Méthode des moindres carrés

Cette technique d'estimation consiste à minimiser la somme des carrées des erreurs sur un certain domaine des paramètres contenant la vraie valeur de θ. Elle a été abordée pour le modèle bilinéaire diagonal d'ordre 1 BL(1,0,1,1) par Pham et Tran (1981),

ensuite, elle fut généralisée par Guégan et Pham (1987), pour étudier les modèles bilinéaires diagonaux, en utilisant la représentation markovienne (Pham (1985)).
Cette méthode s'utilise pour tout modèle bilinéaire qui admet une représentation markovienne inversible.

1.6.2 Méthode des moments

L'esprit de cette méthode se base sur le calcul des moments pour estimer les paramètres des modèles bilinéaires régis par l'équation (1.2). Mais, vu la complexité des calculs des moments du modèle (1.2), l'estimation par cette méthode n'a été obtenue explicitement que pour des cas particuliers dont les moments d'ordres supérieurs existent, comme par exemple le modèle BL(0,0,2,1) (Guégan (1984)).
L'avantage de cette méthode est qu'elle permet d'obtenir des estimateurs initiaux, et son inconvénient paraît avec la difficulté des calculs dans des modèles bilinéaires complexes.

1.6.3 Méthode d'autocovariance

Cette méthode utilise la fonction d'autocovariance empirique pour estimer les paramètres des modèles bilinéaires. Elle a été abordée par Kim et Billard (1990) pour le modèle bilinéare diagonal d'ordre 1 BL(1,0,1,1) et par Mathews et Moon (1991) pour le modèle bilinéaire BL(p,0,p,1).

1.6.4 Méthode du Maximum de vraisemblance

Approche courante

La méthode du maximum de vraisemblance offre une approche générale à l'estimation des paramètres inconnus à l'aide de données. Soit X une variable aléatoire de densité de probabilité $f(x, \theta)$. Le problème consiste donc à construire une expression analytique fonction des réalisations de cette variable dans un échantillon de taille n,

permettant de trouver la valeur numérique la plus vraisemblable pour le paramètre.

L'estimation des paramètres des modèles bilinéaires par cette technique fut d'abord abordée par Guégan (1981) pour le modèle superdiagonal d'ordre 1 BL(0,0,2,1), parallèlement par Subba Rao (1981) pour les modèles superdiagonaux BL(p,0,p,p). Enfin, ce résultat fut généralisé pour les modèles superdiagonaux BL(p,0,P,Q), par Gabr et Subba Rao (1984).
Le principe de cette méthode au regard du modèle bilinéaire général (1.2) est le suivant :
On assume que le processus $(e_t, t \in \mathbb{Z})$ est gaussien, et on suppose que $(X_t, t \in \mathbb{Z})$ est inversible, ce qui permet d'écrire e_t en fonction de $X_t, t \in \mathbb{Z}$

$$e_t = f(X_t, X_{t-1}, \ldots).$$

On définit $\hat{e}_t$ de la même façon, en prenant $X_k = 0$ pour $k \leq 0$ et telque $\hat{e}_t - e_t \to 0$ quand $t \to \infty$. Soit ϕ l'application définie de Ω^n dans Ω^n telle que

$$\phi(X_1, X_2, \ldots, X_n) = (\hat{e}_1, \ldots, \hat{e}_n),$$

ϕ est une application bijective. En supposant que le jacobien de ϕ égal à la matrice identité, et que $\hat{e}_1, \ldots, \hat{e}_n$ sont de loi conjointe gaussienne, la log-vraisemblance de $X_1, \ldots, X_n$ s'écrit

$$L(X, \theta) = -\frac{1}{2}\ln(det((2\pi)^n \Sigma)) - \frac{1}{2}[\hat{e}_1, \ldots, \hat{e}_n]\Sigma^{-1}[\hat{e}_1, \ldots, \hat{e}_n]', \qquad (1.12)$$

où Σ est la matrice de variance-covariance du vecteur $(\hat{e}_1, \ldots, \hat{e}_n)$. Lorsque n est grand, les $\hat{e}_t$ approchent $e_t, t = k, \ldots, n$, pour k assez grand.
Si k est petit devant n, l'évaluation de la fonction log-vraisemblance de $(\hat{e}_1, \ldots, \hat{e}_n)$ sera négligeable par rapport aux autres variables.

Dans ces conditions, on approxime la log-vraisemblance par

$$L(X,\theta) = -\frac{1}{2}\ln(2\pi\sigma)^n - \frac{1}{2\sigma^2}\sum_{t=1}^{n}\hat{e}_t^2,$$

où on a remplacé dans (1.12), Σ^{-1} par $\frac{1}{\sigma^2}I_n$(I_n matrice identité d'ordre n).
Ainsi, le problème d'estimation par maximum de vraisemblance se réduit à la minimisation de

$$l(\theta) = \sum_{t=1}^{n}\hat{e}_t^2,$$

et il consiste à estimer σ^2 par $\frac{1}{n}$ fois le minimum ainsi obtenu. Pour obtenir les estimateurs du maximum de vraisemblance, on choisit judicieusement une méthode d'optimisation qui converge rapidement, à titre d'exemple, les méthodes du gradient. Les méthodes d'optimisation obligent un choix convenable des valeurs initiales des paramètres. Or, il n'y a pas de méthodes pour obtenir ces valeurs initiales pour les modèles bilinéaires, sauf l'approche que Subba Rao a proposé et qui consiste à ajuster un modèle ARMA et à utiliser les estimations des paramètres de ce modèle comme valeurs initiales. Néanmoins, cette approche n'est pas pertinente et en plus, il n'y a pas de résultats de convergence des estimateurs du maximum de vraisemblance ni de propriétés asymptotiques, pour les modèles bilinéaires.

Nouvelle approche

La méthode du maximum de vraisemblance traditionnelle dépend du choix d'un modèle ARMA (Auto-Régressif Moyenne Mobile) convenable, par le biais duquel on estime les valeurs iniltiales. Or, ce choix peut mener à des résultats insatisfaisantes. En plus, elle se base sur le calcul du gradient, qui peut être très coûteux, surtout pour les modèles bilinéaires de grand ordre.
Notre nouvelle approche d'estimation, suggère d'utiliser la méthode du maximum de vraisemblance combinée avec la méthode du filtre de Kalman pour estimer les paramètres des modèles bilinéaires. Dans cette thèse, nous nous sommes intéressés

à des cas particuliers des modèles bilinéaires pour appliquer notre approche. Le détail de cette étude est rapporté dans les chapitres suivants, ici, nous allons expliciter l'esprit de cette approche.

Notre idée s'est inspirée des travaux, sur les modèles ARMA, de plusieurs auteurs: Pearlman (1980), Mélard (1983), Shea (1987,1989), Harti (1995)... qui utilisent l'algorithme du filtre de Kalman pour évaluer la log-vraisemblance et fournir par la suite les estimations des paramètres des modèles ARMA.

Notre point de départ, était d'exprimer les modèle bilinéaires en un modèle d'état équivalent à (1.7), ensuite fournir l'expression de la log-vraisemblance à partir de cet espace.

Soit $x = (x_1, \ldots, x_n)$ les réalisations du processus $X_1, X_2, \ldots, X_n$. La densité du processus $X_1, X_2, \ldots, X_n$ s'obtient à partir de la densité des observations $x = (x_1, \ldots, x_n)$, cette densité, on peut la décomposer en un produit de densités conditionnelles

$$f(x;\theta) = f(x_1;\theta) f(x_2|x_1;\theta) \ldots f(x_n|x_1, \ldots, x_{n-1};\theta),$$

sous les hypothèses faites au paragraphe (1.4.2), ces densités conditionnelles sont normales, le terme général $f(x_t|x_1, \ldots, x_{t-1};\theta)$ est la densité de la loi normale de moyenne $\hat{X}_{t|t-1} = E(X_t|X_1, \ldots, X_{t-1})$ et de variance $\hat{M}_{t|t-1} = E(X_t - E(X_t|X_1, \ldots, X_{t-1})^2$.
L'expression de log-vraisemblance s'exprime alors (voir, Hamilton (1994)) par :

$$L(X;\theta) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\ln(\hat{M}_{t|t-1}) - \frac{1}{2}\sum_{t=1}^{n}\frac{(x_t - \hat{x}_{t|t-1})^2}{\hat{M}_{t|t-1}}. \qquad (1.13)$$

L'évaluation de la fonction de log-vraisemblance sera faite par la méthode du filtre de Kalman, qui va nous permettre le calcul de $\hat{X}_{t|t-1}$ et $\hat{M}_{t|t-1}$. Dans les problèmes d'estimation par maximum de vraisemblance, l'objectif est de réaliser un estimateur qui maximise la fonction de la log-vraisemblance, or, lors de notre étude expérimentale, nous avons trouvé des difficultés dans la recherche des dérivées partielles de la fonction (1.13) et pour éviter cette difficulté nous avons choisi des méthodes d'optimisation

sans dérivées. Dans la littérature on trouve plusieurs méthodes d'optimisation sans dérivées, rattachées à deux classes de méthodes d'optimisation, la classe des méthodes déterministes et la classe des méthodes stochastiques. Le paragraphe suivant est concerné par un bref exposé sur les méthodes d'optimisation et donne plus d'intérêt aux méthodes qui traitent les fonctions non différentiables.

1.7 Méthodes d'optimisation sans contraintes

Dans la pratique, les problèmes d'optimisation sans contraintes sont classifiés selon la nature mathématique de la fonction objectif à optimiser. Celle-ci peut être unidimensionnelle ou multidimensionnelle, continue ou discontinue, linéaire ou non linéaire, convexe ou non convexe, différentiable ou non différentiable.

Selon les caractéristiques du problème d'optimisation sans contrainte, nous pouvons appliquer différentes méthodes de résolution pour identifier sa solution. Ces méthodes sont séparées en deux grands groupes : les méthodes déterministes et les méthodes stochastiques.

1.7.1 Méthodes d'optimisation déterministes

Une méthode d'optimisation est dite déterministe lorsque son évolution vers la solution du problème est toujours la même pour un même contexte initial donné, ne laissant aucune place au hasard. Ce sont en général des méthodes efficaces, peu coûteuses, mais qui nécessitent une configuration initiale (point de départ) pour résoudre le problème. Ce sont souvent des méthodes locales, c'est-à-dire qu'elles convergent vers l'optimum le plus proche du point de départ, qu'il soit local ou global. Selon la dimension de la fonction objectif à optimiser, les méthodes déterministes peuvent être classifiées en *unidimensionnelles* ou *multidimensionnelles*.

Méthodes Déterministes Unidimensionnelles

Les méthodes déterministes unidimensionnelles sont utilisées dans l'optimisation de fonctions à un seul paramètre. Ces méthodes, aussi appelées méthodes de *Recherche Linéaire* (Line Search Methods), sont normalement basées sur des techniques qui permettent de localiser le point minimal de la fonction à partir de réductions successives de l'intervalle de recherche.

Dans la littérature, nous trouvons différentes méthodes unidimensionnelles parmi lesquelles la méthode de Brent (Brent (1973))(Press (1992)), la méthode de Dichotomie (Minoux (1983)), et la méthode de la Section Dorée (Minoux (1983))(Press (1992)). La plupart de ces méthodes ne supposent pas que la fonction à minimiser soit différentiable. Dans cette thèse, nous nous intéressons à la méthode de Brent, dont l'algorithme sera donné en annexe.

Méthodes Déterministes Multidimensionnelles

Les méthodes déterministes multidimensionnelles sont consacrées à l'optimisation de fonctions à un paramètre ou plus. Nous pouvons les diviser, en deux différents groupes : les méthodes analytiques ou de descente et les méthodes heuristiques ou géométriques. Les méthodes heuristiques explorent l'espace par essais successifs en recherchant les directions les plus favorables. Les implémentations de méthodes géométriques les plus souvent utilisées sont celles de la méthode du Simplex (Nelder et Mead (1965)), et la méthode de variations locales de Hooke et Jeeves (Cherruault (1999)). Les méthodes analytiques se basent sur la connaissance d'une direction de recherche souvent donnée par le gradient de la fonction. La plupart de ces méthodes sont d'ordre 1 (c'est-à-dire, nécessitent le calcul du gradient de la fonction) et exécutent des minimisations linéaires successives en faisant appel à des méthodes unidimensionnelles (Press (1992)). Les exemples les plus significatifs de méthodes analytiques sont la méthode de la Plus Grande Pente (Minoux (1983)), le Gradient Conjugué (Minoux (1983)) (Press (1992)), les méthodes Quasi-Newton (Press (1992)) et la méthode de Powell (Press (1992))(Pow-

ell (1964)).

Dans cette thèse nous avons utilisé la méthode de Powell pour résoudre le problème de maximisation multidimentionnelle de la fonction log-vraisemblance. Ci-dessous nous donnons une description de cette méthode.

Méthode de Powell

Cette méthode, sans dérivées, peut se rattacher à la famille des méthodes de directions conjugées. Elle repose essentiellement sur l'idée suivante : supposons tout d'abord que l'on cherche le minimum d'une fonction f successivement suivant p directions conjugées $d_1, d_2, \ldots, d_p$ en repartant à chaque fois du dernier point obtenu. En partant du point x^0, on construit ainsi la séquence $x^1, x^2, \ldots, x^p$, définie par :

$$\begin{aligned}
f(x^1) &= f(x^0 + \lambda_1 d_1) = \min_{\lambda}\{f(x^0 + \lambda d_1)\},\\
f(x^2) &= f(x^1 + \lambda_2 d_2) = \min_{\lambda}\{f(x^1 + \lambda d_2)\},\\
&\vdots\\
f(x^p) &= f(x^{p-1} + \lambda_p d_p) = \min_{\lambda}\{f(x^{p-1} + \lambda d_p)\},
\end{aligned}$$

avec λ un scalaire. On peut alors décrire l'algorithme de powell de la façon suivante :

Algorithme (Powell)

Etape 1 : Choisir un point initiale θ^0et p directions linéairement indépendantes $d_1, d_2, \ldots, d_p$
(initialement on peut partir des p directions définies par les axes de coordonnées). On engendre ensuite une séquence $\theta^1, \theta^2, \ldots, \theta^p$ de points tel que :
$f(\theta^i) = f(\theta^{i-1} + \lambda_i d_i) = \min_{\lambda}\{f(\theta^{i-1} + \lambda d_i)\}$

Etape 2 : Soit $f_0 = f(\theta^0), f_1 = f(\theta^p)$ et $f_2 = f(2\theta^p - \theta^0)$
(f_2 valeur de la fonction f au point symétrique de θ^0 par rapport à θ^p).
Soit d'autre part :
$\Delta = \max_{i=1,\ldots,p}\{f(\theta^{i-1}) - f(\theta^i)\}$
le maximum étant atteint par l'indice m (la plus grande diminution de f obtenue (en étape 1) dans la direction d_m), alors deux cas se présentent

Etape 3 : **Si** $f_2 \geq f_1$ et/ou **Si**
$(f_0 - 2f_1 + f_2)(f_0 - f_1 - \Delta)^2 \geq 1/2\Delta(f_0 - f_2)^2$
utiliser les anciennes directions $d_1, \ldots, d_p$ pour l'itération suivante avec comme nouveau point θ^p.
Sinon,
Si $f_2 < f_1$et/ou **Si** $(f_0 - 2f_1 + f_2)(f_0 - f_1 - \Delta)^2 < 1/2\Delta(f_0 - f_2)^2$
rechercher le minimum de $f(x)$ dans la direction $\theta^p - \theta^0$.
Le point obtenu sera pris comme nouveau point de départ θ^0 à l'itération suivante.
D'autre part l'ordre dans lequel ces directions seront utilisées au cours des itérations suivantes étant :
$d_1, d_2, \ldots, d_{m-1}, d_{m+1}, \ldots, d_p, d$

Fin Algorithme

Comme on peut le remarquer, cet algorithme requiert, dans l'étape 1, la résolution d'un problème de minimisation unidimentionnel. Pour résoudre ce problème, nous avons opté pour la méthode de Brent dont l'algorithme est donné en annexe A.
Comme nous l'avons signalé auparavant, ces méthodes déterministes nécessitent un point de départ, contrairement aux méthodes d'optimisation stochastiques qui ne nécessitent ni de point de départ, ni la connaissance du gradient de la fonction objectif pour atteindre la solution optimale. Cependant, elles demandent un nombre important d'évaluations avant d'arriver à la solution du problème.

1.7.2 Méthodes d'optimisation Stochastique

Les méthodes d'optimisation stochastiques s'appuient sur des mécanismes de transition probabilistes et aléatoires. Cette caractéristique indique que plusieurs exécutions successives de ces méthodes peuvent conduire à des résultats différents pour une même configuration initiale d'un problème d'optimisation. Ces méthodes ont une grande capacité de trouver l'optimum global du problème.
Parmi les méthodes stochastiques les plus employées, nous distinguons SPSA (Approximation Stochastique de Perturbation Simultanée)(Spall (1992)) et le Recuit Simulé (Corana et al. (1987)), dans ce qui suit nous donnerons un descriptif succinct de la méthode du Recuit Simulé, un descriptif de la méthode SPSA est donné en annexe A.
La méthode du Recuit Simulé est basée sur le processus de recuit utilisé en métallurgie, dans lequel on cherche à obtenir un matériau sans impureté, représenté par son état d'énergie minimal.

L'algorithme proposé par Corana (1987) commence avec une configuration de paramètres choisie au hasard θ_0 et une température initiale T_0 élevée. Ensuite, un nouveau candidat θ est généré à partir de l'équation

$$\theta = \theta_0 + rS_{m_h}e_h$$

où r est un nombre aléatoire généré dans l'interval $[-1; 1]$ par un générateur de nombres pseudo-aléatoires. e_h est un vecteur de base avec 1 dans la $h^{\text{ème}}$ position et S_{m_h} et la $h^{\text{ème}}$ composante du vecteur d'ajustement.

La fonction est donc évaluée sur ces deux configurations, θ et θ_i, ce qui nous permet de calculer l'écart $\Delta f = (f(\theta_i) - f(\theta))$. Si $\Delta f < 0$, nous remplaçons la configuration originale par la nouvelle configuration obtenue. Dans le cas contraire, nous considérons la probabilité donnée par $p = \exp\left(\frac{f(\theta_i)-f(\theta)}{T_i}\right)$ pour décider si la configuration initiale doit être remplacée ou pas. À chaque itération de la méthode, ce processus est répété jusqu'à ce que nous obtenions l'équilibre thermique, c'est-à-dire, la solution optimale.

Algorithme du Recuit simulé

Etape 0 : (Initialisation)

Choisissez :

-Un point initial θ_0.

-un vecteur initial v_0 (appelé : step vector).

-Une température initiale T_0.

-Un critère d'arrêt ϵ et un nombreN_ϵ

de réductions de température successives pour tester l'arrêt.

-$N_s$nombre de variation et c critère de variation.

-N_T nombre de réduction de la température

-r_T le coefficient de reduction de la température.

-Initialisez i, j, m, k à 0.

i est l'indice qui désigne les points successifs,

j désigne les cycles successifs dans chaque direction,

m décrit les étapes d'ajustements successives, et

k désigne les réductions de températures successives.

Initialisez h à 1. h est l'indice qui désigne la direction à travers

laquelle le point suivant est généré, en partant du dernier point accepté.

Calculez : $f_0 = f(\theta_0)$.

Mettez : $\theta_{opt} = \theta_0, f_{opt} = f_0$.

Mettez : $n_u = 0, u = 1, \ldots, n$.

Mettez : $f_u^* = f_0, u = 0, -1, \ldots, -N, +1$

Etape 1 : Partant du point θ_i, générez le point aléatoire θ dans la direction d_h :

$\theta = \theta_i + r\nu_{m_h} d_h$

où r est un nombre généré dans l'intervalle $[-1, l]$ par un générateur

de nombre pseudo-aléatoire; d_h est le vecteur dont la $h^{\text{ème}}$ position

égale à 1 et 0 ailleurs; et ν_{m_h} est la composante du vecteur v_m, dans la même direction.

Etape 2 : **Si** le $h^{\text{ème}}$ coordonné de θ n'appartient pas au domaine de définition de f, **Alors** retournez à l'Etape 1.

Etape 3 : Calculez $f_\theta = f(\theta)$.

Si $f_\theta < f_i$ **Alors** acceptez le nouveau point :

Mettez $\theta_{i+1} = \theta_i$,

Mettez $f_{i+1} = f_\theta$,

$i = i + 1$,

$n_h = n_h + 1$;

Si $f_\theta < f_{opt}$, **Alors** mettez

$\theta_{opt} = \theta$,

$f_{opt} = f_\theta$.

Sinon acceptez ou rejetez le point avec une probabilité d'acceptation égale à p où $p = exp\left(\frac{f_i - f_\theta}{T_k}\right)$.

Généré un nombre p' pseudo-aléatoire dans l'intervalle $[0, 1]$.

Si $p' > p$ le point est rejeté.

Sinon

Mettez $\theta_{i+1} = \theta_i$,

Mettez $f_{i+1} = f_\theta$,

Mettez $i = i + 1$,

Mettez $n_h = n_h + 1$.

Etape 4 : $h = h + 1$.

Si $h \leq n$ **Alors** allez à l'Etape 1.

Sinon $h = 1$ et $j = 1$.

Etape 5 : **Si** $j < N_s$ **Alors** allez à l'Etape 1.

Sinon mettez à jour le vecteur v, pour chaque direction u

la nouvelle composante de v'_u est

$v'_u = \nu_{m_u}\left(1 + c_u \frac{n_u/N_s - 0.6}{0.4}\right)$ Si $n_u > 0.6N_s$

$v'_u = \frac{\nu_{m\,u}}{1 + c_u \frac{0.4 - n_u/N_s}{0.4}}$ Si $n_u < 0.4N_s$

$v'_u = \nu_{m_u}$

Mettez $v_{m+1} = v'$,

Mettez $j = 0$,

Mettez $n_u = 0$ pour $u = 1, \ldots, n$,

Mettez $m = m + 1$.

Etape 6 : **Si** $m < N_T$ **Alors** allez à l'Etape 1.

Sinon

Mettez $T_{k+1} = r_T T_k$,

Mettez $f_k^* = f_i$,

Mettez $k = k + 1$,

Mettez $m = 0$.

Etape 7 : (Critères d'arrêt)

Si $\mid f_k^* - f_{k-u}^* \leq \epsilon \mid, u = l, \ldots, N_\epsilon$

et $f_k^* - f_{opt} < \epsilon$,

Alors arrêtez la recherche de l'optimum.

Sinon

$i = i + 1$,

Mettez $\theta_i = \theta_{opt}$,

Mettez $f_i = f_{opt}$.

Allez à l'Etape 1.

FinAlgorithme

1.8 Fiabilité des logiciels

Le logiciel est une entité inhérente aux systèmes informatiques dont notre société devient de plus en plus dépendente.
Au fur et à mesure que sa taille croît, il devient de plus en plus difficile et coûteux de détecter et corriger ces fautes.
Sachant qu'un logiciel commercial standard fait en moyenne 350000 lignes de codes et dans des applications plus complexes, plusieurs millions de codes, on comprend que les logiciels contiendront toujours des fautes susceptibles d'engendrer des défaillances. Ces défaillances peuvent avoir des conséquences catastrophiques : parmi les défaillances logicielles célèbres, on peut citer le bug qui a provoqué l'explosion de la fusée Ariane 5 en Juin 1999. Il est donc très important de pouvoir prévoir l'occurence des défaillances et d'évaluer la fiabilité des logiciels.
Dans la suite nous donnerons les concepts de base en fiabilité des logiciels et nous définissons le processus des défaillances permettant de modéliser les défaillances agissant sur son comportement et nous présentons aussi le modèle de fiabilité des logiciels de Chen et Singpurwalla.

1.8.1 Concepts de base en fiabilité des logiciels

Définition 15

- **Un logiciel** *est un système qui par l'intermédiaire d'un programme transforme des données d'entrée* (*instructions, chiffres, images, fichiers...etc*) *en données de sorite* (*résultats*).

- **La fiabilité** *d'un logiciel est la probabilité qu'il fonctionne sans défaillances pendant une durée donnée et dans un environnement spécifié.*

- **Une défaillance** *se produit lorsque le résultat fourni par le logiciel n'est pas conforme au résultat prévu par les spécifications.*

- **Les spécifications** *du logiciel définissent quels doivent être les résultats fournis pour les différentes données d'entrées.*

- **Une faute** *logiciel ou bug est un défaut du programme qui, exécuté dans certaines conditions, entraînera une défaillance. En supposant que l'on puisse corriger les fautes, les performances du logiciel se trouvent améliorées. Donc le logiciel constitue un exemple typique de* **systèmes améliorables**.

- **Une correction** *ou débogage est l'opération à travers laquelle on élimine une faute qui a provoqué une défaillance.*

Le comportement d'un logiciel et en particulier sa fiabilité évolue au cours du temps en fonction de trois facteurs fondamentaux

- Les fautes de conception.

- Le profil d'utilisation qui décrit le comportement des utilisateurs du logiciel (choix des entrées, fréquence des sollicitations...etc).

- Les modifications et les corrections que subit le logiciel au cours de son cycle de vie.

1.8.2 Approches boîte noire et boîte blanche

L'approche qu'on adoptera dans le chapitre 2 est l'approche boîte noire. Dans cette approche le logiciel est vu comme une entité permettant de transformer des données d'entrées en données de sorties qui doivent être conformes aux spécifications, sans faire référence à la structure interne du programme. Contrairemnet au modèle boîte blanche (appelé aussi approche structurelle ou boîte de verre) qui considère un logiciel comme un système de modules ou composants logiciels, introduisant ainsi la structure interne du logiciel.

1.8.3 Processus de défaillances

Le processus de défaillance est un processus ponctuel de $\mathbb{R}^+$ défini indifféremment par

- $T = (T_i)_{i\geq 1}$ où T_i est l'instant de la $i^{\text{ème}}$ défaillance.
- $X = (X_i)_{i\geq 1}$ où $X_i = T_i - T_{i-1}$ (avec $T_0 = 0$) est la durée séparant la $(i-1)^{\text{ème}}$ et la $i^{\text{ème}}$ défaillance, c'est le temps interdéfaillance.
- $N = (N_t)_{t\geq 0}$ où N_t est le nombre cumulé de défaillance entre l'instant initial et l'instant t.

Dans cette étude nous nous intéresserons au processus des temps interdéfaillances.

1.8.4 Modèle ICD de Chen et Singpurwalla

Chen et Singpurwalla (1992) ont développé leur modèle ICD du filtre de Kalman non-gaussien, en se basant sur le travail de Bather (1965) intitulé "*Invariant Conditional Distibutions* (ICD)", d'où l'appelation du modèle. Dans ce paragraphe, nous allons le décrire brièvement.

Soit X_n l'observation au temps n, qui dénote aussi le temps interdéfaillance, Z_n l'état au temps n et $D_n = X_1, \ldots, X_n$ un ensemble d'observations jusqu'au temps n.
Les auteurs supposent que :

- $(X_n|Z_n)$ suit la loi de Gamma de paramètres (ω, Z_n),
- $(Z_n|D_n)$ suit la loi de Gamma de paramètres $(\upsilon + \omega, u_{n-1})$,
- $(Z_0|D_0)$ suit la loi de Gamma de paramètres $(\upsilon + \omega, u_0)$, où ω, C et υ sont supposées connues, et où $u_n = Cu_{n-1} + x_n$.

Le calcul de la prédiction des temps d'interdéfaillance X_n se fait à partir de l'équation suivante :

$$\hat{X}_n = E[X_n|D_{n-1}; C] = 2Cu_{n-1},$$

où $u_n = x_n + Cx_{n-1} + C^2x_{n-2} + \ldots + C^nx_0$.

Le paramètre C joue un ample rôle dans la description de la croissance ou la décroissance de la fiabilité.

Pour plus de détails consulter (Chen et Singpurwalla (1994)).

Les prédictions des temps interdéfaillances rapportées dans Chen et Singpurwalla (1994), seront comparées avec celles obtenues par le modèle bilinéaire superdiagonal d'ordre un présenté dans le chapitre suivant.

1.9 Conclusion

Dans ce chapitre les fondements de la conception de nos algorithmes d'estimations des paramètres de certain modèles bilinéaires particuliers ont été présentés. Cette conception débute par la présentation des modèles bilinéaires sous forme espace d'état, ensuite, formule la fonction de vraisemblance et emploie le filtre de Kalman pour le calcul de cette vraisemblance.

Le chapitre suivant, sera consacré, en premier lieu, à l'élaboration de l'algorithme d'estimation pour le modèle bilinéaire superdiagonal d'ordre un BL(0,0,2,1), et à la concrétisation de la performance de cet algorithme par le biais des simulations.

Ensuite, une étude applicative des modèles de séries chronologiques linéaires ARIMA et le modèle bilinéaire BL(0,0,2,1) en fiabilité des logiciels sera illustrée.

Chapitre 2

Estimation des paramètres du modèle bilinéaire superdiagonal d'ordre un BL(0,0,2,1) : Application en fiabilité des logiciels

2.1 Introduction

L'estimation des paramètres du modèle BL(0,0,2,1) a été évoquée par Guégan (1984), qui a proposé des estimateurs des paramètres de ce modèle par la méthode des moments, et qui a affirmé que les valeurs de ces estimateurs peuvent servir de valeurs initiales, ce qui montre que, ces estimateurs ne sont pas performants, et que nous avons besoin d'une technique fiable d'estimation.

Dans notre contribution, nous proposons un nouvel algorithme d'estimation des paramètres de ce modèle, qui se base sur la méthode du maximum de vraisemblance et l'algorithme du filtre de Kalman. Le présent chapitre se compose de deux parties, la première partie présente notre algorithme d'estimation des paramètres du modèle bilinéaire superdiagonal d'ordre un et les résultats numériques des simulations par la

méthode de Monte Carlo pour justifier les performances de cet algorithme. La seconde partie traite deux applications en fiabilité des logiciels. Ce chapitre a fait l'objet de deux articles (Bouzaachane et al (2006a), (2006b)).

2.2 Algorithme d'estimation des paramètres du modèle superdiagonal d'ordre un

Dans ce paragraphe, nous allons expliciter notre nouvel algorithme d'estimation pour le modèle BL(0,0,2,1) et nous montrerons sa fermeté par rapport à la méthode des moments (Guégan (1984)), à travers des simulations de Monte Carlo.

2.2.1 Modèle bilinéaire superdiagonal d'ordre un

Un modèle bilinéaire superdiagonal d'ordre un noté BL(0,0,2,1) est une équation aux différences stochastiques non linéaire de la forme

$$X_t = b_{21} X_{t-2} e_{t-1} + e_t, \tag{2.1}$$

où $(e_t, t \in \mathbb{Z})$ est une suite de variables aléatoires indépendantes et équidistribuées de moyenne nulle et de variance σ^2 finie.
Le processus $(X_t, t \in \mathbb{Z})$ défini par l'équation (2.1) est dit processus bilinéaire superdiagonal d'ordre un.
Guégan (1981) a montré l'existence, la stationnarité et l'inversibilité d'un tel processus, ces résultats sont résumés dans les théorèmes ci-dessous.

Théorème 4 *(Guégan, 1981)*
Etant donné une suite de variables aléatoires $(e_t, t \in \mathbb{Z})$, *indépendantes équidistribuées, centrées de variance finie, définies sur un certain espace de probabilité* $(\Omega, \Re, \wp)$, *une condition nécessaire et suffisante d'existence d'un processus* $(X_t, t \in \mathbb{Z})$ *strictement*

stationnaire, de carré intégrable défini par (2.1) *est :*

$$b_{21}^2\sigma^2 < 1. \tag{2.2}$$

Sous cette condition, le processus $(X_t, t \in \mathbb{Z})$ *est unique et est donné par l'équation :*

$$X_t = e_t + \sum_{j\geq 1} b_{21}^j e_{t-2j} \prod_{k=1}^{j} e_{t-2k+1}. \tag{2.3}$$

Cette somme converge en moyenne quadratique.

Théorème 5 *(Guégan, 1981)*
Etant donné une suite de variables aléatoires $(e_t, t \in \mathbb{Z})$*, indépendantes équidistribuées, centrées de variance finie, définies sur un certain espace de probabilité* $(\Omega, \Re, \wp)$*, une condition suffisante d'inversibilité du processus* $(X_t, t \in \mathbb{Z})$ *est :*

$$2b_{21}^2\sigma^2 < 1. \tag{2.4}$$

Sous cette condition le processus a pour expression

$$e_t = X_t - b_{21}X_{t-2}X_{t-1} - \sum_{i=1}^{\infty}[(-b_{21})^i]X_{t-i-2}X_{t-i-1}\prod_{j=1}^{i} X_{t-j-1}. \tag{2.5}$$

Si la condition (2.4) est vérifiée, (2.5) est presque sûrement convergente et de somme :

$$e_t^{b_{21}} = X_t - b_{21}X_{t-2}X_{t-1} - \sum_{i=1}^{t-2}[(-b_{21})^i b_{21}X_{t-i-2}X_{t-i-1}\prod_{j=1}^{i} X_{t-j-1}]. \tag{2.6}$$

Cette expression découle de la procédure suivante : soit $X_t = b_{21}X_{t-2}e_{t-1} + e_t$, on pose $Z_t = b_{21}X_{t-1}e_t$ alors X_t peut s'écrire $X_t = Z_{t-1} + e_t$.

$$\begin{aligned}
Z_t &= b_{21}X_{t-1}e_t \\
&= b_{21}X_{t-1}(X_t - Z_{t-1}) \\
&= b_{21}X_{t-1}X_t - b_{21}X_{t-1}Z_{t-1} \\
&= b_{21}X_{t-1}X_t - b_{21}X_{t-1}(b_{21}X_{t-2}X_{t-1} - b_{21}X_{t-2}Z_{t-2}) \\
&= b_{21}X_{t-1}X_t + (-b_{21})b_{21}(X_{t-1}X_{t-2})X_{t-1} + (-b_{21})^2X_{t-1}X_{t-2}Z_{t-2} \\
&= b_{21}X_{t-1}X_t + (-b_{21})b_{21}(X_{t-1}X_{t-2})X_{t-1} + (-b_{21})^2X_{t-1}X_{t-2}(b_{21}X_{t-3}X_{t-2} \\
&\quad -b_{21}X_{t-3}Z_{t-3})
\end{aligned}$$

De proche en proche on obtient pour Z_t l'expression suivante :

$$Z_t = b_{21}X_{t-1}X_t + \sum_{i=1}^{t-1}(-b_{21})^i(b_{21}X_{t-i}X_{t-i-1})\{\prod_{k=1}^{i}X_{t-k}\} + (-b_{21})^t\{\prod_{k=1}^{t}X_{t-k}\}Z_0.$$

En posant $Z_0 = z_0$, nous obtenons l'équation suivante

$$\begin{aligned}
e_{t,z0}^{b_{21}} = X_t - b_{21}X_{t-2}X_{t-1} - \sum_{i=1}^{t-2}[(-b_{21})^i b_{21}X_{t-i-2}X_{t-i-1}\prod_{j=1}^{i}X_{t-j-1}] \\
-(-b_{21})^{t-1}[\prod_{j=1}^{t-1}X_{t-j-1}]z_0.
\end{aligned} \tag{2.7}$$

Lorsque la condition (2.4) est vérifiée nous avons $e_{t,z0}^{b_{21}} - e_t^{b_{21}}$ converge presque sûrement vers 0 quand $t \to \infty$. La notion d'inversibilité est très importante dans les méthodes d'estimation en général, et dans notre algorithme en particulier. Dans le paragraphe suivant, nous verrons que la vérification de la condition d'inversibilité va nous faciliter le calcul de la log-vraisemblance.

2.2.2 Conception de l'algorithme

La phase essentielle de notre algorithme est de réécrire le modèle (2.1) sous forme d'espace d'état. Dans le chapitre précédent nous avons présenté le modèle affine en

état polynomial en entrée, introduit par Guégan, qui exprime les modèles bilinéaires sous forme d'espace d'état. En se basant sur cette représentation, nous définissons la représentation sous forme d'état du modèle (2.1) par :

$$\begin{cases} \xi_{t+1} &= A(e_t)\xi_t + v_{t+1} & : \text{équation d'état} \\ X_t &= H\xi_t & : \text{équation d'obseration} \end{cases} \tag{2.8}$$

où le vecteur d'état est $\xi_t = \begin{bmatrix} X_t \\ X_{t-1} \\ X_{t-2} \end{bmatrix}$, $v_t = [e_t, 0, 0]'$, $H = [1, 0, 0]$ et

$$A(e_t) = \begin{pmatrix} 0 & b_{21}e_t & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Si la condition (2.4) d'inversibilité est vérifiée, $e_t - e_t^{b_{21}}$ converge presque sûrement vers 0 ce qui implique la convergence presque sûre de $A(e_t) - A(e_t^{b_{21}})$ vers 0, cette dernière sera exprimée en fonction de X_k pour $k = 1, \ldots, t$ d'après l'équation (2.6). Ceci, nous ramène à la présentation d'espace d'état avec paramètres stochastiques (1.7).
Soit $\theta = (b_{21}, \sigma^2)$ le vecteur des paramètres inconnus pour lesquels nous souhaitons construire des estimateurs par la méthode du maximum de vraisemblance. La vraisemblance des observations $x_1, \ldots, x_n$ est la suivante (Hamilton 1994)

$$f(x;\theta) = f(x_1;\theta) \prod_{i=2}^{n} f(x_i|x_1, \ldots, x_{i-1};\theta),$$

elle est décomposée en un produit de densités conditionnelles. En supposant que les X_t sont normales, ces densités conditionnelles sont normales. Le terme général $f(x_t|x_1, \ldots, x_{t-1};\theta)$ est la densité de la loi normale de moyenne $\hat{X}_{t+1|t}$ et de variance $\hat{M}_{t+1|t}$. Pour toute valeur fixée de θ, nous pouvons déterminer $\hat{X}_{t+1|t}$ et $\hat{M}_{t+1|t}$ par le filtre de Kalman. Nous pouvons donc utiliser le filtre de Kalman pour calculer la

log-vraisemblance en une valeur θ donnée. Cette log-vraisemblance a pour expression

$$L(x;\theta) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\ln(\hat{M}_{t|t-1}) - \frac{1}{2}\sum_{t=1}^{n}\frac{(x_t - \hat{x}_{t|t-1})^2}{\hat{M}_{t|t-1}} \tag{2.9}$$

Les valeurs de la vraisemblance ainsi déterminées par le filtre de Kalman vont être utilisées dans un algorithme de maximisation, ayant pour but de fournir l'estimation du maximum de vraisemblance. Maximiser $L(x;\theta)$ est équivalent à minimiser $-L(x;\theta) = l(x;\theta)$. Dans la suite, nous allons considérer le problème de minimisation de $l(x;\theta)$.

En s'apercevant de la difficulté des calculs des dérivées partielles de $l(x;\theta)$, nous avons choisi la méthode de Powell, décrite dans le chapitre précédent, qui est une méthode multidimensionnelle déterministe sans dérivées, que nous avons combinée avec la méthode de Brent pour résoudre le problème de minimisation dans le cas unidimensionnel. Nous allons voir dans l'algorithme, que nous décrirons ci-dessous, l'étape où intervient cette méthode.

Comme tout algorithme d'optimisation, le choix des valeurs initiales est fondamental. Pour notre algorithme d'estimation, nous avons choisi les valeurs issues des estimateurs de Guégan dont l'expression est la suivante

$$\hat{b}_{21} = \frac{\hat{c}(0)\hat{c}(0,1,4)}{2\hat{c}^3(1,2)}, \quad \hat{\sigma}^2 = \frac{\hat{c}^2(1,1,3)}{2\hat{c}(0)\hat{c}(0,1,4)},$$

où

$\hat{c}(k) = \frac{1}{n}\sum_{t=1}^{n} X_t X_{t-k}$,

$\hat{c}(i,j) = \frac{1}{n}\sum_{t=1}^{n} X_t X_{t-i} X_{t-j}$,

$\hat{c}(i,j,k) = \frac{1}{n}\sum_{t=1}^{n} X_t X_{t-i} X_{t-j} X_{t-k}$.

expriment respectivement les estimateurs de

$c(k) = E[X_t X_{t-k}]$,

$c(i,j) = E[X_t X_{t-i} X_{t-j}]$,
$c(i,j,k) = E[X_t X_{t-i} X_{t-j} X_{t-k}]$.

Présentement, et avant de décrire notre algorithme d'estimation, que nous notons MLKF (Maximum Likelihood and Kalman Filter estimation), nous allons présenter le sous algorithme qui calcule la log-vraisemblance par le filtre de Kalman et le sous algorithme qui teste la vérification de la condition d'inversibilité (la vérification de la condition d'existence et de stationnarité en découle directement puisque $2b_{21}^2\sigma^2 < 1 \Longrightarrow b_{21}^2\sigma^2 < 1$).

/* Sous algorithme qui calcule la log-vraisemblance par le filtre de Kalman*/

Sous algorithme KF(θ)

Etape 1 : Entrez les valeurs initiales de $\hat{\xi}_{1|0}$ et $P_{1|0}$.
Calculez : $\hat{X}_{1|0} = H\hat{\xi}_{1|0}$ et $\hat{M}_{1|0} = HP_{1|0}H'$.

Etape 2 : **Pour** $t = 1$ **jusqu'à** n **Faire**
(n est la taille de l'échantillon)
Calculez $K_t, \hat{\xi}_{t|t}, P_{t|t}, \hat{\xi}_{t+1|t}, P_{t+1|t}$.
Calculez $\hat{X}_{t+1|t}, \hat{M}_{t+1|t}$.
FinPour

Etape 3 : $som = \frac{n}{2}\ln(2\pi)$
Pour $t = 1$ **jusqu'à** n**Faire**
$som = som + \frac{1}{2}ln(\hat{M}_{t+1|t}) + \frac{1}{2}\frac{(x_{t+1}-\hat{x}_{t+1|t})^2}{\hat{M}_{t+1|t}}$
FinPour
$l(x;\theta) = som$.

Fin Sous algorithme

Les valeurs initiales de $\hat{\xi}_{1|0}$ et $P_{1|0}$ du filtre de Kalman sont déduites d'une étude faite par Guégan (1984) qui a montré que

- $E[X_t] = 0 \ \forall \ t$

- $cov(X_t, X_{t-k}) = 0 \ \forall \ k \neq 0$
- $var(X_t) = \frac{b_{21}^2}{1-b_{21}^2\sigma^2}$

d'où on a

$$\text{- } \hat{\xi}_{1|0} = E[\xi_1] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\text{- } P_{1|0} = E[(\xi_1 - E[\xi_1])(\xi_1 - E[\xi_1])'] = \begin{bmatrix} \frac{b_{21}^2}{1-b_{21}^2\sigma^2} & 0 & 0 \\ 0 & \frac{b_{21}^2}{1-b_{21}^2\sigma^2} & 0 \\ 0 & 0 & \frac{b_{21}^2}{1-b_{21}^2\sigma^2} \end{bmatrix}.$$

/* Sous algorithme test */

Sous algorithme Test(θ)

Etape 1 : **Si** $(b_{21}^2\sigma^2 < 1/2)$ **Alors** continuez

Etape 2 : **Sinon** retournez à l'étape précedente et prenez le point antérieur comme un point initiale.

Fin Sous algorithme

Algorithme **MLKF**

Etape 1 : Soit $\theta^{(0)}$ point initial(vérifie le test) et soit $d_1, \ldots, d_m$ les vecteurs de base (ensemble initiale de directions).

Etape 2 : **Pour** $k = 1$ **jusqu'à** m **Faire**

(m est le nombre de paramètres)

Appelez le sous algorithme KF $(\theta^{(k)})$;

Etape 3 : Résolvez $\min_{\lambda} l(X; \theta^{(k-1)} + \lambda * d_k) = l(X; \theta^{(k-1)} + \lambda_k * d_k)$;

Etape 4 : Appelez le sous algorithme $Test(\theta^{(k-1)} + \lambda_k * d_k)$;

Etape 5 : Mettez $\theta^{(k)} \longleftarrow \theta^{(k-1)} + \lambda_k * d_k$;

FinPour

Etape 6 : Calculez :

- $l_0 = l(X;\theta^{(0)})$;
- $l_m = l(X;\theta^{(m)})$;
- $l_E = l(X;2*\theta^{(m)} - \theta^{(0)})$;
- $\Delta l = \max_{i=1,\ldots,m}\{l(X;\theta^{(i-1)}) - l(X;\theta^{(i)})\}$;

Soit s l'indice pour lequel le maximum est atteint (la plus grande diminution de l obtenu à l'étape 3 dans la direction d_s).

Alors nous avons deux cas :

Etape 7 : **Si** $((l_E \geq l_0)$et/ou $(l_0 - 2*l_m + l_E)(l_0 - l_m - \Delta L)^2 \geq \frac{1}{2}\Delta l(l_0 - l_E)^2)$

Alors utilisez les anciennes directions dans l'itération suivante et $\theta^{(m)}$ comme un point initial.

Sinon,

Si $((l_E < l_0)$et/ou $(l_0 - 2*l_m + l_E)(l_0 - l_m - \Delta l)^2 < \frac{1}{2}\Delta l(l_0 - l_E)^2)$

Alors Allez à l'étapte 1

Remplacez la direction d_s par $d = \theta^{(m)} - \theta^{(0)}$

Utilisez l'ensemble des directions $(d_1, d_2, \ldots, d_{s-1}, d_{s+1}, \ldots, d_m, d)$ dans l'iteration suivante.

Etape 8 : Les étapes 1 jusqu'à 7 se répètent jusq'à ce que l'un des critères d'arrêts soit vérifié. Ces critères sont :le nombre d'iteration maximum est dépassé ou le minimum est obtenu.

Fin Algorithme

Dans cet algorithme, la méthode de Brent intervient dans l'étape 3 considérée par la minimisation unidimentionnelle de la fonction $g(\lambda) = l(X;\theta + \lambda * d)$.

Pour examiner la qualité de cet algorithme, nous avons réalisé une série de simulations numériques, après l'avoir codé en langage C.

Dans le paragraphe suivant nous allons décrire l'expérience menée et présenter les résultats obtenus.

2.2.3 Simulations

Les résultats des simulations que nous dressons dans ce paragraphe, ont pour objectif l'évaluation de la performance et la qualité de notre algorithme conçu pour estimer les paramètres du modèle bilinéaire BL(0,0,2,1).
Dans tout algorithme d'estimation il est important de choisir judicieusement les valeurs initiales pour avoir une bonne estimation et une convergence rapide.
Dans cette étude nous nous sommes interessés à deux manières de choix des valeurs initiales. Dans la première, les valeurs sont les estimations obtenues par la méthode des moments fournis par Guégan (1984), et dans la deuxième les valeurs sont choisies arbitrairement.
Dans les deux études le processus $(e_1, e_2 \ldots, e_n)$ simulé suit une loi normale centrée. Cette simulation a été réalisée à partir du générateur de nombres aléatoires existant dans la version 4.0 du logiciel libre Scilab développé par l'INRIA.
Dans la première étude nous considérons les modèles bilinéaires BL(0,0,2,1) définis par les équations suivantes :

- $X_t = 0.5X_{t-2}e_{t-1} + e_t, \quad e_t \sim iidN(0,1)$
- $X_t = 0.85X_{t-2}e_{t-1} + e_t, \quad e_t \sim iidN(0,\sqrt{0.5})$

Les conditions d'inversibilité et de stationnarité sont vérifiées par les coefficients des deux modèles. L'objectif dans cette expérience est d'approcher les vecteurs (0.5,1) et (0.85,0.5). Pour chaque modèle, ci-dessus, nous avons généré des séries de tailles $n = 50, 100$ et 150, et de 300 réplications.
Les résultats obtenus par cette expérience sont résumés dans les tableaux 1, 2 et 3, où pour chaque estimateur nous donnons la moyenne, le biais, la T-statistic $(\frac{biais}{ecart-type})$ et MSE, et où nous avons utilisé les notations suivantes, MM désigne l'estimation par la méthode du moments et MLKF désigne l'estimation par notre algorithme.

Paramètres	b_{21}	σ^2	b_{21}	σ^2
Vraies valeurs	0.5	1	0.85	0.5
Moyenne de MLKF	0.5112	0.9886	0.842	0.5649
Biais de MLKF	-0.0112	0.0114	0.008	-0.064
MSE de MLKF	0.0491	0.225	0.0931	0.1396
T-statistic	-0.05	0.024	0.0262	-0.173
Moyenne de MM	0.4001	0.6551	0.6574	0.3469
MSE de MM	0.0501	0.272	0.0922	0.1396

Table 2.1: Moyenne, Biais, MSE et T-statistic des paramètres estimés pour n=50

Paramèters	b_{21}	σ^2	b_{21}	σ^2
Vraies valeurs	0.5	1	0.85	0.5
Moyenne de MLKF	0.4999	0.9986	0.8175	0.5220
Bias of MLKF	0.0001	0.0014	0.0325	-0.022
MSE of MLKF	0.0273	0.1341	0.1024	0.0371
T-statistic	0.0006	0.0038	0.101	-0.114
Moyenne de MM	0.395	0.643	0.6376	0.2686
MSE of MM	0.0325	0.1767	0.0958	0.0471

Table 2.2: Moyenne, Biais, MSE et T-statistic des paramètres estimés pour n=100

Paramètres	b_{21}	σ^2	b_{21}	σ^2
Vraies valeurs	0.5	1	0.85	0.5
Moyenne de MLKF	0.4987	1.0412	0.8419	0.4999
Biais de MLKF	0.0013	-0.0412	0.0081	0.001
MSE de MLKF	0.017	0.1603	0.1065	0.0622
T-statistic	0.0099	-0.1029	0.024	0.0004
Moyenne de MM	0.380	0.5633	0.6774	0.2908
MSE de MM	0.0243	0.1356	0.1172	0.0679

Table 2.3: Moyenne, Biais, MSE et T-statistic des paramètres estimés pour n=150

Les meilleurs résultats obtenus au sens des critères (moyenne, biais, MSE, T-statistic), sont ceux fournis par notre algorithme d'estimation MLKF. La T-statistic indique que le biais est non significatif et on observe de manière évidente une amélioration des estimations obtenues par les estimateurs de Guégan. Ce qui nous permet à ce stade d'affirmer que les performances de notre algorithme sont bonnes.

Dans la deuxième étude nous avons considéré les modèles définis par les équations suivantes :

- $X_t = 0.9X_{t-2}e_{t-1} + e_t, \quad e_t \sim iidN(0, \sqrt{0.5})$
- $X_t = 0.99X_{t-2}e_{t-1} + e(t), \quad e_t \sim iidN(0, \sqrt{0.4})$

Les conditions d'inversibilité et de stationnarité sont vérifiées par les coefficients. Dans cette étude nous avons généré 300 séries de taille 100. Les valeurs initiales ont été choisies arbitrairement à partir de la loi uniforme distribuée entre 0 et 1. Les résultats de cette expérience sont dressés dans le tableau suivant

Paramètres	b_{21}	σ^2	b_{21}	σ^2
Vraies valeurs	0.9	0.5	0.99	0.4
Moyenne MLKF	0.9185	0.4977	1.0047	0.4096
Biais MLKF	-0.0185	0.023	-0.0147	-0.0096
MSE MLKF	0.025	0.0061	0.0041	0.0033
T-statistic	-0.117	0.0294	-0.2295	-0.1671

Table 2.4: Moyenne, Biais, MSE et T-statistic des paramètres estimés, les valeurs initiales sont choisies arbitrairement

En conlusion, les deux études montrent que les performances de notre algorithme sont bonnes, et nos estimations améliorent nettement celles obtenus par Guégan.
Dans le paragraphe suivant nous allons montrer à travers une application à la fiabilité des logiciels que notre algorithme est de bonne performance.

2.3 Application en fiabilité des logiciels du modèle bilinéaire superdiagonal d'ordre un

La fiabilité des logiciels suscite, depuis son apparition, un intérêt exceptionnel, vu l'utilisaion répandue des logiciels dans presque tout les secteurs économiques, industriels, sociaux...etc, et vu les conséquences qui peuvent être engendrées suite

aux défaillances de ces entités, plusieurs auteurs se sont interessés à l'étude de ces défaillances.

Au départ, notre attention a été portée à l'application des modèles ARIMA, modèles largement développés et faciles à appliquer en utilisant la méthode de Box et Jenkins (1970), aux temps interdéfaillances du logiciel. Lors de cette étude expérimentale, nous avons observé des explosions au cours du processus des interdéfaillances, en effet, ces explosions sont dûes à la grande durée entre deux défaillances successives, T_i et T_{i+1}, après la correction de la faute qui a engendré la défaillance T_i.

Mais, face à la difficulté d'identification des modèles bilinéaires, question qui reste encore ouverte sauf pour le modèle bilinéaire BL(0,0,P,P) (Oyet (2000)), nous avons pris le modèle superdiagonal d'ordre un, et nous l'avons appliqué aux temps interdéfaillances d'un logiciel.

Dans la suite nous dressons les résultats numériques parus dans Bouzaachane et Benghabrit (2002), et qui concerne l'application des modèles ARIMA aux temps interdéfaillance du logiciel. Ensuite, nous allons rapporter les résultats de l'étude comparative des performances du modèle bilinéaire superdiagonal d'ordre un avec le modèle ICD (Invariant Conditional Distributions) de Chen et Singpurwalla(1999), parus dans Bouzaachane et al.(2006 a).

2.3.1 Aplication des modèles ARIMA

Les données qui font l'objet de cette étude émanent de deux projets de réalisations de logiciels, appelés respectivement SYS1 et SYS2, développés pour le compte de l'US-DOD (département de la défense américaine), et de leur amélioration avec la stratégie de remise en service avec correction immédiate de fautes. Ces données ont été rapportées par Musa (1975), et elles ont été utilisées par Hamlili (1999) lors de son étude comparative des modèles de croissance de fiabilité des logiciels. Nous dressons dans les tableaux (2.5) et (2.6) les données interdéfaillances récoltées durant le test de ces deux logiciels. La représentation graphique est donnée dans la figure

(2.1). Les séries de données seront notées dans la suite par $X1$ et $X2$, et elles sont respectivement de taille 136 et 36.

i	T_i	i	T_i	i	T_i	i	T_i	i	T_i
1	3	28	1146	55	357	82	296	109	875
2	30	29	600	56	193	83	1755	110	245
3	113	30	15	57	236	84	1064	111	729
4	81	31	36	58	31	85	1783	112	4897
5	115	32	4	59	369	86	860	113	447
6	9	33	1	60	748	87	983	114	386
7	2	34	7	61	1	88	707	115	446
8	91	35	227	62	231	89	33	116	122
9	112	36	65	63	330	90	868	117	990
10	15	37	476	64	365	91	724	118	948
11	138	38	58	65	1222	92	2323	119	1082
12	50	39	457	66	543	93	2930	120	22
13	77	40	300	67	10	94	1461	121	75
14	24	41	97	68	16	95	843	122	482
15	108	42	263	69	529	96	12	123	5509
16	40	43	452	70	379	97	261	124	100
17	670	44	255	71	44	98	1800	125	10
18	120	45	197	72	129	99	865	126	1071
19	26	46	193	73	810	100	1435	127	371
20	114	47	6	74	290	101	30	128	790
21	325	48	79	75	300	102	143	129	6150
22	55	49	816	76	529	103	108	130	3321
23	242	50	1351	77	281	104	1	131	1045
24	68	51	148	78	160	105	3109	132	648
25	422	52	21	79	828	106	1247	133	5485
26	180	53	233	80	1011	107	943	134	1160
27	10	54	134	81	445	108	700	135	1864
								136	4116

Table 2.5: Les interdéfaillances du système SYS1

i	T_i	i	T_i	i	T_i	i	T_i
1	191520	10	937620	19	228315	28	91260
2	2074020	11	72240	20	51480	29	1225620
3	514560	12	737700	21	44820	30	120
4	1140	13	250680	22	850080	31	1563300
5	3120	14	2965	23	361860	32	513000
6	327480	15	196	24	39300	33	177660
7	15420	16	65173	25	545280	34	2469000
8	60000	17	2370	26	256980	35	1678260
9	140160	18	1581	27	396780	36	170760

Table 2.6: Les interdéfaillances du système SYS2

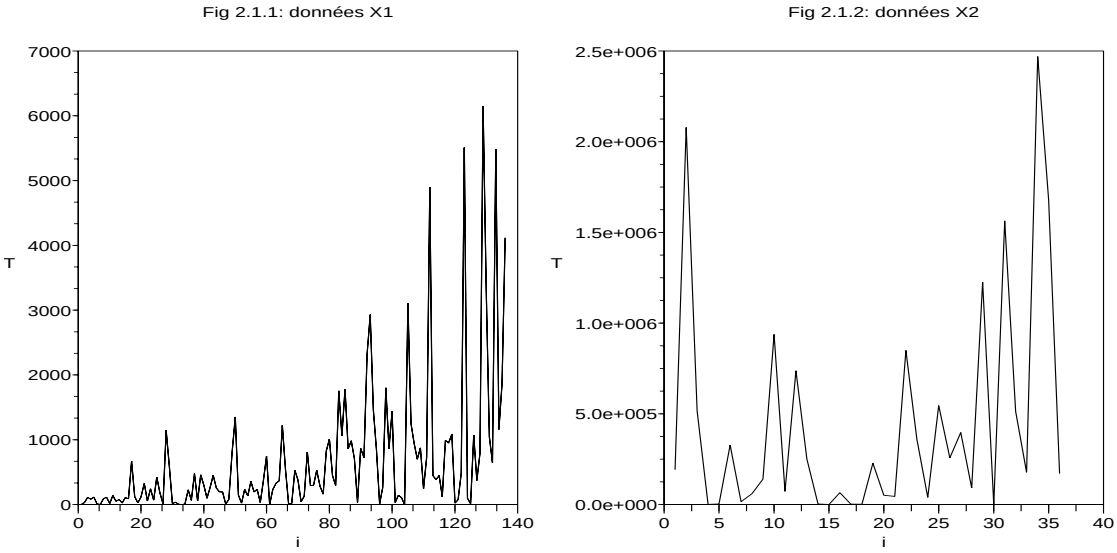

Figure 2.1: Graphique des données représentant les temps interdéfaillances

Ces données récoltées présentent une croissance de fiabilité, car les intervalles des temps interdéfaillances sont de plus en plus grands. La courbe révèle une dispersion des données non stationnaire, ce qui nous a incité à appliquer une transformation logarithmique aux deux séries; cette transformation a suscité deux séries $LX1$ et $LX2$ (voir figure (2.2)), représentant les logarithmes des temps interdéfaillances de $X1$ et $X2$ respectivement.

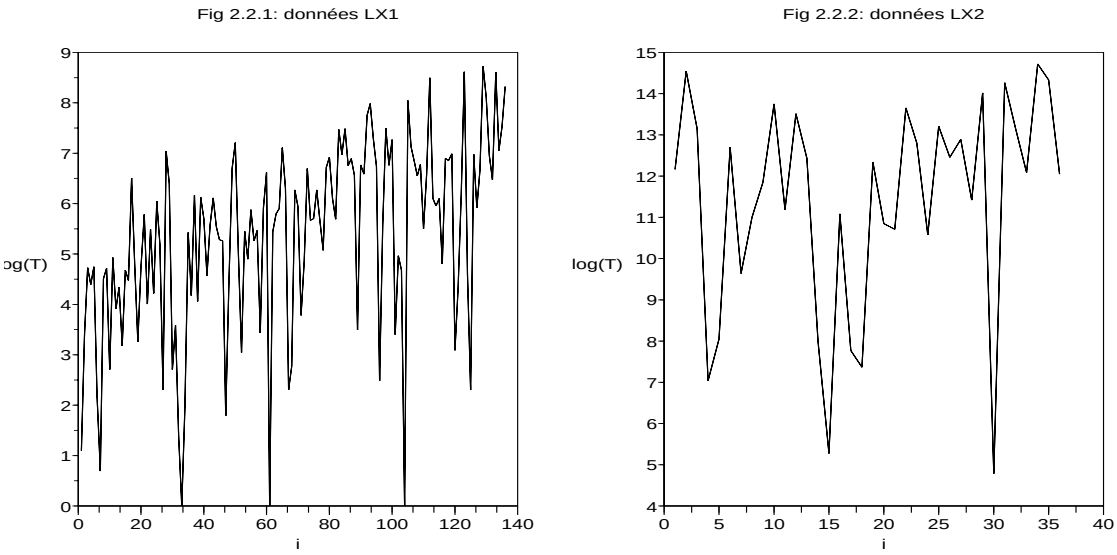

Figure 2.2: Graphique des logarithmes des temps interdéfaillances

Pour chercher le modèle ARIMA(p, d, q) susceptible de représenter notre jeu de données des temps interdéfaillances, nous avons eu recours à la méthode de Box et Jenkins (1970). Cette méthode est constituée de trois étapes distinguées

1. Étape d'identification : dans cette étape on cherche les valeurs plausibles de (p, d, q) à partir des observations, les autocorrélations et les autocorrélations partielles entre ces observations. Généralement, cette étape nous permet de retenir plus qu'un triplet des valeurs (p, d, q).

2. Étape d'estimation : les paramètres des modèles retenus seront estimés, dans cette étape, en utilisant la méthode du maximum de vraisemblance. Ces modèles seront soumis, par la suite, à divers tests statistiques afin de vérifier la compatibilité des résultats avec les hypothèses qu'on a supposées.

3. Étape de vérification : Cette étape est composée de plusieurs tests de vérification qui nous permettent de spécifier les bons modèles et de rejetter ceux qui n'ont pas satisfait un ou tous les tests. Les tests que l'on fait subir au modèle sont de

deux types : tests concernant les paramètres du modèle et ceux concernant les hypothèses faites sur le processus $(e_t)_{t\in\mathbb{Z}}$.

- Test concernant les paramètres :

  Il s'agit de tester la significativité des coefficients φ_p et θ_q(voir chapitre un), ce qui peut être fait au moyen d'un test de type Student. Soient φ_p^* et θ_q^* les estimateurs de φ_p et θ_q respectivement, $V(\varphi_p^*)$ et $V(\theta_q^*)$ leurs variances; on acceptera la modélisation ARMA(p,q) si :$\frac{|\varphi_p^*|}{\sqrt{V(\varphi_p^*}} \geq 1.96$ et $\frac{|\theta_q^*|}{\sqrt{V(\theta_q^*)}} \geq 1.96$.

- Test de la nullité de la moyenne des résidus $(e_t)_{t\in\mathbb{Z}}$

  L'hypothèse à tester est :"la moyenne m des résidus est nulle". Ce test se base sur la moyenne $\overline{e} = \frac{1}{n}\sum_{i=1}^{n} e_i$ comme statistique de test. On peut utiliser $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}$ comme estimation de l'écart-type. D'après le théorème de la limite centrale, si n est assez grand le rapport dit de Student, $Tstatistic = \frac{\overline{e}}{RMSE}$ est approximativement distribué selon la loi normale centrée réduite. On rejette l'hypothèse que m est nulle, au niveau de probabilité de 5%, si $T - statistic < -1.96$ ou $T - statistic > 1.96$.

- Test individuel de bruit blanc :

  Que le processus des (e_t) constitue un processus bruit blanc implique que les autoccorélations $\rho_1 = \rho_2 = \ldots = 0$. Dans ce contexte nous nous intéressons à tester la nullité d'une autocorrélation de retard k quelconque. On met ceci en évidence en écrivant l'hypothèse à tester sous la forme $H_k : \rho_k = 0$. Si le processus est un bruit blanc et pourvu que n soit grand, la statistique $\sqrt{n}\rho_k^*$ a une loi normale centrée réduite. On rejette H_k au niveau de probabilité de 5% si la statistique ρ_k^* est en dehors de l'intervalle $\left[\ \frac{-1,96}{\sqrt{n}}, \frac{1.96}{\sqrt{n}}\ \right]$, dans ce cas l'autocorrélation est dite significative, autrement elle est non significative.

- Test global de bruit blanc : Test portmanteau

  Le test global que nous énonçons concerne les K premières autocorrélations,

où K est choisi à l'avance. On teste l'hypothèse

$$H : \rho_1 = \rho_2 = \ldots = \rho_K = 0$$

La statistique de test due à Box et Pierce est de la forme $BP = n \sum_{h=1}^{K} (\rho_h^*)^2$. Si le processus est de type bruit blanc et si n est assez grand, la distribution de BP est approximativement χ^2 à $K - p - q$ degrés de liberté. On rejette l'hypothèse que le processus est de type bruit blanc, au niveau de probabilité de 5%, si la valeur de BP dépasse le quantile d'ordre 0.95 de la loi χ^2 à $K - p - q$ degrés de liberté.

Il se peut que plusieurs modèles franchissent la phase de vérification et qu'il faille choisir dans cet ensemble. Le choix est alors, on le devine, difficile. Il existe cependant un certain nombre de critères de choix. Dans notre cas nous avons utilisé le carré moyen des erreurs MSE et l'erreur absolue moyenne en pourcentage $MAPE$.
L'application du schéma récursif de Box and Jenkins nous a permis, d'éliminer les modèles qui n'ont pas franchi la phase des tests et ceux qui ont satisfait tous les tests mais dont le pouvoir prédictif est insatisfaisant.
Les modèles que nous avons retenus à la fin de cette procédure sont les suivants
$\star$ Le jeu de données LX1 est modélisé par

$$(I-\phi_6 B^6)(I-\phi_5 B^5)(I-\phi_4 B^4)(I-\phi_3 B^3)(I-\phi_2 B^2)(I-\phi_1 B)(I-B)LX1_t = e_t \quad (2.10)$$

$\star$ Le jeu de données LX2 est modélisé par

$$(I - \phi_5 B^5)(I - B)LX2_t = (I - \theta_1 B)e_t \quad (2.11)$$

Nous constatons des résultats dressés dans les tableaux (2.7) et (2.8) que les statistiques du test des paramètres, en valeur absolue, sont toutes supérieures à 1.96 ce qui montre que les modèles sont acceptables.

Paramètres	Statistique	Estimation
ϕ_1	-7.8	-0.676
ϕ_2	-7.4	-0.755
ϕ_3	-4.8	-0.531
ϕ_4	-4.9	-0.548
ϕ_5	-3.5	-0.355
ϕ_6	-2.5	-0.219

Table 2.7: Représentation des valeurs de la statistique du test et des estimations des paramètres du modèle retenu pour le jeu de données LX1

Paramètres	Statistique	Estimation
θ_1	-2.5	-0.418
ϕ_5	6.5	0.826

Table 2.8: Représentation des valeurs de la statistique du test et des estimateurs des paramètres du modèle retenu pour le jeu de données LX2

Les résultats présentés dans les tableaux (2.9) et (2.10), révèlent qu'aucune information ne peut être extraite des autoccorélations et des autoccorélations partielles, et que le processus des résidus est un bruit blanc puisque les valeurs de la statistique de Box et Pierce ne dépassent pas les valeurs des quantiles d'ordre D.

D	BP	χ^2_D
6	4.37	12.6
12	9.69	21
18	15.96	28.9
24	19.2	36.4
30	23.81	43.8
32	24.34	46.1

Table 2.9: Représentation de l'estimation de la statistique de Box et Pierce et des quantiles d'ordre D de de la loi de χ^2 pour les données LX1

D	BP	χ^2_D
4	0.83	9.49
10	3.89	18.3
12	4.38	21

Table 2.10: Représentation de l'estimation de la statistique de Box et Pierce et des quantiles d'ordre D de la loi de χ^2 pour les données LX2

Les résultats numériques fournis dans le tableau (2.11), indiquent que la moyenne des résidus des deux modèles est significativement nulle puisque la T-statistic en valeur absolue est inférieure à 1.96. Ils indiquent aussi que le pouvoir prédictif des deux modèles est bon puisque la valeur de MAPE est inférieure à 5%. Théoriquement, la performance d'un modèle est très bonne si MAPE est inférieur ou égale à 5 %.

Critères	LX1	LX2
T-statistic	-0.24	-0.79
Moyenne	-0.03	-0.349
Observation	8.32	12.04
Prévision	8.247	11.76
Erreur	0.073	0.28
Erreur en %	0.8	2
MAPE	0.92	2.3
MSE	0.00529	0.038

Table 2.11: Résultats numériques de quelques critères statistiques pour LX1 et LX2

En conclusion, nous avons montré à travers les résultats statistiques et graphiques, que les modèles obtenus ont de bonnes performances, ils nous ont permis de prédire les temps interdéfaillances des systèmes logiciels "SYS1" et "SYS2", en conséquence, ils peuvent être utilisés pour prévoir leurs prochain temps interdéfaillances, de corriger la faute qui a causé leurs défaillances et ainsi contribuer à améliorer la fiabilité de ces deux systèmes logiciels.

L'observation des explosions dans les représentations graphiques des temps interdéfaillances des logiciels "SYS1" et "SYS2", a suscité l'idée d'utiliser le modèle superdiagonal d'ordre un pour modéliser les jeux de données issus de ces deux systèmes,

mais, les résultats obtenus, lors de l'ajustement n'étaient pas bons. Nous avons ensuite essayé un autre jeu de données, dont les résultats de l'étude expérimentale que nous avons menée, seront exposés dans le paragraphe suivant.

2.3.2 Application du modèle bilinéaire superdiagonal d'ordre un

Les données qui font l'objet de cette étude, émanent d'un projet réel de réalisation du logiciel appelé "système 40". Lors du test de ce logiciel, chaque défaillance détectée, est enregistrée et corrigée simultanément. Après chaque opération de correction on obtient une version améliorée du logiciel "système 40".
Dans le tableau (2.12), nous dressons les données des temps interdéfaillances reportées par Musa (1979).

i	T_i	i	T_i	i	T_i	i	T_i	i	T_i
1	510	22	12744.48	43	3246.33	64	182153.15	85	642753.98
2	1204.36	23	11043.04	44	3563.77	65	204574.95	86	813990.24
3	22058.89	24	8872.73	45	1699.38	66	96365.12	87	651416.14
4	23924.2	25	51652.02	46	19390.86	67	69572.82	88	1555295.16
5	20255.5	26	38217.43	47	59924.55	68	65522.52	89	729633.94
6	25632.41	27	17364.48	48	69002.42	69	316144.1	90	482663.53
7	48750.13	28	14942.98	49	37764.68	70	390860.27	91	381221.35
8	88110.8	29	64362.37	50	16920.65	71	329747.45	92	235176.38
9	244956.63	30	53669.84	51	15401.1	72	330965.95	93	182481.61
10	290003.82	31	45508.51	52	13336.63	73	280318.57	94	1634926.93
11	110755.53	32	21629.86	53	32765.46	74	254832.05	95	678584.4
12	65888.62	33	22453.09	54	35197.25	75	123497.4	96	287712.72
13	46894.56	34	24529.41	55	267522.66	76	291172.69	97	298391.67
14	21388.95	35	42373.38	56	114274.11	77	1935391.9	98	431429.47
15	16430.52	36	65608.08	57	65709.57	78	1351849.4	99	626867.43
16	16066.28	37	31062.48	58	41151.91	79	629965.61	100	551257.58
17	11269.56	38	89922.38	59	20532.78	80	2591260.15	101	266743.05
18	13728.46	39	52177.42	60	42022.25	81	3667195.44		
19	19914.57	40	24836.98	61	64982.47	82	1677352.01		
20	19629.38	41	14455.37	62	65464.91	83	731019.82		
21	13669.69	42	5929.40	63	254521.68	84	477826.72		

Table 2.12: Les interdéfaillances du système 40

Ces données sont représentées dans la figure (2.3) ci-dessous. Cette figure montre que l'interval du temps entre deux défaillances croît considérablement, ceci résulte du processus de corrections que subit le logiciel durant son test. Ces corrections ont pour objectif la suppression des fautes qui ont engendré ces défaillances, et obtenir par la suite une version améliorée du logiciel.

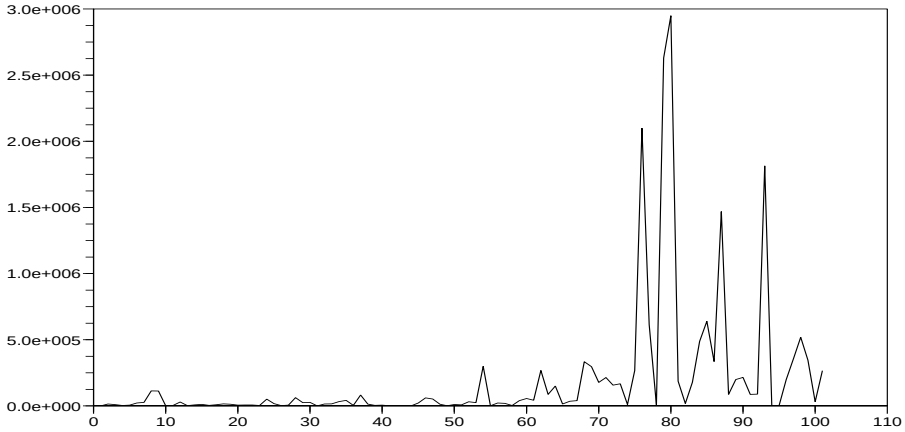

Figure 2.3: Représentation graphique des temps interdéfaillances du logiciel "système 40"

Pour simplifier les calculs nous avons appliqué une transformation logarithmique à ce jeu de données, ce qui a suscité des données représentées sur la figure (2.4).

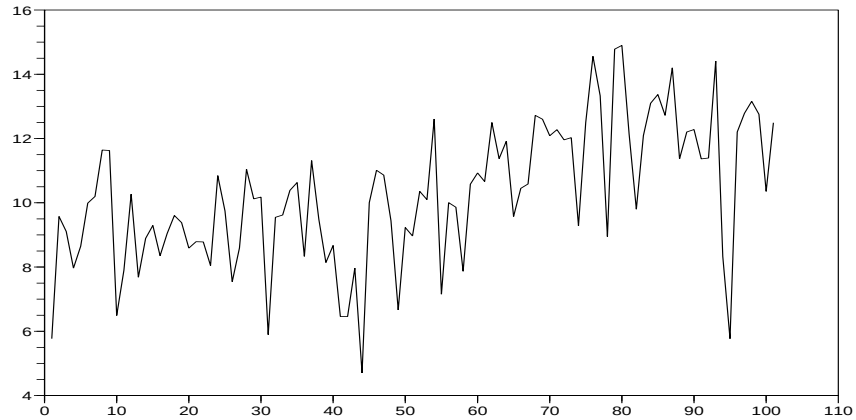

Figure 2.4: Représentation graphique du logarithme des temps interdéfaillances du logiciel "système 40"

À partir de ces données transformées nous avons utilisé notre algorithme MLKF décrit précédemment, pour trouver les valeurs estimées de b_{21} et σ^2, ces valeurs sont respectivement **0.054177** et **54.325463**. Il est évident que ces deux valeurs vérifient les conditions de stationnarité et d'inversibilité.

Nous avons ensuite ajusté le modèle BL(0,0,2,1) à ces données pour calculer les prédictions. Dans la figure (2.5) nous représentons les prédictions entre le $(50)^{\text{ème}}$ et le $(101)^{\text{ème}}$ temps inter-défaillance, superposées aux données.

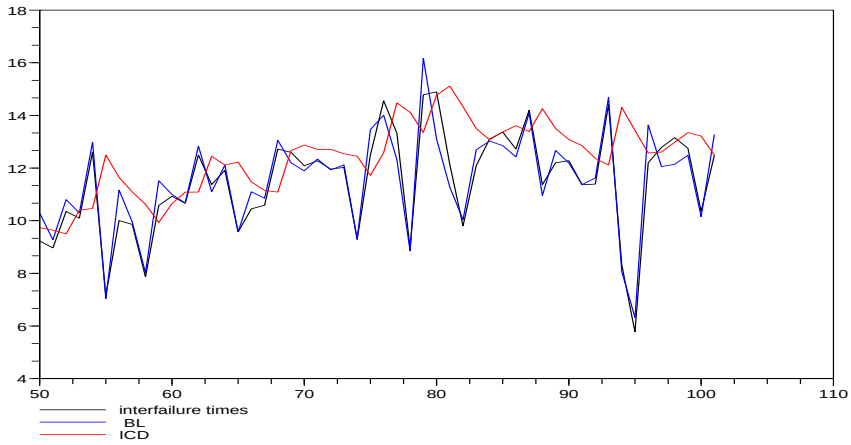

Figure 2.5: Représentation graphique du logarithme des temps interdéfaillances du logiciel "système 40" et les prédictions obtenues par le modèle bilinéaire superdiagonal d'ordre un et le modèle ICD

Cette figure révèle que le modèle BL(0,0,2,1) est significativement meilleur que le modèle ICD.

Pour plus de justification nous avons calculé quelques critères de comparaison présentés dans le tableau (2.13).

Critères	BL(0,0,2,1)	ICD
Erreur moyenne	-0.0861337	-0.9692787
Variance des erreures(VarE)	0.3679005	4.1934257
MAPE	0.0390292%	0.1232053%
100×(VarE/Variance des données)	1.052579%	11.997569%

Table 2.13: Comparaison des Critères obtenus pour les deux modèles

En parcourant ce tableau, on remarque que les valeurs numériques des critères, relatives au modèle bilinéaire BL(0,0,2,1), sont inférieures à celles obtenues pour le modèle ICD. En plus, les valeurs de MAPE et le quotient de la variance des erreurs par la variance des données en pourcentage, sont respectivement inférieures à 5 % et à 10%, ceci montre que notre modèle est plus adéquat que le modèle ICD à modéliser les temps interdéfaillances du "système 40".

En conclusion, cette étude nous a permis de montrer simultanément les bonnes performances de notre algorithme d'estimation par maximum de vraisemblance et la bonne qualité prédictive du modèle bilinéaire superdiagonal d'ordre un par rapport au logiciel considéré.

2.4 Conclusion

Les différents résultats numériques relatés dans ce chapitre, ont montré que notre nouvelle approche d'estimation est performante, pour le modèle en question, car elle a distinctement amélioré les estimations issues des estimateurs de Guégan, et elle nous a permis d'obtenir de bonnes estimations des paramètres du modèle BL(0,0,2,1) que nous avons ajusté à certaines données de fiabilité des logiciels.

Dans les chapitres suivants, nous allons montrer que cette approche est prometteuse, via d'autres modèles bilinéaires particuliers.

Chapitre 3

Estimation des paramètres du modèle bilinéaire diagonal d'ordre un BL(1,0,1,1) : Application au Séisme d'Al Hoceima (Février 2004)

3.1 Introduction

La question d'estimation des paramètres du modèle bilinéaire diagonal d'ordre un a été abordée par Kim et Billard (1990), qui, à partir de la fonction d'autocovariance ont conçu les estimateurs de ces paramètres, et ils ont montré leurs convergences presque sûr. Cependant l'extension de leur méthode d'estimation à d'autre modèles bilinéaires s'avère très difficile eu égard à la complexité des calculs.

Nous avons vu dans le chapitre précédent un algorithme d'estimation des paramètres du modèle bilinéaire superdiagonal d'ordre un, dans ce chapitre, nous allons présenter la conception de cet algorithme pour le modèle bilinéaire diagonal d'ordre un, et

nous justifierons ces bonnes performances à travers des simulations de Monte Carlo. Ensuite nous allons évoquer les résultats expérimentaux d'une application du modèle bilinéaire diagonal d'ordre un aux données issues du séisme qui a attaqué Al Hoceima en Février 2004. Ce chapitre a fait l'objet d'une communication (Bouzaachane et al(2005)) et d'un article soumis (Bouzaachane et al(2006c)).

3.2 Algorithme d'estimation des paramètres du modèle bilinéaire diagonal d'ordre un

Dans ce paragraphe, nous allons, en premier lieu, présenter le modèle bilinéaire diagonal d'ordre un et ces propriétés de stationnarité et d'inversibilité. Ensuite nous décrirons l'algorithme d'estimation de ces paramètres.

3.2.1 Modèle bilinéaire diagonal d'ordre un

Un modèle bilinéaire diagonal d'ordre un noté $BL(1,0,1,1)$ est une équation aux différences stochastiques non linéaire de la forme

$$X_t = a_1 X_{t-1} + b_{11} X_{t-1} e_{t-1} + e_t. \tag{3.1}$$

où $(e_t, t \in \mathbb{Z})$ est une suite de variables aléatoires indépendantes et équidistribuées de moyenne nulle et de variance σ^2 finie.
Le processus $(X_t, t \in \mathbb{Z})$ défini par l'équation (3.1) est dit processus bilinéaire diagonal d'ordre un.
Les propriétés probabilistes, existence et stationnarité, et inversibilité de ce modèle ont été étudiées par Pham et Tran (1981), elles sont données dans les théorèmes suivants :

Théorème 6 *Soit une suite de variables aléatoires indépendantes équidistribuées* $(e(t), t \in \mathbb{Z})$, *centrées, de variance* $\sigma^2 < \infty$, *admettant des moments d'ordre qua-*

tre finis, définis sur un certain espace de probabilité $(\Omega, \mathcal{A}, \mathcal{P})$*, alors une condition nécessaire et suffisante d'existence d'un processus strictement stationnaire de carré intégrable* $(X_t, t \in \mathbb{Z})$ *vérifiant* (3.1) *est*

$$a_1^2 + b_{11}^2\sigma^2 < 1. \tag{3.2}$$

Si la condition (3.2) est vérifiée, alors X_t est unique et donné par l'expression

$$X_t = e_t + \sum_{j=1}^{+\infty}[\prod_{k=1}^{j}(a_1 + b_{11}e_{t-k}]e_{t-j}, \tag{3.3}$$

qui converge fortement et en moyenne quadratique.

Théorème 7 *Si* $\mid a_1 \mid < 1$ *et* $(a_1^2 + b_{11}^2\sigma^2) \leq 1$ *alors il existe un seul processus strictement stationnaire* $(X_t, t \in \mathbb{Z})$ *satisfaisant* (3.1)

Théorème 8 *Le modèle* (3.1) *est inversible en* $\vartheta = (a_1, b_{11})$ *relativement au processus observé* X_t*, si*

$$\mid b_{11} \mid < exp[-E(log \mid X_t \mid)]. \tag{3.4}$$

Dans ce cas

$$e_t^{\vartheta} = X_t - \sum_{j=1}^{t-1}(-b_{11})^{j-1}(\prod_{k=1}^{j} X_{t-k})(a_1 + b_{11}X_{t-j}).$$

En effet, si la condition (3.4) est établie la série

$$X_t - \sum_{j=1}^{\infty}(-b_{11})^{j-1}(\prod_{k=1}^{j} X_{t-k})(a_1 + b_{11}X_{t-j})$$

est presque sûrement convergente avec une somme égale à e_t^{ϑ} et nous avons $e_{t|z_0}^{\vartheta}$-$e_t^{\vartheta} \rightarrow$ 0 quand $t \rightarrow \infty$, avec

$$e_{t|z_0}^{\vartheta} = X_t - \sum_{j=1}^{t-1}(-b_{11})^{j-1}(\prod_{k=1}^{j} X_{t-k})(a_1 + b_{11}X_{t-j}) - (-b_{11})^{t-1}(\prod_{k=1}^{t} X_{t-k})z_0.$$

Pour plus de détail sur la démonstration je vous invite à consulter le livre de Guégan (1993).

3.2.2 Conception de l'algorithme

Les étapes de conception de l'algorithme d'estimation des paramètres du modèle bilinéaire diagonal d'ordre un (3.1) sont identiquement les mêmes que celles de l'algorithme d'estimation des paramètres du modèle superdiagonal d'ordre un (2.1). Dans ce paragraphe, nous présentons, sommairement, la conception de l'algorithme d'estimation.

La première étape de conception de l'algorithme est constituée de la présentation du modèle (3.1) sous forme espace d'état défini par :

$$\begin{cases} \xi_{t+1} &= A(e_t)\xi_t + v_{t+1} & : \text{équation d'état} \\ X_t &= H\xi_t & : \text{équation d'obseration} \end{cases} \tag{3.5}$$

où le vecteur d'état est $\xi_t = \begin{bmatrix} X_t \\ X_{t-1} \end{bmatrix}$, $v_t = [e_t, 0]'$, $H = [1, 0]$ et

$A(e_t) = \begin{pmatrix} a_1 + b_{11}e_t & 0 \\ 1 & 0 \end{pmatrix}$.

Lorsque la condition (3.4) est vérifiée, $A(e_t)$ devient une fonction de X_k pour $k = 1, \ldots, t$, ceci découle du fait que $e_t - e_t^{\vartheta}$ converge presque sûrement vers 0.

Soit $\theta = (a_1, b_{11}, \sigma^2)$ le vecteur des paramètres à estimer et $(x_1, \ldots, x_n)$ les réalisations du processus bilinéaire BL(1,0,1,1), $(X_1, \ldots, X_n)$. L'expression de la log-vraisemblance du processus $X_1, \ldots, X_n$ est identique à celle (2.9) du processus bilinéaire BL(0,0,2,1), nous la rappelons ci-dessous :

$$L(x;\theta) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\ln(\hat{M}_{t|t-1}) - \frac{1}{2}\sum_{t=1}^{n}\frac{(x_t - \hat{x}_{t|t-1})^2}{\hat{M}_{t|t-1}} \tag{3.6}$$

Les autres étapes de l'algorithme d'estimation des paramètres du modèle BL(1,0,1,1)

sont semblables à celle de l'algorithme d'estimation des paramètres du modèle BL(0,0,2,1), la différence réside en les valeurs initiales, de l'algorithme d'estimation d'une part et du filtre de Kalman d'autre part.

En effet, les valeurs initiales de notre algorithme ont été choisies arbitrairement et les valeurs initiales de l'algorithme du filtre de Kalman, à savoir $\hat{\xi}_{1|0}$ et $P_{1|0}$, sont données par

$$\hat{\xi}_{1|0} = \begin{bmatrix} \frac{b_{11}\sigma^4}{1-a_1} \\ \frac{b_{11}\sigma^4}{1-a_1} \end{bmatrix},$$

$$P1 = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix},$$

où $\lambda = (\frac{\sigma^2}{1-(a_1^2)-b_{11}^2\sigma^4})(1 + \frac{4a_1}{1-a_1-b_{11}^2\sigma^4}) - \frac{b_{11}^2\sigma^4}{(1-a_1)^2}$

Pour approuver la qualité des estimateurs des paramètres du modèle BL(1,0,1,1), obtenues par notre algorithme MLKF, nous l'avons mis en comparaison avec ceux obtenus par la méthode d'autocovariance. Soient n observations $x_1, \ldots, x_n$, du modèle (3.1), on définit la moyenne, la fonction d'autocovariance empirique et la fonction d'autocorrélation empirique par

$$\overline{x} = \frac{1}{n}\sum_{t=1}^{n} x_t$$

$$\hat{\gamma}(h) = \frac{1}{n}\sum_{t=1}^{n-h}(x_t - \overline{x})(x_{t+h} - \overline{x})$$

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}$$

Kim et Billard (1990) ont proposé de prendre comme estimateurs des paramètres du modèle (3.1), a_1 et b_{11}, les suivants

$$\hat{a}_1 = \frac{\hat{\gamma}(2)}{\hat{\gamma}(1)},$$

$$\hat{b}_{11} = \frac{\overline{X}(1-\hat{a}_1)}{\hat{\sigma}^2} = \frac{2\overline{X}}{\hat{\gamma}(0) - \hat{\gamma}(1) + \sqrt{\alpha}},$$

où

$\alpha = -3\hat{\gamma}^2(1) - 6\hat{\gamma}(0)\hat{\gamma}(1) + \hat{\gamma}(0) - 4\hat{\gamma}(2)\frac{\hat{\gamma}^2(1)+\hat{\gamma}(0)\hat{\gamma}(1)+2\hat{\gamma}^2(0)}{\hat{\gamma}(1)-\hat{\gamma}(2)}$.

L'avantage de ces estimateurs est qu'ils convergent presque sûrement et en loi.

Dans le paragraphe suivant nous allons expliciter les résultats numériques obtenus à travers l'étude par simulation de notre algorithme d'estimation des paramètres du modèle BL(1,0,1,1), que nous allons confronter aux valeurs des estimateurs de Kim et Billard.

3.3 Simulations

Ce paragraphe est consacré à l'évaluation numérique des performances de notre algorithme d'estimation des paramètres du modèle BL(1,0,1,1) et de le situer par rapport à celui de Kim et Billard.

L'étude par simulation menée, se repose sur deux modèles bilinéaires diagonal d'ordre un, suivants :

$$\begin{cases} (a) \quad X_t = 0.5X_{t-1} + 0.85X_{t-1}e_{t-1} + e_t \quad e_t \sim iidN(0,1) \\ (b) \quad X_t = 0.9X_{t-1} + 0.1X_{t-1}e_{t-1} + e_t \quad e_t \sim iidN(0,1) \end{cases}$$

Le modèle (a) a été considéré par la recherche des valeurs approchées de $\theta = (0.5, 0.85, 1)$ via notre algorithme, alors que le modèle (b) a été considéré par la recherche des valeurs approchées de $\theta = (0.9, 0.1)$ en considérant la variance de e_t constante égale à 1.

Dans un premier temps, nous fixons la variance de e_t et nous estimons les paramètres a_1 et b_{11}. L'expérience menée est constituée de 500 réplications dans chaque réplication nous générons des échantilons de taille 50 et 150.

Les résultats de cette expérience sont dressés dans les tables (3.1) et (3.2), où KB

désigne la méthode de Kim et Billard.

Parmètres	a_1	b_{11}
Vraies valeurs	0.9	0.1
Mean de MLKF	0.9013	0.0958
Biais de MLKF	-0.0013	0.0042
MSE de MLKF	0.0025	3.6923e-4
T-statistic	-0.0259	0.2165
Mean de KB	0.8614	0.1012

Table 3.1: Moyenne, Biais, MSE et T-statistic des paramètres estimés pour n=50

Parmètres	a_1	b_{11}
Vraies valeurs	0.9	0.1
Mean de MLKF	0.9001	0.0987
Biais de MLKF	-1.0e-4	0.0013
MSE de MLKF	1.5893e-4	5.685e-4
T-statistic	-0.0079	0.0545
Mean de KB	0.8566	0.064

Table 3.2: Moyenne, Biais, MSE et T-statistic des paramètres estimés pour n=150

Dans un second temps, nous estimons les paramètres a_1 et b_{11} et σ^2. Pour ce cas, nous avons généré des séries de tailles $n = 50, 150$ de même nombre de réplications qui est 500. Les résultats ainsi obtenus sont résumés dans les tables (3.3) et (3.4).

Parmètres	a_1	b_{11}	σ^2
Vraies valeurs	0.5	0.85	1
Mean de MLKF	0.4987	0.8477	0.9904
Biais de MLKF	0.0013	0.0023	0.0096
MSE de MLKF	0.0073	4.386e-4	0.0004
T-statistic	0.0152	0.1098	0.48
Mean de KB	0.1086	0.2404	12.4095

Table 3.3: Moyenne, Biais, MSE et T-statistic des paramètres estimés pour n=50

Parmètres	a_1	b_{11}	σ^2
Vraies valeurs	0.5	0.85	1
Mean de MLKF	0.4995	0.8496	0.9958
Biais de MLKF	5.0e-4	4.0e-4	0.0042
MSE de MLKF	2.493e-4	3.581e-4	0.0082
T-statistic	0.0317	0.0211	0.0464
Mean de KB	0.3515	0.2616	4.2865

Table 3.4: Moyenne, Biais, MSE et T-statistic des paramètres estimés pour n=150

À travers les résultats illustrés dans les tables ci-dessus, on constate que la méthode de Kim et Billard est nettement moins performante que notre nouvelle technique d'estimation. La non significativité, des moyennes carrés des erreurs des biais ainsi que T-statistic relatifs à notre algorithme, prouve sans conteste l'efficacité de notre algorithme.

3.4 Application au séisme d'Al Hoceima

Les tremblements de terre sont dûs à des instabilités frictionnelles, dans la lithosphère, liées aux contraintes crées par le mouvement des plaques tectoniques. Ces mouvements localisés sur les failles, aux frontières des plaques, ne sont plus réguliers. Les failles restent bloquées pendant de longues périodes, tandis que le mouvement des plaques se poursuit, de part et d'aures. Lors d'un séisme, la faille cède soudainement à la volonté tenace, lente et continue des contraintes tectonique. Cette rupture se propage avec une vitesse de plusieurs kilomètres par seconde et fait glisser en quelques secondes les deux

compartiments de la faille l'un par rapport à l'autre.
Récemment, les tremblements de terre sont survenus près de grandes agglomérations comme la ville d'Al-Hoceima. Ce tremblement de terre a été provoqué, indiquent les experts, par la rencontre de la plaque africaine avec le bloc ibérique, un morceau un peu indépendant de la plaque Eurasie. Cette interaction provoque un mouvement nord-ouest de 3-4 mm par an.
Les conséquences sont d'énormes dégâts et de nombreuses personnes tuées. Fautes de prédire les séismes et encore moins de les contrôler nous sommes obligées de les subir. Ce chapitre concerne la modélisation et la prédiction des plus grandes magnitudes du séisme qui a frappé la ville d'Al-Hoceima en Février 2004, par le modèle bilinéaire BL(1,0,1,1) (3.1). La première partie sera consacrée à la présentation des données que nous avons utilisées, et la deuxième partie aux résultats obtenus.

3.4.1 Données séismiques

L'étude repose sur les données recensées en Février (2004) au cours du tremblement de terre qui a secoué Al-Hoceima, par l'Institut Géographique National : National Geographic Institute. Depuis 24 Février 2004, la date de la plus grande magnitude du tremblement qui a secoué AL-Hoceima, les secousses ont été répétées continuement avec des magnitudes différentes. Nous avons remarqué que dans l'intervalle de temps qui sépare deux heures successives, Al-Hoceima est secouée trois à six fois avec une différence, parfois remarquable parfois non, des magnitudes. Ainsi, nous avons eu l'idée de construire un échantillon en s'intéressant uniquemment à la secousse de haute magnitude entre deux heures successives. La table (3.5) représente les données de l'échantillon ainsi construit.

i	X_i	i	X_i	i	X_i	i	X_i
1	4.3	31	3.2	61	3.4	91	3.5
2	4	32	4.4	62	4	92	3.4
3	4.5	33	3.5	63	3	93	3.3
4	3.9	34	5.2	64	3.7	94	5
5	4.3	35	3.8	65	4.6	95	3.5
6	4.4	36	3.6	66	3.7	96	3.2
7	3.8	37	3.6	67	3.1	97	3.2
8	4.6	38	4.5	68	5.1	98	2.7
9	4	39	3.7	69	3.4	99	3.3
10	4.9	40	3.7	70	3.5	100	3.3
11	3.7	41	3.9	71	3.6	101	4.1
12	3.9	42	3.3	72	3.2	102	3.1
13	4.1	43	3.6	73	3.8	103	3
14	4	44	3.8	74	3.9	104	3.5
15	4.7	45	3.4	75	3.6	105	2.8
16	3.5	46	3.6	76	3.6	106	3.5
17	4.7	47	3.1	77	3.9	107	4
18	3.5	48	3.2	78	4.8	108	3.7
19	4.7	49	3.7	79	3.5	109	3.3
20	3.9	50	3.9	80	3.3	110	3.1
21	3.4	51	3.5	81	3.8	111	2.8
22	4	52	4.2	82	3.6	112	3.6
23	3.2	53	4.2	83	3.3	113	3
24	3.5	54	3.2	84	3.1	114	3.6
25	3.7	55	5.4	85	3.8		
26	3.5	56	3.9	86	3.5		
27	4	57	3.4	87	3.5		
28	5.1	58	3	88	3		
29	3.1	59	3.4	89	3		
30	4.1	60	3.6	90	3.4		

Table 3.5: Les données de L'échantillon extrait du séisme d'Al-Hoceima

Aux données dressées dans la table ci-dessus, nous avons appliqué une transformation logarithmique et une différence d'ordre un. Les données suscitées sont représentées dans la figure (3.1)

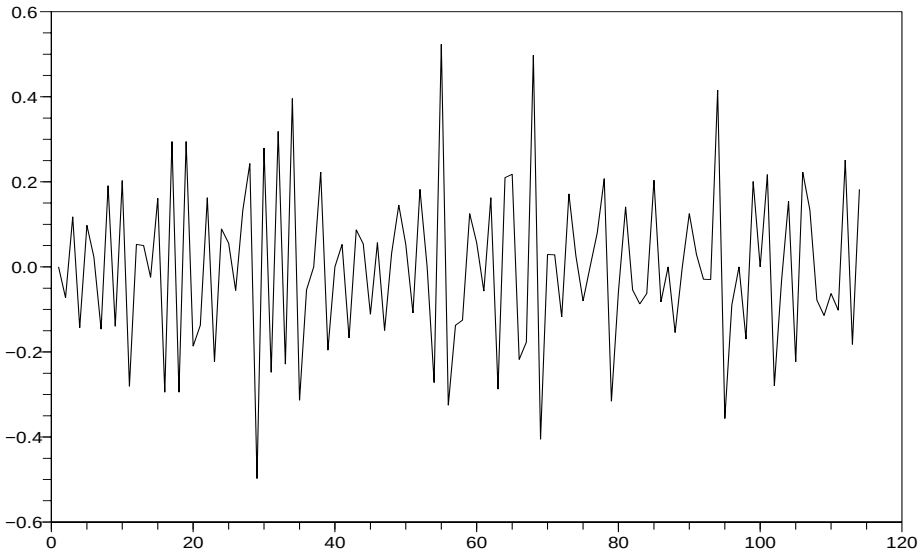

Figure 3.1: Représentation graphique des données transformées du tremblement de terre d'Al-Hoceima

3.4.2 Résultats expérimentaux

À l'aide de notre algorithme d'estimation MLKF, nous avons approché les valeurs des paramètres du modèle bilinéaire diagonal d'ordre un (3.1), nous avons obtenu les valeurs suivantes

$$\hat{a}_1 = 0.0517, \hat{b}_{11} = 0.499, \hat{\sigma}^2 = 2.64,$$

ainsi nous pouvons écrire le modèle bilinéaire résultant par

$$X_t = 0.0517X_{t-1} + 0.499X_{t-1}e_{t-1} + e_t,$$

nous le notons MOD1.

En effet, nous avons utilisé la méthode de Kim et Billard pour estimer les valeurs

initiales de l'algorithme MLKF. Ces valeurs sont données numériquement par

$$\tilde{a}_1 = -0.198229, \tilde{b}_{11} = -0.000949, \tilde{\sigma}^2 = 1.967418,$$

le modèle obtenu avec ces valeurs est

$$X_t = -0.198229 X_{t-1} + -0.000949 X_{t-1} e_{t-1} + e_t,$$

nous le notons MOD2.

Dans les figures (3.2) et (3.3), nous présentons graphiquement la superposition des données du tremblement et les prédictions obtenues par les deux modèles, MOD1 et MOD2.

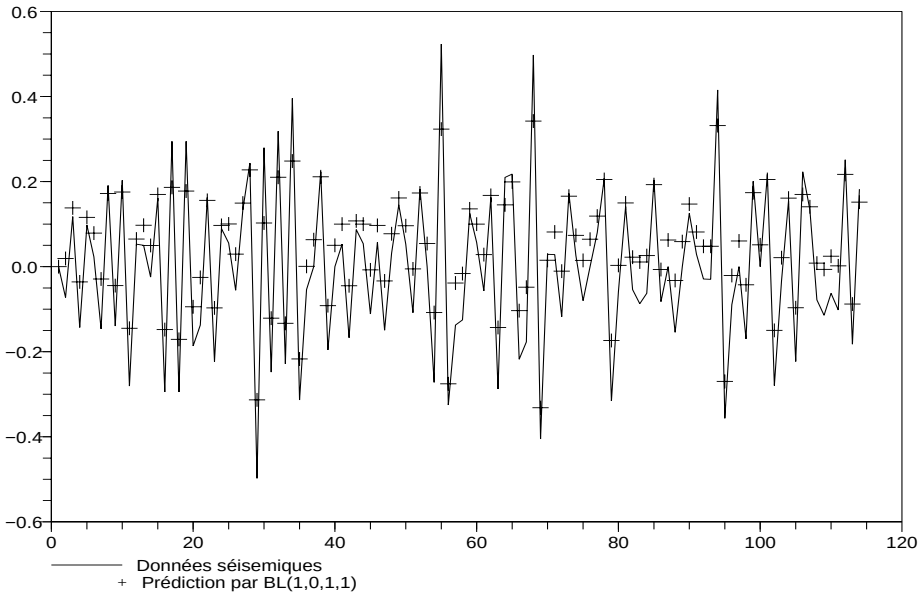

Figure 3.2: Superposition des données transformées du tremblement d'Al-Hoceima et les prédictions par le modèle bilinéaire diagonal MOD1.

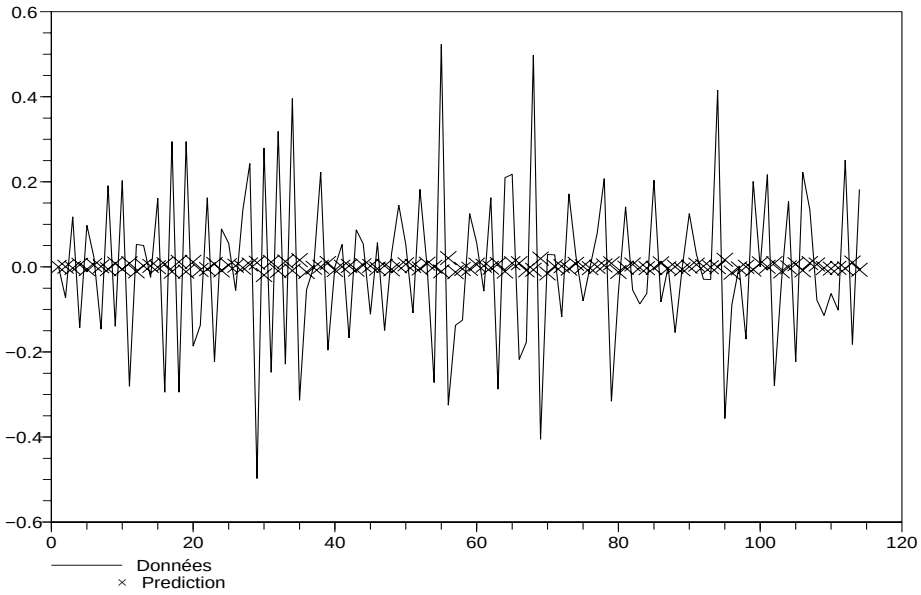

Figure 3.3: Superposition des données transformées du tremblement d'Al-Hoceima et les prédictions par le modèle bilinéaire diagonal MOD2.

Ces figures montrent que MOD1 a efficacement amélioré les prédictions obtenus par le modèle MOD2. Les valeurs numériques des critères statistiques, relatives au modèle MOD1, relatés dans la table (3.6), sont satisfaisantes et encourageantes dans le sens d'exploiter d'autres modèles bilinéaires au profit des séismes.

Critères	Valeurs Estimées
Variance des Données	0.04
Variance des Erreurs	0.005
VarE/VarD	0.12
MSE	0.007
Moyenne des erreurs	-0.047

Table 3.6: Valeurs des Critères Statistiques relatives au modèle MOD1

3.5 Conclusion

Dans ce chapitre, nous avons proposé le modèle bilinéaire BL(1,0,1,1) pour modéliser

les données issues du tremblement de terre d'Al-Hoceima et nous avons utilisé notre nouvel algorithme d'estimation des paramètres du modèle bilinéaire diagonal d'ordre un pour approcher les paramètres de ce modèle.

À travers les simulations nous avons exhibé l'efficacité et la bonne performance de cet algorithme.

Dans le chapitre suivant, nous allons concevoir un algorithme d'estimation des paramètres des modèles bilinéaires diagonaux purs d'ordre p, cet algorithme adoptera une autre représentation d'espace d'état que celle proposée pour les modèles BL(0,0,2,1) et BL(1,0,1,1) et utilise la méthode de recuit simulé pour minimiser la log-vraisemblance. Nous présenterons aussi notre application, développée en Java.

Chapitre 4

Estimation des paramètres du modèle bilinéaire diagonal pur d'ordre p BL(0,0,p,p) et conception d'une application

4.1 Introduction

L'estimation des paramètres du modèle bilinéaire diagonal pur d'ordre p par la méthode des moindres carrés, a été abordée par Guégan et Pham (1989), le résultat obtenu fut généralisé par Houfaidi et Benghabrit (1995a, 1995b, 1998a) en introduisant les familles des M-estimateurs et GM-estimateurs.

L'objectif de ce chapitre, est la présentation de notre nouvel algorithme d'estimation pour le modèle bilinéaire diagonal pur d'ordre p.

Une fois faite la présentation et la spécification du modèle d'état du modèle bilinéaire en question, il s'agit d'utiliser l'algorithme de Kalman pour calculer la log-vraisemblance et ensuite la maximiser par la méthode du recuit simulé. Cette procédure sera résumée dans un algorithme d'estimation.

À partir des cas particuliers du modèle bilinéaire diagonal pur d'ordre p, nous conduirons une série de simulations pour examiner la performance de cet algorithme.
Nous présenterons aussi une description du logiciel que nous avons conçu et qui s'intéresse d'une part à la simulation d'un modèle bilinéaire et à l'estimation de ces paramètres par notre nouvelle technique.

4.2 Algorithme d'estimation des paramètres du modèle bilinéaire diagonal pur d'ordre p

Cette partie de notre travail est attribuée aux modèles bilinéaires diagonaux amputés de leurs termes autorégressifs. Pour ces modèles, nous avons adopté une nouvelle démarche dans la conception de notre algorithme. Cette démarche est constituée de l'utilisation de la méthode du recuit simulé, décrite au chapitre un, au lieu de la méthode de Powell, et de la définition de son espace d'état autrement.
Dans la suite, après la définition du modèle bilinéaire diagonal pur d'ordre p et ces propriétés probabilistes constituées de la stationnarité et l'inversibilité, nous évoquerons l'algorithme d'estimation de ces paramètres.

4.2.1 Modèle bilinéaire diagonal pur d'ordre p

Un modèle bilinéaire diagonal pur d'ordre p noté $BL(0,0,p,p)$ est une équation aux différences stochastiques non-linéaire de la forme

$$X_t = \sum_{i=1}^{p} b_{ii} X_{t-i} e_{t-i} + e_t. \tag{4.1}$$

où $(e_t, t \in \mathbb{Z})$ est une suite de variables aléatoires indépendantes et équidistribuées de moyenne nulle et de variance σ^2 finie. Le processus $(X_t, t \in \mathbb{Z})$ défini par l'équation (4.1) est dit processus bilinéaire diagonal pur d'ordre p.
Dans le chapitre un, nous avons présenté le théorème de Liu et Brockwell (1988), où la

condition d'existence et de stationnarité d'un processus vérifiant l'équation (1.2), a été donné. Nous allons reprendre cette condition pour le modèle (4.1) objet de ce chapitre. Sous les hypothèses faites au théorème 3, la condition d'existence d'un processus $(X_t, t \in \mathbb{Z})$ stationnaire et ergodique vérifiant l'équation (4.1) est :

$$\lambda = \varrho(\Gamma) < 1,$$

avec

$$\Gamma = \begin{bmatrix} \sigma^2 B_1 \otimes B_1 & 0 & B_2 \otimes B_2 \\ 0 & \sigma^2(B_2 \otimes B_1 + B_1 \otimes B_2) & 0 \\ \sigma^2(\gamma^4 B_1 \otimes B_1) & 0 & \sigma^2(B_2 \otimes B_2) \end{bmatrix},$$

et où

$$B_1 = \begin{bmatrix} b_{11} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix} \text{ et } B_2 = \begin{bmatrix} 0 & b_{22} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix}, \text{ sont des matrices } (p \times p).$$

Concernant la propriété d'inversibilité, elle peut être déduite de la proposition de Liu (1990 b), qui se présente comme suit :

$$E(\log \| \prod_{j=1}^{p} B(t-j) \|) < 0,$$

Où, en se situant dans le cas du modèle bilinéaire diagonal pur d'ordre p défini par (4.1), la matrice $B(t)$ (voir la proposition de Liu) est définie par

$$B(t) = \begin{bmatrix} b_{11}X_{t-1} & b_{22}X_{t-2} & \dots & b_{pp}X_{t-p} \\ 1 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

4.2.2 Conception de l'algorithme

Le passage indispensable de notre algorithme est la représentation du modèle BL(0,0,p,p) sous forme espace d'état. Pour le modèle, qui fait l'objet de ce chapitre, et vu que la présentation d'état d'un modèle n'est pas unique, nous l'avons réécrit sous la forme espace d'état suivante :

$$\begin{cases} \xi_{t+1} &= A\xi_t + v_{t+1} & : \text{équation d'état} \\ X_t &= H_{t-1}\xi_t & : \text{équation d'obseration} \end{cases} \tag{4.2}$$

où ξ_t est un vecteur colonne d'ordre $(p \times 1)$

$$\xi_t = [e_t, e_{t-1}, \ldots, e_{t-p}]',$$

v_t est un vecteur colonne d'ordre $(p \times 1)$

$$v_t = [e_t, 0, \ldots, 0]',$$

H_t est un vecteur ligne d'ordre $(1 \times p)$

$$H_{t-1} = [1, b_{11}X_{t-1}, b_{22}X_{t-2}, \ldots, b_{pp}X_{t-p}],$$

et A est une matrice d'ordre $(p \times p)$

$$A = \begin{pmatrix} 0 & 0 & 0 & \ldots & \ldots & 0 \\ 1 & 0 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & 1 & 0 \end{pmatrix}.$$

La log-vraisemblance de l'espace d'état (4.2), s'obtient à partir de la densité des observations $x_1, \ldots, x_n$, son expression est identique à celle donnée au chapitre deux, nous la représentons ci-dessous :

$$L(x;\theta) = -\frac{n}{2}\ln(2\pi) - \sum_{t=1}^{n}\ln(\hat{M}_{t|t-1}) - \sum_{t=1}^{n}\frac{(x_t - \hat{x}_{t|t-1})^2}{\hat{M}_{t|t-1}} \tag{4.3}$$

Cette fonction, pour un vecteur de paramètres θ fixé, sera calulée par le filtre de Kalman. Dans la suite nous considérons le problème de minimisation de $l(x;\theta) = -L(x;\theta)$, qui est homologue du problème de maximisation de $L(x;\theta)$.

Dans la littérature, il existe une variété de méthodes, déterministes ou stochastiques, pour résoudre un problème de minimisation donné. Pour ce cas d'étude, nous avons opté pour la méthode stochastique de recuit simulé, qui est une méthode d'optimisation globale. Lors de notre expérience de simulation nous avons remarqué que, parfois, l'algorithme se stagne dans un minimum local, pour remédier à ce problème nous avons utilisé la méthode SPSA, qui est une méthode d'optimisation stochastique introduite par Spall (1992). Cette méthode est décrite brièvement en annexe.

Présentement, nous allons traduire les conditions de stationnarité et d'inversibilité en un sous algorithme Test(θ), ensuite nous décrirons succinctement notre algorithme d'estimation des paramètres du modèle bilinéaire BL(0,0,p,p), que nous avons appelé MLKF2, où nous avons implémenté le sous-algorithme KF(θ) qui décrit, le calcul de la log-vraisemblance par le filtre de Kalman et le sous-algorithme qui teste la vérification des propriétés probabilistes relatives à la stationnarité et l'inversibilité.

Algorithme Test(θ)

Etape 1 : Recherche des valeurs propres de Γ.

Etape 2 : **Si** $\varrho(\Gamma) < 1$ et $E(Log \parallel \prod_{j=1}^{p} B(t-j) \parallel) < 0$,

Etape 3 : **Alors** allez à l'étape 3 de l'algorithme MLKF2

Etape 4 : **Sinon** allez à l'étape 1 de l'algorithme MLKF2

FinAlgorithme

Algorithme MLKF2

Etape 0 : (Initialisation)

Initialisation du recuit simulé.

Initialisation du filtre de Kalman.

Etape 1 : Partez du point θ_i et générez le point aléatoire θ dans la direction d_h :

$\theta = \theta_i + r\nu_{m_h} d_h$

où r est un nombre généré dans l'intervalle $[-1, 1]$ par un générateur de nombre pseudo-aléatoire; d_h est le vecteur dont la $h^{\text{ème}}$ position égale à 1 et 0 ailleurs; et v_{m_h} est la composante du vecteur v_m, dans la même direction.

Etape 2 : Appelez le sous algorithme Test(θ)

Etape 3 : Appelez le sous-algorithme $KF(\theta)$ et Calculez $f_\theta = f(\theta)$.

Si $f_\theta < f_i$ **Alors** acceptez le nouveau point :

Mettez $\theta_{i+1} = \theta_i$,

Mettez $f_{i+1} = f_\theta$,

$i = i + 1$,

$n_h = n_h + 1$;

Si $f_\theta < f_{opt}$ **Alors** mettez

$\theta_{opt} = \theta$,

$f_{opt} = f_\theta$.

Sinon acceptez ou rejettez le point avec une probabilité d'acceptation égale à p où

$p = exp\left(\frac{f_i - f_\theta}{T_k}\right)$.

Générez un nombre p' pseudo-aléatoire dans l'intervalle $[0,1]$.

Si $p' > p$ le point est rejeté.

Sinon

Mettez $\theta_{i+1} = \theta_i$,

Mettez $f_{i+1} = f_\theta$,

Mettez $i = i + 1$,

Mettez $n_h = n_h + 1$.

Etape 4 : $h = h + 1$.

Si $h \leq n$ **Alors** allez à l'Etape 1.

Sinon $h = 1$ et $j = 1$.

Etape 5 : **Si** $j < N_s$ **Alors** allez à l'Etape 1.

Sinon mettez à jour le vecteur v, pour chaque direction u

la nouvelle composante de v'_u est

$v'_u = \nu_{m_u}\left(1 + c_u \frac{n_u/N_s - 0.6}{0.4}\right)$ Si $n_u > 0.6N_s$

$v'_u = \frac{\nu_{m_u}}{1 + c_u \frac{0.4 - n_u/N_s}{0.4}}$ Si $n_u < 0.4N_s$

$v'_u = \nu_{m_u}$

Mettez $v_{m+1} = v'$,

Mettez $j = 0$,

Mettez $n_u = 0$ pour $u = 1, \ldots, n$,

Mettez $m = m + 1$.

Etape 6 : **Si** $m < N_T$ **Alors** allez à l'Etape 1.

Sinon

Mettez $T_{k+1} = r_T T_k$,

Mettez $f_k^* = f_i$,

Mettez $k = k + 1$,

Mettez $m = 0$.

Etape 7 : (Critères d'arrêt)

Si $\mid f_k^* - f_{k-u}^* \leq \epsilon \mid, u = l, \ldots, N_\epsilon$

et $f_k^* - f_{opt} < \epsilon$,

Alors arrêtez la recherche de l'optimum.

Sinon

$i = i + 1$,

Mettez $\theta_i = \theta_{opt}$,

Mettez $f_i = f_{opt}$.

Allez à l'Etape 1.

FinAlgorithme

4.2.3 Simulations

L'objet de cette partie, est l'étude par simulation de l'efficacité de l'algorithme d'estimation MLKF2 à travers les modèles bilinéaires diagonaux purs d'ordre un et deux. En premier lieu, nous présentons les résultats numériques relatifs au modèle bilinéaire diagonal d'ordre un, ensuite nous discutons ceux concernant le modèle diagonal d'ordre deux.

Modèle BL(0,0,1,1)

Le modèle bilinéaire diagonal d'ordre un est défini par l'équation suivante :

$$X_t = b_{11} X_{t-1} e_{t-1} + e_t. \tag{4.4}$$

Granger et Anderson (1978 b) ont établi une condition suffisante d'inversibilité qui s'exprime par

$$2b_{11}^2 \sigma^2 < 1,$$

la condition d'existence et de stationnarité se déduit facilement de celle du modèle BL(1,0,1,1).

Ces deux conditions constitueront le sous-algorithme test implémenté dans l'algorithme MLKF2, dans le cas du modèle (4.5). Au cours de notre expérience, nous avons remarqué que la méthode du recuit simulé, parfois, stagne dans un minimum local; et pour remédier à ce problème, nous avons utilisé la méthode SPSA (décrite en annexe). L'expérience réalisée, consiste en la simulation de séries de taille 50, 100, 200 et 500, à partir des valeurs suivantes $0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7$ du paramètre b_{11}, en considérant le processus $(e_1, \ldots, e_n)$ gaussien de variance égale à 1 fixée.

En effet, pour $n = 50$ nous avons pris $b_{11} = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7$, pour $n = 100, 200$ nous avons pris $b_{11} = 0.1, 0.2, 0.3, 0.4$ et pour $n = 500$ nous avons pris $b_{11} = 0.1, 0.2, 0.3$. l'expérience a été répétée 1000 fois.

Pour justifier la bonne performance de notre algorithme relativement au modèle BL(0,0,1,1), nous avons confronté ces résultats numériques obtenus avec celles de la méthode de Kim et Billard (1990).

Les tables (4.1,4.3,4.5,4.7), résument les résultats numérique associés à notre algorithme, et les tables (4.2,4.4,4.6,4.8), présentent les résultats numériques de la méthode de Kim et Billard.

b_{11}	Moyenne	Biais	MSE	T-statistic
0.1	0.1021532	-0.00231532	3.9433e-004	-0.1084
0.2	0.199577	0.000423	7.2219e-006	0.1574
0.3	0.292	0.008	1.0767e-004	0.7723
0.4	0.3964204	0.035796	2.6852e-004	0.2184
0.5	0.4974275	0.0025725	0.0032	0.0455
0.6	0.5953456	0.0045644	0.0102	0.0461
0.7	0.6997	2.670310^{-4}	0.0225	0.0018

Table 4.1: Moyenne, Biais, MSE et T-statistic des paramètres estimés par l'algorithme MLKF2 pour n=50

b_{11}	Moyenne	Biais
0.1	0.000034	0.099966
0.2	0.019782	0.180218
0.3	0.028374	0.271626
0.4	0.037541	0.362459
0.5	0.056334	0.443666
0.6	0.071366	0.528634
0.7	0.091583	0.608417

Table 4.2: Moyenne, Biais des paramètres estimés par la méthode de Kim et Billard pour n=50

b_{11}	Moyenne	Biais	RMSE	T-statistic
0.1	0.106	-0.006	2.5776e-004	-0.3884
0.2	0.2017	-0.0017	7.8762e-004	-0.0589
0.3	0.297	0.003	0.0015	0.0767
0.4	0.402	-0.0024	0.0028	-0.0384

Table 4.3: Moyenne, Biais, MSE et T-statistic des paramètres estimés par l'algorithme MLKF2 pour n=100

b_{11}	Moyenne	Biais
0.1	0.007339	0.092661
0.2	0.016948	0.183052
0.3	0.022533	0.277467
0.4	0.032609	0.367391

Table 4.4: Moyenne, Biais des paramètres estimés par la méthode de Kim et Billard pour n=100

b_{11}	Moyenne	Biais	RMSE	T-statistic
0.1	0.1063	-0.0063	2.6287e-004	-0.3894
0.2	0.2036	-0.0036	0.0097	-0.0365
0.3	0.2905	0.0095	0.0014	0.2559
0.4	0.3973523	0.0026477	0.0013	0.0736

Table 4.5: Moyenne, Biais, MSE et T-statistic des paramètres estimés par l'algorithme MLKF2 pour n=200

b_{11}	Moyenne	Biais
0.1	0.007397	0.92603
0.2	0.012548	0.187452
0.3	0.016893	0.283107
0.4	0.023595	0.36405

Table 4.6: Moyenne, Biais des paramètres estimés par la méthode de Kim et Billard pour n=200

b_{11}	Moyenne	Biais	MSE	T-statistic
0.1	0.1062	-0.0062	9.9670e-004	-0.1962
0.2	0.199318	6.82e-004	7.6654e-004	0.0246
0.3	0.29057	0.0094	0.0010	0.2914

Table 4.7: Moyenne, Biais, MSE et T-statistic des paramètres estimés par l'algorithme MLKF2 pour n=500

b_{11}	Moyenne	Biais
0.1	0.005639	0.094361
0.2	0.008454	0.191546
0.3	0.011305	0.288695

Table 4.8: Moyenne, Biais des paramètres estimés par la méthode de Kim et Billard pour n=500

Les différents résultats relatés dans les tables, ci-dessus, montrent la supériorité de notre algoithme par rapport à la méthode de Kim et Billard.
Les tables (4.1), (4.3), (4.5) et (4.7), illustrent aussi la bonne performance de notre algorithme, pour le modèle qui fait l'objet de cette étude, puisque les valeurs de la T-statistic révèlent la non significativité du biais.

Modèle BL(0,0,2,2)

Le modèle bilinéaire diagonal d'ordre deux est défini par l'équation suivante :

$$X_t = b_{11}X_{t-1}e_{t-1} + b_{22}X_{t-2}e_{t-2} + e_t. \tag{4.5}$$

Nous avons réalisé 1000 simulations de taille $N = 500$ du modèle bilinéaire suivant :

$$X_t = 0.05X_{t-1}e_{t-1} + 0.1X_{t-2}e_{t-2} + e_t \quad e_t \sim iidN(0,1)$$

La Table (4.9), pour ce modèle simulé, présente le comportement empirique des estimateurs $\tilde{b}_{11}$ et $\tilde{b}_{22}$ fournis par notre algorithme MLKF2 au moyen, toujours, des statistiques : "Moyenne", "MSE" et "T-statistic". Les résultats empiriques dressés dans cette table permet de conclure que pour le cas choisi la moyenne des estimations calculée sur les 1000 simulations s'approche de la vraie valeur des paramètres b_{11} et b_{22}, ce qui a suscité des biais très petits, et le carré moyen des estimations s'approche de zéro.
Ces résultats laissent entrevoir des résultats encouragents de notre algorithme MLKF2, pour les autres modèles bilinéaires diagonaux pur d'ordre p.

Paramètres	Vraie valeur	Moyenne	Biais	MSE	T-statistic
b_{11}	0.05	0.0499801	0.0000199	0.0000083	0.0068790
b_{22}	0.1	0.0999689	0.0000311	0.0003710	0.0016132

Table 4.9: Moyenne, Biais, MSE et T-statistic des paramètres estimés pour n=500

Dans le paragraphe suivant, nous avons implémenté cet algorithme dans une application vouée aux modèles bilinéaires particuliers qui ont fait l'objet de cette thèse. Nous avons utilisé dans le développement de cette application, en ce qui concerne la partie optimisation de la fonction de log-vraisemblance, la méthode de recuit simulé. Et pour le calcul de cette log-vraisemblance nous utilisons le filtre de Kalman.

4.3 Application pour l'estimation des paramètres des modèles bilinéaires : LogBL

L'objectif de ce paragraphe est la présentation d'un outil de calculs, que nous avons conçu, et qui concerne les modèles bilinéaires faisant l'objet de ce mémoire. L'outil se présente sous forme d'une application développée avec le langage Java. Actuellement, la plupart des logiciels existants abordent les modèles linéaires et les modèles non-linéaires ARCH et GARCH de séries chronologiques. Notre application présente l'avantage de permettre de simuler aisément certains modèles bilinéaires et d'estimer leurs paramètres.

Sa conception s'appuie sur certaines méthodologies développées dans le premier chapitre. L'application en question présente une interface conviviale qui fonctionne dans l'environnement Windows, et qui pourra aussi être implémentée dans l'environnement Unix et Linux.

Nous consacrons la première partie de ce paragraphe à la définition de quelques concepts inhérents au langage java, qui est un pur langage orientée objet.

L'objectif de cette section ne sera pas de fournir au lecteur un manuel de programmation avec toutes les informations de syntaxe du langage java, mais de lui présenter quelques définitions.

La deuxième partie, présente une description de notre application codé en java.

4.3.1 Quelques concepts du langage Java

Java est un langage de programmation développé par sun Microsystems. Il n'a que quelques années de vie, et pourtant il a réussi à intéresser et intriguer beaucoup de développeurs à travers le monde. Sa réputation est dû aux nouveaux aspects et avantages qu'il intègre, qu'on peut résumer dans :

- C'est un langage orienté objet dérivé du C, mais plus simple que le C++.
- Il est multi-plateforme : tous les programmes tourneront sans modification sur toutes les plateformes où existe Java.
- Il est doté d'une riche bibliothèque de classes, comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, menus...etc), la programmation muli-threads (multitâches), la gestion des exceptions, les accès aux fichiers et au réseau...etc

Dans cette section, nous allons présenter les concepts les plus importants de la programmation avec java.

Java, un langage de programmation orienté objet (POO)

La programmation orientée objet consiste à modéliser informatiquement un ensemble d'éléments d'une partie du monde réel (que l'on appelle domaine) en un ensemble d'entités informatiques. Ces entités informatiques sont appelées objet.

Objets

Les objets ont été conçus de façon à reproduire les mêmes propriétés que nous trouvons dans un objet du monde réel. Ces propriétés peuvent être résumées en deux caractéristiques principales que les objets du monde réel présentent : ils sont dans un certain état et ils possèdent un certain comportement.
L'état d'un objet est représenté par un ensemble de données associées qu'on appelle

variables ou attributs de l'objet. En même temps, son comportement est caractérisé par l'ensemble d'actions qu'il est capable de réaliser. Ces actions sont exécutées par des méthodes qu'on associe à l'objet et qui lui permettent de réagir aux sollicitations extérieures en utilisant les valeurs de ses attributs.

Exemple :

Pour une gestion ferroviaire, le programmeur par objets sera tenté de définir les objets suivants : train, voitures, lignes, gares,... Pour chaque type d'objet, il définira les propriétés de ces objets et les actions que l'on peut leur faire subir : un train est formé d'une collection de voitures, on peut lui accrocher une voiture, lui décrocher une voiture, chaque voiture a une capacité...

Classes

Une classe est une structure de données qui contient les attributs et les méthodes communes à tous les objets d'une même nature. Elle représente un "moule" utilisé pour créer des objets qui ont des caractéristiques en commun mais qui peuvent être dans des états différents. On dit qu'un objet est une instance d'une classe.

Attributs

Un attribut est une valeur de données détenue par les objets de la classe. Chaque attribut a une valeur pour chaque instance d'objet.

Méthodes

Une méthode est une fonction ou une transformation qui peut être appliquée aux objets ou par les objets dans une classe. Tous les objets d'une même classe partagent les mêmes méthodes.

Le langage Java est portable et indépendant des plates-formes

Le code intermédiaire produit est indépendant des plates-formes : il pourra être exécuté sur tous types de machines et systèmes pour peu qu'ils possèdent l'interpréteur de code Java.

Le langage Java est dynamique et multithread

Le langage Java est dynamique et s'adapte à l'évolution du système sur lequel il s'exécute. Les classes sont chargées en fur et à mesure des besoins, à travers le réseau s'il le faut. Les mises à jour des applications peuvent se faire classe par classe sans avoir à recompiler le tout en un exécutable final. De nos jours, les applications possèdent un haut degré de parallélisme : il faut pouvoir écouter une musique, tout en regardant une animation graphique etc. Java permet le mutlithreading de manière simple.
Dans la section suivante, nous allons décrire notre application développée en Java.

4.3.2 Description de l'application

Java est un langage de programmation orienté objet à usage général, à partir duquel il est possible de développer toute sorte d'applications : IHM (interface homme-machine), accès aux bases de données, applications Web, applets (exécutées depuis un navigateur Internet, etc...).
Notre application, que nous avons appelé LogBL (Logiciel pour les Modèles Bilinéaires), est une IHM constituée de trois interfaces chaque interface traduit les classes et les méthodes que nous utilisons en arrière plan.
La première interface (4.1) est l'interface accueil du logiciel, elle contient deux boutons qui servent à lancer soit l'interface simulation où l'interface estimation.
La deuxième interface (4.2) nommée **Simulation des modèles bilinéaires**, inclut les fonctionnalités qui permettent de simuler un modèle bilinéaire.

Figure 4.1: Interface accueil

Elle comporte un "ButtonGroup" qui permet à l'utilisateur de séléctionner le modèle bilinéaire à simuler. Nous nous sommes intéressés, dans cette première version de notre application, aux modèles BL(0,0,2,1), BL(1,0,1,1) et le modèle BL(0,0,1,1). L'interface comporte aussi une liste déroulante "JComboBox", qui propose aux utilisateurs différents choix de la taille de la série à simuler, cette taille varie entre 50 et 1000.

Les zone de texte "jTextField" (voir (4.3)) permet de saisir les valeurs des coefficients du modèle bilinéaire.

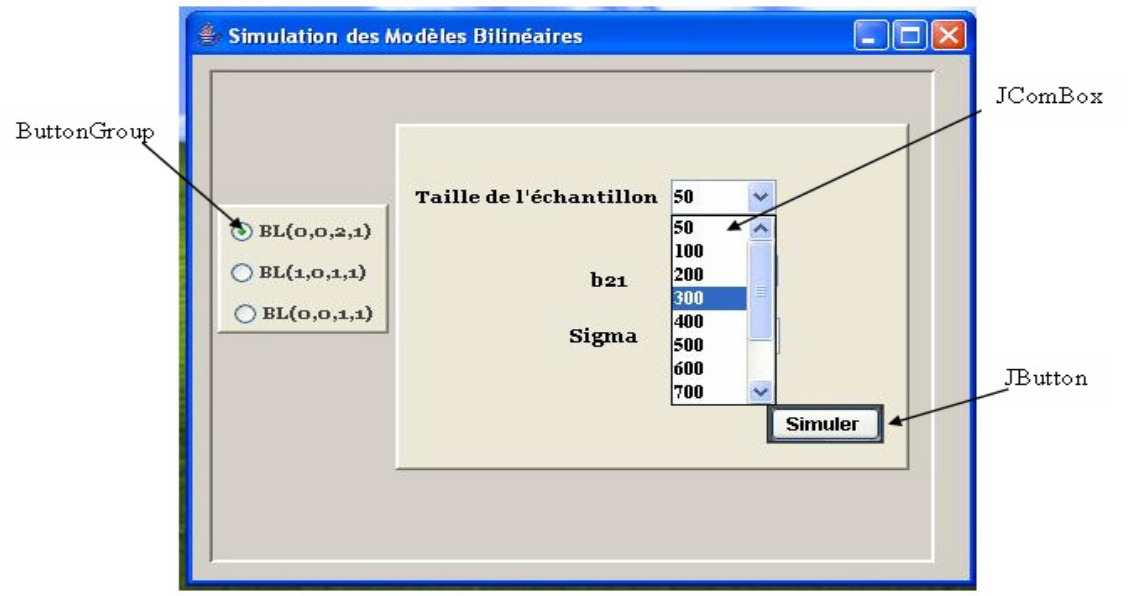

Figure 4.2: Interface graphique du projet simulation

Une fois les choix sont remplis, l'utilisateur peut commencer à simuler la série du modèle choisi en cliquant sur le bouton "**Simuler**", suite à cet événement, le logiciel sauvegarde cette série dans un fichier.

L'interface nommée **Estimation des paramètres des modèles bilinéaires**, est destinée à estimer les paramètres d'un des modèles bilinéaires précités. La classe "Estimation" qui a suscité cet interface, est composée d'objets qui ne sont autres que les modèles bilinéaires, et chaque objet contient des attributs : ensemble des paramètres à estimer, et possède des méthodes qui permettent d'accéder à l'objet et d'autres à l'estimer.

Les méthodes relatives à chaque objet sont **FiltreKalman()** et **RecuitSimule()**. La méthode **FiltreKalman()** s'occupe du calcul de la log-vraisemblance par le filtre de Kalman pour chaque valeur du vecteur des paramètres. Et la méthode **RecuitSimule()**, permet d'évaluer le minimum de la log-vraisemblance.

Les différents aspects de l'interface graphique qui présente cette classe sont donnés

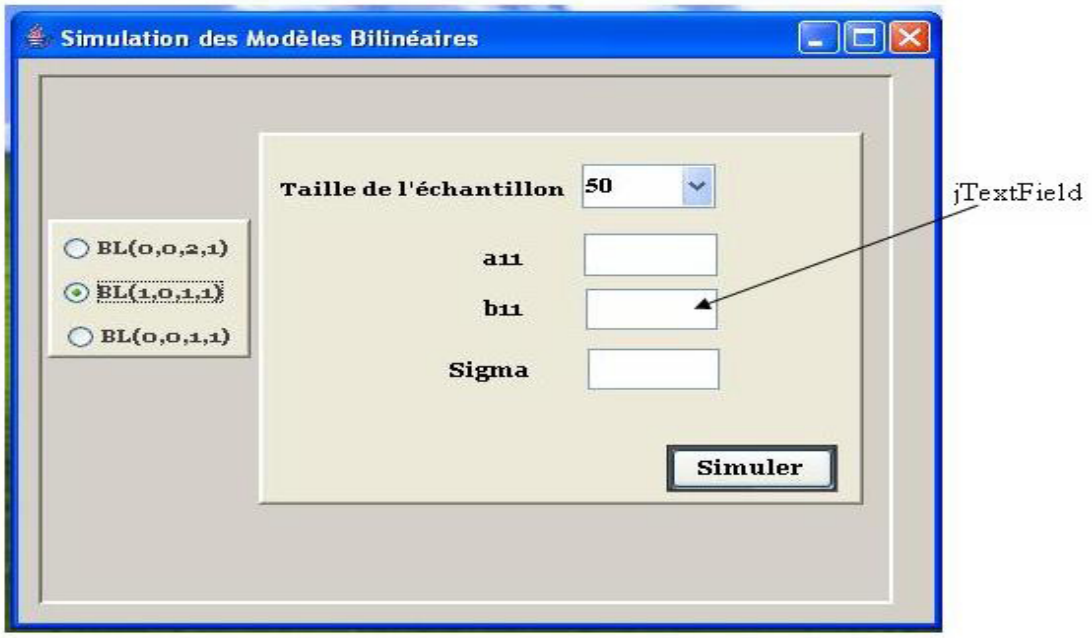

Figure 4.3: Interface graphique du projet simulation

dans les figures (4.4,4.5,4.6).

Figure 4.4: Interface graphique du projet Estimation

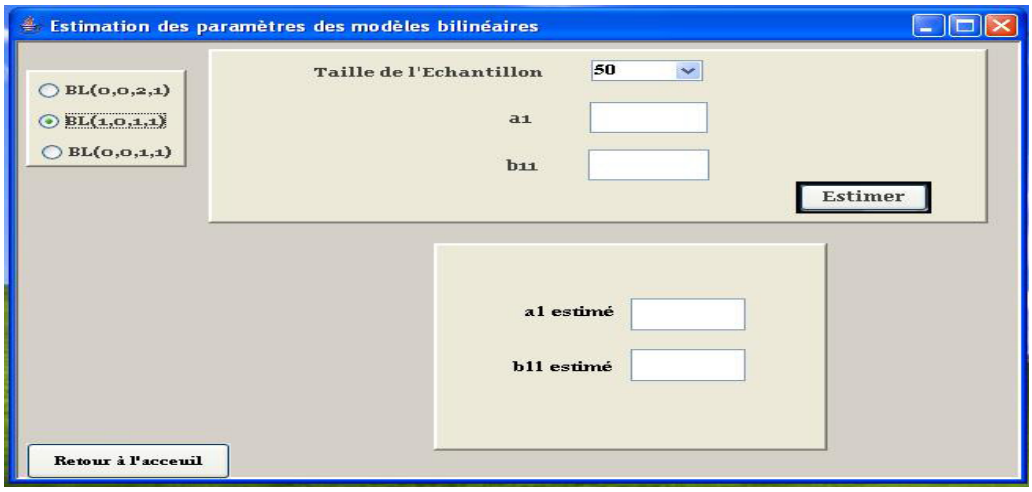

Figure 4.5: Interface graphique du projet Estimation

Figure 4.6: Interface graphique du projet Estimation

Sur cet interface, nous observons deux zones relatives, respectivement, à l'entrée des vraies valeurs des paramètres du modèle bilinéaire sélectionné et l'affichage des valeurs estimées de ces paramètres, après l'exécution de la méthode *RecuitSimule* en cliquant sur le boutton "Estimer".

Dans cette section, nous avons présenté la première version du logiciel LogBL, nous continuons à développer et à améliorer ce logiciel pour subvenir aux besoins de l'utilisateur.

4.4 Conclusion

Dans cette section, nous avons introduit une autre façon d'écrire la représentation d'état des modèles bilinéaires et nous avons utilisé la méthode recuit simulé qui est une méthode d'optimisation globale pour minimiser notre fonction de log-vraisemblance. Cette méthode a été combinée avec la méthode SPSA pour éviter que l'algorithme se stagne dans un minimum local. Les résultats numériques obtenus montrent l'efficacité et la bonne performance de l'algorithme MLKF2.

À travers cette contribution, nous avons essayé aussi de développer un logiciel avec des interfaces conviviales et maniables, qui traitent la partie simulation et estimation des paramètres des cas particuliers des modèles bilinéaires.
Cette application représente la première version de notre logiciel, dans la perspective de le développer pour qu'il englobe tous les modèles bilinéaires.

Conclusion générale et perspectives

Cette thèse aborde les thèmes de l'estimation, la simulation, la modélisation et la prédiction. Nous venons d'élaborer des algorithmes d'estimation de certains modèles bilinéaires particuliers : modèle bilinéaire superdiagonal d'ordre un (BL(0,0,2,1)), modèle bilinéaire diagonal d'ordre un (BL(1,0,1,1)) et le modèle bilinéaire diagonal pur d'ordre p (BL(,0,p,p)).

Ces algorithmes se basent sur la méthode du maximum de vraisemblance et l'algorithme du filtre de Kalman qui requiert un passage indispensable constitué de la présentation des modèles bilinéaires, objet de cette thèse, en espace d'état. Le rôle essentiel de l'algorithme du filtre de Kalman est de calculer la log-vraisemblance des modèles bilinéaires à chaque itération de nos algorithmes. Pour maximiser la log-vraisemblance, nos algorithmes utilisent la méthode d'optimisation déterministe : Powell, et la méthode d'optimisation stochastique : recuit simulé.

Nous avons confirmé la bonne performance et la supériorité de nos algorithmes par rapport à des méthodes existantes, à travers des simulations numériques sur les cas étudiés.

Il nous a paru utile de montrer les possibilités pratiques des modèles bilinéaires BL(0,0,2,1) et BL(1,0,1,1), en les ajustant respectivement aux données réelles émanant de la fiabilité des logiciels et du tremblement de terre qui a frappé Al-Hoceima, après avoir estimé les paramètres de ces modèles par nos algorithmes.

Les résultats numériques obtenus suite à l'étude de la qualité prédictive des deux modèles, sont bons et encourageants pour le modèles BL(0,0,2,1) et satisfaisants pour le modèle BL(1,0,1,1).

Enfin, nous avons conçu une application, appelée LogBL, qui permet de simuler les modèles bilinéaires étudiés dans cette thèse, et d'estimer leurs paramètres via notre algorithme d'estimation.

Une très bonne continuation de ce travail, sera la généralisation de ces algorithmes à des modèles bilinéaires plus complexe, la construction d'un algorithme d'identification des modèles bilinéaires et l'amélioration de notre première version du logiciel LogBL, par l'implémentation d'autres modèles bilinéaires et d'autres options comme la prédiction et l'identification.

Annexe : Algorithme de Brent et la méthode SPSA

Méthode de Brent

La méthode de Brent (1973) est une méthode unidimentionnelle qui effectue la réduction de l'intervalle de recherche en utilisant une interpolation polynomiale de la fonction objective, calculée à partir d'un triplet (x_1, x_2, x_3). Dans cette méthode, le point de découpage est donné par l'abscisse de la parabole définie par le triplet. Son algorithme est le suivant :

Soit f la fonction à minimiser, définie sur l'intervalle $[a, b]$.

Algorithme de Brent

Etape 1 : On pose $c = (3 - \sqrt{5})/2, e = 0,$

$v = w = x = a + c * (b - a)$

$fv = fw = dx = f(x)$

Etape 2 : Faire

$m = 0.5 * (a + b)$

$tol = eps* \mid x \mid +t$

$t2 = 2 * tol$

Si $(\mid x - m \mid > t2 - 0.5 * (b - a))$ **Alors**

$p = q = r = 0$

Si $(\mid e \mid > tol)$ **Alors**

$r = (x - w) * (fx - fv)$

$q = (x - v) * (fx - fw)$

$p = (x - v) * q - (x - w) * r$

$q = 2 * (q - r)$

Si $(q > 0)$ **Alors**

$p = -q$

Sinon

$q = -q, r = e, e = d$

Si $(\mid p \mid < \mid 0.5 * q * r \mid)$ **et** $(p < q * (a - x))$ **Alors**

$d = p/q, u = x + d.$

Si $((u - a) < t2)$**ou bien** $((b - u) < t2)$ **Alors**

$d =$ (Si $(x < m)$ Alors tol Sinon $-tol$).

Sinon

$e =$ (Si $(x < m)$ Alors b Sinon a) $- x$

$d = c * e.$

$u = x +$ (

Si $\mid d \mid \geq tol$ Alors d Sinon (Si $d > 0$ Alors tol Sinon $-tol$))

$fu = f(u)$

Si $(fu \leq fx)$ **Alors**

Si $(u < x)$ **Alors** $b = x$ **Sinon** $a = x$

$v = w, fv = fw, w = x, fw = fx, x = u, fx = fu;$

Sinon

Si $(u < x)$ **Alors** $a = u$ **Sinon**$b = u$;

Si $(fu \leq fw)$ **ou bien** $(w = x)$ **Alors**

$v = w, fv = fw, w = u, fw = fu$;

Sinon $(fu \leq fv)$ ou bien $(v = x)$ ou bien $(v = w)$ **Alors**

$v = u, fv = fu$;

Allez à Faire

Etape 3 : min=fx.

Méthode SPSA(Simultaneous Perturbation Stochastic Perturbation)

Récemment, la méthode SPSA a été considérée avec un intérêt majeur. Elle se base sur le calcul de la fonction objective uniquement, contrairement aux méthodes déterministes qui s'appuient sur le calcul direct du gradient.

Cette méthode a été introduite par Spall (1992). Nous rapportons ici une description succinte de cette méthode, pour plus de détaille sur la méthode et ses applications vous pouvez consulter les travaux de Spall(1992, 1994, 1997, 1998).

Etape 1 : **initialisation et sélection des coefficients: k=0**

choisir un θ_0 initial

choisir des valeurs pour les coefficients strictement positifs suivants: $\lambda, \lambda_0, c, \alpha$ et γ.

α et γ vérifient $3\gamma - \alpha/2 \geq 0, \alpha - 2\gamma > 0$

Etape 2: **génération du vecteur simultanément perturbé**

Générer un vecteur Δ_k de p variables aléatoires indépendantes symétriquement distribuées autour de zéro, à l'iteration k. Un choix simple qui satisfait a de telles conditions est d'utiliser la distribution de Bernoulli ± 1 avec la probabilité de $1/2$ pour chaque réalisation de ± 1.

Etape 3: **évaluation de la fonction objective**

Obtenir deux mesures de la fonction $f(\theta)$ basées sur la perturbation simultané autour de $y(\theta_k + c_k\Delta_k)$ et $y(\theta_k - c_k\Delta_k)$

avec c_k, Δ_k et tel qu'elles sont décrites dans l'étape 2 et 4 respectivement.

Etape 4: **approximation du gradient**

Génère l'approximation des perturbations simultanées au gradient inconnu $g(\theta_k) = \frac{\partial f}{\partial \theta_k}$ tel que:

$$g_k(\theta_k) = \begin{bmatrix} \frac{y(\theta_k + c_k\Delta_k) - y(\theta_k - c_k\Delta_k)}{2c_k\Delta_{kl}} \\ \vdots \\ \frac{y(\theta_k + c_k\Delta_k) - y(\theta_k - c_k\Delta_k)}{2c_k\Delta_{kp}} \end{bmatrix}$$

avec $c_k = \frac{c}{k^\gamma}$

Etape 5: **mise à jour de l'estimateur de l'optimum θ**

Utiliser la forme suivante: $\theta_{k+1} = \theta_k - \lambda_k(g(\theta_k))$

avec $\lambda_k = \frac{\lambda}{(\lambda_0 + k)^\alpha}$

Etape 6: **itérer ou terminer**

Si $\| \theta_{k+1} - \theta_k \| < \epsilon$

Retourner à l'étape 2 avec $k+1$ qui remplace k

ou encore jusqu'au maximum de nombres d'itérations permis.

Les suites $(\lambda_k)_{k\geq 0}$ et $(c_k)_{k\geq 0}$ sont telles que : $\lambda_k \to 0$ et $c_k \to 0$ lorsque $k \to \infty$, $\sum_{k=0}^{\infty} \lambda_k = \infty$ et $\sum_{k=0}^{\infty} \left(\frac{\lambda_k}{c_k}\right)^2 < \infty$.
Un possible choix de α et γ est : $\alpha = 0.602$, $\gamma = 0.101$.

Le paramètre λ est égal à 5-10% du nombre d'itérations.

Bibliographie

H. Akaik, *Fitting autoregressive models for prediction*, Annals of the institute of Statistica Mathematics, 21, (1969), pp. 243–247.

S. I. Akamanan , R. M. Bhaskara et S. Subramanyan, *On the ergodicity of bilinear time series models*, J. T. S. A. 7, (1986), pp. 157–163.

A. Alan et B. Pritseker, *Introduction to Simulation*, Systems Publishing Corporation, West lafayette, Indiana, (1984).

B. D. O. Anderson et J. B. Moore, *Optimal Filtering*, Prentice-Hall, (1979).

C. F. Ansley et R. Kohn, *Exact likelihood of vector autoregressive moving average process with missing or aggregated data*, Biometrika, (1983), pp. 275–278.

M. S. Arulampalam, S. Maskell, N. Gordon et T. Clapp, *A tutorial on particle filters for online nonlinear-non-gaussian Bayesian tracking*, IEEE Transactions on Signal Processing, 50(2), february (2002), pp. 174-188.

J. A. Bather, *Invariant conditional distributions*, Ann. Math, 36, (1965), pp. 829–846.

Y. Benghabrit et M. Hallin, *Optimal rank based tests against first order superdiagonal bilinear dependence.* J. statist. Plann. Inference, 32, (1992), pp. 45–61.

Y. Benghabrit et M. Hallin, *Locally asymptotically optimal tests for autoregressive against bilinear serial dependence.* Statistica Sinica 6, Numéro 1, (1996), pp. 147–169.

Y. Benghabrit et M. Hallin, *Locally asymptotically optimal tests for AR(p) against diagonal bilinear dependence*, J. statist. Plann.Inference 68, (1998), pp. 47–

63.

R. M. Bhaskara, R. T. Subba et A. M. Walker, *On the existence of some bilinear time series models*, J. T. S. A. 4, (1983), pp. 95–110.

C. Bourin et P. Bondon, *On the identifiability of bilinear stochastic systems*, IEEE, (1997).

K. Bouzaachane et Y. Benghabrit, *Modélisation des temps d'inter défaillance par les modèles ARMA*, CIMASI, 23-25 Octobre (2002).

K. Bouzaachane, M. Harti et Y. Benghabrit, *Modelling Earthquake of Al Hocima by First-order Bilinear time series*, CIRO'05, Mai (2005).

K. Bouzaachane, M. Harti et Y. Benghabrit, *First-order superdiagonal bilinear time series for tracking software reliability*, Interstat Journal, Feburary (2006 a).

K. Bouzaachane, M. Harti et Y. Benghabrit, *Parameter estimation for first-order superdiagonal bilinear time series : An algorithm for maximum likelihood procedure*, Interstat Journal, Juillet (2006 b).

K. Bouzaachane, M. Harti et Y. Benghabrit, *Algorithme d'estimation des paramètres d'un modèle bilinéaire*, ARIMA Journal, soumis (2006 c).

M. J. Box, D. Davies et W. H. Swann, *Non-linear optimization techniques* ICI Monograph N 5, Oliver and Boyed, London(5.4, 5.5, 7.1), (1969).

G. E. Box et G. M. Jenkins, *Time series analysis, forecasting and control*, San Francisco, CA, Holden-Day, (revised edition), (1970).

R. B. Brent, *Algorithms for Minimization without derivatives*, Prentice-Hall, Series in automatic computation, (1973).

Y. Chen et N. D. Singpurwalla, *A non-Gaussian Kalman filter model for tracking software reliability*, Statistica Sinica 4, (1994), pp. 535–548.

Y. Cherruault, *Optimisation : Méthodes Locales et Globales*, Presses Universitaires de France, ISBN 2-130-49910-4, (1999).

A. Corana, M. Marchesi, C. Martini et S. Ridella, *Minimizing Multimodal*

functions of continuous variables with simulated annealing Algorithm, ACM Transactions on Mathematical software, 35, (1987), pp. 262–280.

L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, (1986)

J. J. Droesbeke, B. Fichet et P. Tassi, *Series chronologiques, Théorie et pratique ARIMA*, ASU, Economica, (1989).

P. D. Flangan, P. A. Vatale et J. A. Mendelsohn, *A numerical investigation of several one-dimensional search procedures in non linear regression problems*, Technometrics 11, (5.4), (1969), pp. 256–284.

V. Giard, *Statistique appliquée à la gestion*, Economica, (1995).

D. E. Goldberg, *Genetic Algorithms: in search, optimization and machine learning*, Addison Wesley Longman, (1989).

C. Gouriéroux et A. Monfort, *Séries temporelles et modèles dynamiques*, Economica, (1990).

G. W. J Granger et A. P. Anderson, *An introduction to bilinear time series analysis*, Vandehoeck and Ruprecht, Götting, (1978 a).

G. W. J Granger et A. P. Anderson, *On the invertibility of time series models*, Stoch, Proc, and their appl 8, (1978 b), pp. 27–92.

D. Guegan, *Etude d'un modèle non linéare, le modèle superdiagonal d'ordre un*, C. R. Acad. sci. Paris ser, 1 293, (1981), pp. 95–98.

D. Guegan, *Une condition d'ergodicité pour les modèles bilinéaires à temps discret*, C.R.A.S, t. 297, Serie I, (1983), pp. 537–540.

D. Guegan, *Tests de modèles non linéaires.* Proceeding of the third franco belgium meeting of statisticians, J.-P. Florens et al. Eds. FUSL, Brussels, (1984), pp. 45–66.

D. Guegan, *Représentation l-markovienne et existence d'une représentation affine en l'état des modèles bilinéaires* C.R.A.S. t. Serie I. N. 47, (1986), pp. 289–292.

D. Guegan, *Different representation for bilinear Models.* C.N.R.S. Orsay, France,

(1987), pp. 389-408.

D. Guegan, *Modèles bilinéaires et pôlynomiaux de séries chronologiques: Etude probabiliste et Analyse statistique*, Thèse d'état, Université Joseph Fourrier, Grenoble, (1988).

D. Guegan, *Séries chronologiques non linéaires à temps discret*, Collection statistique mathématique et probabilité, (1993).

D. Guegan et T. D Pham, *Minimalité et inversibilité des modèles bilinéaires à temps discret*, C.R.A.S, 448, (1987 a), pp. 159–162.

D. Guegan et T. D Pham, *A note on the estimation of the the parameters of the diagonal bilinear models by least squares method*, Scand. Journ. of Stat. Theory and Appl, (1987 b).

D. Guegan et T. D Pham,*A note on the estimation of the parameters of the diagonal bilinear model by the method of the least squares*, Scand. J. Statist, 16, (1989), pp. 159–136.

D. Guegan et T. D Pham, *Power of the score test against bilinear time series models*, Statistica Sinica 2 numéro 1, (1992), pp. 157–169.

J. D. Hamilton, *Time series analysis*, Princeton: Princeton University Press, (1994).

A. Hamlili, *Etude Comparative des modèles de croissance de fiabilité des logiciels*, Actes de la deuxième conférence internationale en recherche opérationnelle, Marrackech, Maroc, 24-26 Mai (1999).

M. Harti, *Algorithmes pour l'estimation par pseudo-maximum de vraisemblance exacte pour des modèles VARMA sous forme classique et sous forme structurée*, Thèse d'état, (1996).

S. Haykin, *Adaptative Filter Theory*, Prentice-Hall, (1991).

D. M. Himmerblau, *Applied non linear programming*, Mc Graw-Hill book company, (1972).

S. Houfaidi et Y. Benghabrit, *Etude asymptotique d'un M-estimateur des*

paramètres d'un modèle bilinéaire diagonal d'ordre un, XXVIIème journées de statistiques, Jouy-en-Josas, (1995 a), pp. 378–380.

S. Houfaidi et Y. Benghabrit, *Robustesse et comportement asymptotique d'un GM-estimateur des paramètres d'un modèle bilinéaire diagonal d'ordre un*, Nonlinear time series models, XVI$^{\text{th}}$ Franco-Belgian meeting of statisticians, Bruxelles, novembre (1995 b).

S. Houfaidi et Y. Benghabrit, *Inférence asymptotique paramètrique et non paramètrique pour étudier les problèmes d'estimation et des tests pour les modèles bilinéaires*, Deuxième conférence internationale de mathématiques appliquées et sciences de l'ingénieurs, EST, Casablanca, octobre (1998 a).

S. Houfaidi et Y. Benghabrit, *Tests paramétriques asymptotiquement localement "most stringent" pour une hypothèse nulle de modèle AR(1) contre un BL(1,0,2,1)*, deuxième Conférence Internationale de Mathématiques Appliquées et Sciences de l'Ingénieurs, EST, Casablanca, Octobre (1998 b).

R. E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*Transaction of the ASME, Journal of Basic Engineering, 82, march (1960).

R. E. Kalman, *New methods in Wiener filtering theory*, In John L. Bogdanoff and Frank Kozin, eds. Proceeding of the first symposium of engineering applications of random function Function theory and probability, New York:Wiley, (1963), pp. 270–388.

W. K. Kim et L. Billard, *Asymptotic properties for the first-order bilinear time series model*, Comm. Statist.theory.math, 19(4), (1990), pp. 1171-1183.

J. S. Kowalik et M. R. Osborne, *Methods for unconstrained optimization problems*, Elseiver, New York, (1968).

J. Lifermann, *Les principes du traitement statistique du signal*, Masson, (1981).

J. Liu, *A note on causality and invertibility of a general bilinear time series models*, Adv. Appl. Prob, 22, (1990 b), pp. 247-250.

J. Liu et P. J. Brockwell, *On the general bilinear time series models*, J. Appl.

Prob, 25, (1988), pp. 553–564.

V. J. Mathews et T. K. Moon, *Parameter estimation for a bilinear time series model*, IEEE, E7, 22, (1991).

G. Mélard, *A fast algorithm for the exact likelihood of autoregressive-moving average models*, Applied statistics, (1983), pp. 104-114.

G. Mélard, *Méthodes de prévision à court terme*, Bruxelles: Editions de l'Université Libre de Bruxelles, et Paris: Editions Ellipses, (1990).

G. Mélard et J. M. Pasteels, *Manuel d'utilisateur de Time series expert, TSE version 2.2*, (1994).

M. Minoux, *Programmation Mathématique: théorie et algorithme, Tome 1*, C.N.E.T et E.N.S.T, Paris, (1983).

R. R. Molher, *Natural bilinear control processes*, I.E.E.E. Trans. Syst. Sci. Cybern, vol. S.C.C.G, (1970), pp. 192–197.

J. D. Musa, *Software reliability data. Report available from data and analysis center of software*, Rome air development center, New York, (1975).

J. D. Musa, *Software reliability data*, IEEE Comput. Soc. Repository, (1979).

J. A. Nelder et R. Mead, *A Simplex Method for Function Minimization*, Computer Journal, vol. 7, (1965), pp. 308-312.

A. J. Oyet, *Nonlinear time series modeling: Order identification and Wavelet filtering*, Interstat, Septembre (2000).

D. T. Pham, *Bilinear Markovian representation and bilinear models*, Soch. Process and their appl, 20, (1985), pp. 295–306.

D. T. Pham, *The mixing property of bilinear and generalized random coefficient autoregressive models*, Stoch. Proc. and their appl, 23, (1986), pp. 291–300.

D. T. Pham, *Exact maximum likelihood estimate and lagrange multiplier test statistic for ARMA models*, Jour. Time ser.Anal, 8, (1987), pp. 61–87.

D. T. Pham et D. Guegan, *Minimalité et inversibilité des modèles bilinéaires à temps discret*, C.R.A.S. Serie I. t, 448, (1987), pp. 159–162.

D. T. Pham et L. T. Tran, *On the first order bilinear time series model*, J.App. Prob, 18, (1981), pp. 617–627.

J. G. Pearlman, *An algorithm for the exact likelihood of high-order autoregressive-moving average process*, Biometrika, 67, 1, (1980), pp. 232–233.

M. J. D. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Comp.J.7, (1964), pp. 155–162.

W. H. Press, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, ISBN 0-521-43108-5, (1992).

S. S. Rao, *Engineering Optimization: Theory and Practice*, John Wiley and Sons, ISBN 0-471-55034-5, (1996).

J. S. Rustagi, *Optimization technic in statistics*, Academic Press, London, (1994).

G. Saporta, *Probabilités, analyse des données et statistique*, Technip, Paris, (1990).

B. L. Shea, *Estimation of multivariate time series*, Journal of time series. 8. N 1, (1987), pp. 95–109.

B. L. Shea, *The exact likelihood of a vector autoregressive moving average model*, Appl. Statist 38. N 1, (1989), pp. 191-204.

N. D. Singpurwall et R. Soyer, *Assessing software reliability growth using a random coefficient autoregressive process and its ramifications*, IEEE Trans. on Soft. Eng, vol SE-11, N 12, (1985), pp. 1456–1464.

E. D. Sontag, *Realization theory of discrete-time non linear systems: Part I. The bounded case*, Vol. cas-26. N 4, April (1979), pp. 342–356.

J. C. Spall, *Multivariate stochastic approximation using a simultaneous perturbation gradient approximation*, IEEE Trans. Automat. Control, 37, (1992), pp. 332–341.

J. C. Spall, *A One-Measurement Form of Simultaneous Perturbation Stochastic Approximation*, Automatica, vol. 33, (1997), pp. 109–112.

J. C. Spall, *Implementation of the Simultaneous Perturbation Algorithm for*

Stochastic Optimization, IEEE Transactions on Aerospace and Electronic Systems, vol. 34, (1998), pp. 817–823.

J. C. Spall et J.A. Cristion , *Nonlinear Adaptive Control Using Neural Networks: Estimation Based on a Smoothed Form of Simultaneous Perturbation Gradient Approximation*, Statistica Sinica, vol. 4, (1994), pp. 1–27.

R. T. Subba, *On the theory of bilinear time series models*, J.R.S.S, Serie B, Vol 43, 2, (1981), pp. 224–255.

R. T. Subba et M. M. Gabr,*An introduction to bispectral analysis and bilinear time series models*, Lecture note in statistic, N 24, springer, Berlin, (1984).

H. Tong, *Non linear Times series: A Dynamical System Approach*, New York: Oxford univ, Press, (1990).

J. N. Wandji, *Étude de tests paramétriques et non paramétriques asymptotiquement puissants pour les modèles autorégressifs bilinéaires*, Thèse de doctorat, Université paris 13, (1995).

N. Wiener, *Non linear problems in random theory*, M.I.T. Press, (1958).

National Geographic Institut, *http://www.geo.ign.es/*