Ecole Doctorale SMPC

Thèse présentée pour l'obtention du diplôme de

DOCTEUR DE TELECOM SUDPARIS

Doctorat conjoint
Telecom Sudparis et l'Université Pierre et Marie Curie – Paris VI

Spécialité : MATHEMATIQUES APPLIQUEES

Présentée par

Noufel ABBASSI

Sujet de la thèse

CHAINES DE MARKOV TRIPLETS ET FILTRAGE OPTIMAL DANS LES SYSTEMES A SAUTS

Soutenue le 26 avril 2012, devant le jury composé de

Président	Cédric Richard	Professeur à l'Université de Nice
Rapporteur	Fabrice Heitz	Professeur à l'Université de Strasbourg
Rapporteur	Eric Moreau	Professeur à l'Université de Toulon
Examinateur	Michel Broniatowski	Professeur à l'Université Paris VI
Examinateur	Emmanuel Duflos	Professeur à l'Ecole Centrale de Lille
Examinateur	Thierry Chonavel	Professeur à Telecom Bretagne
Directeur de thèse	Wojciech Pieczynski	Professeur à Telecom Sudparis

Thèse no 2012TELE0018

Ecole Doctorale SMPC

Thèse présentée pour l'obtention du diplôme de

DOCTEUR DE TELECOM SUDPARIS

Doctorat conjoint
Telecom Sudparis et l'Université Pierre et Marie Curie – Paris VI

Spécialité : MATHEMATIQUES APPLIQUEES

Présentée par

Noufel ABBASSI

Sujet de la thèse

CHAINES DE MARKOV TRIPLETS ET FILTRAGE OPTIMAL DANS LES SYSTEMES A SAUTS

Soutenue le 26 avril 2012, devant le jury composé de

Président	Cédric Richard	Professeur à l'Université de Nice
Rapporteur	Fabrice Heitz	Professeur à l'Université de Strasbourg
Rapporteur	Eric Moreau	Professeur à l'Université de Toulon
Examinateur	Michel Broniatowski	Professeur à l'Université Paris VI
Examinateur	Emmanuel Duflos	Professeur à l'Ecole Centrale de Lille
Examinateur	Thierry Chonavel	Professeur à Telecom Bretagne
Directeur de thèse	Wojciech Pieczynski	Professeur à Telecom Sudparis

Thèse no 2012TELE0018

Table des Matières

Introduction générale

Au cours des dernières années, la modélisation statistique s'est peu à peu imposée dans tous les domaines relevant du traitement et la diffusion de l'information (traitement du signal, théorie de l'information, reconnaissance de formes, fouille de données, etc.). Dans ces différents contextes, les modèles de Markov cachés (plus souvent désignés par HMM pour « Hidden Markov Models ») constituent une classe de modèles à l'origine de l'une des plus grandes avancées. Le principe central de cette famille de modèles et de supposer l'existence d'un état caché évoluant suivant une dynamique markovienne, les observations étant des fonctions (déterministes ou aléatoires) de cet état caché. Les modèles (HMM) sont très flexibles du fait de l'introduction de variables non observées qui permettent de modéliser des structures de dépendances temporelles complexes.

Ce type de modèles s'adapte particulièrement au filtrage statistique qui consiste à estimer l'état d'un système dynamique, c'est-à-dire évoluant au cours du temps, à partir d'observations partielles, généralement bruitées.

Supposons que l'on dispose d'une suite $Y = (Y_1, ..., Y_N)$ d'observations, obtenues éventuellement après traitement préalable du signal bruité recueilli au niveau des capteurs. Chaque observation Y_n est reliée à l'état inconnu X_n par une relation du type $Y_n = F(X_n, V_n)$, où V_n est un « bruit » qui modélise l'erreur d'observation, et F une fonction a priori quelconque. Le filtrage consiste à restituer les états $X = (X_1, ..., X_N)$ à partir des mesures observées. Il est à noter que les états peuvent être des vecteurs de dimension qui peut être faible ou très grande. Lorsque l'on propose une méthode comme le filtrage, le filtre qui fournit une estimation de l'état à chaque instant à partir des mesures récoltées, doit pouvoir être implémenté dans un ordinateur, le temps de son exécution doit être raisonnable (voir un temps réel pour un grand nombre d'applications), avec une mémoire requise limitée. Les filtres récursifs sont alors préférés ; en effet, pour ces filtres, le calcul de l'estimée à l'instant courant ne dépend que de l'observation courante et de l'estimée à l'instant précédent.

Pour des systèmes dynamiques linéaires le filtre de Kalman a été développé dans les années 60 et 70; cependant, l'intérêt de cet outil pour la communauté statistique n'a été perçu que plus récemment (voir [11] [12][13]).

Dans les cas non linéaires on peut utiliser des extensions du filtre de Kalman, comme c'est le cas des deux algorithmes (E K F) et (U K F). Le premier linéarise le modèle dynamique autour d'une solution approchée et le second correspond à une interpolation autour de ce qu'on appelle les sigmas points. Lorsque le système est fortement non linéaire les différents algorithmes basés sur le filtre de Kalman peuvent diverger. Une autre difficulté s'ajoute lors de la propagation des densités conditionnelles c'est le calcul d'intégrales complexes multidimensionnelles. Pour faire face des méthodes dites de maillage ont été développées, mais lorsque la dimension de l'état est grande (supérieur à 3) ces méthodes sont coûteuses en temps.

Autour des années 90, les méthodes de Monte Carlo ont été proposées pour répondre à ces différents problèmes se posant dans les cas non linéaires. Ces méthodes sont fondées sur la loi des grands nombres, et ont des performances assez satisfaisantes au niveau du temps de calcul ; par ailleurs, elles sont peu sensibles à la dimension de l'état.

Les méthodes particulaires sont une version séquentielle des méthodes de Monte Carlo et sont bien adaptées aux problématiques de filtrage. Elles ont été introduites par Del Moral, Rigal, Salut, Gordan, Salmond et Smith. Elles proposent de représenter la loi conditionnelle de l'état

par un nombre fini de masses de Dirac pondérées. Les masses sont sur les points aléatoires obtenus par simulation des « particules », d'où le nom du « filtrage particulaire ». Ces particules évoluent suivant l'équation d'état du système (étape prédiction) et les poids sont ajustés en fonction des observations (étape correction).
Le filtre particulaire ainsi décrit pose cependant des problèmes car les poids des particules (les pondérations des masses de Dirac) tendent vers des valeurs très faibles, voir nulles. Après avoir lancé l'algorithme au bout d'un certain nombre d'itérations on observe une « dégénérescence des poids », ce qui fausse naturellement la représentation de la densité conditionnelle. Pour faire face à ce phénomène on introduit une étape dite de « ré-échantillonnage », qui permet de dupliquer les particules de poids fort et éliminer ceux des poids les plus faibles. Dans la littérature on trouvera de nombreuses versions de filtres particulaires qui ont été proposées dans ce cadre ; citons par exemple (SIR : Sampling Importance Resampling Filter) proposé par Gordon [14]. Malgré ces différentes retouches astucieuses on observe malgré tout dans plusieurs cas la divergence du filtre particulaire. Une des raisons est que l'approximation Monte Carlo qu'on effectue est en cascade, que se soit pour l'évaluation des intégrales ou dans l'étape de ré-échantillonnage. Pour répondre à ces problèmes il a été proposé un nouveau filtre appelé filtre de Kalman particulaire à noyaux (Kalman Particular Kernel Filter). Cette méthode consiste à approcher la densité conditionnelle par un mélange de gaussiennes au lieu d'une somme de Dirac. Ce type de filtre nécessite moins de particules et peut présenter moins de dégénérescence.
Dans toutes les approches que l'on vient de passer en revue on considère le modèle classique où le processus $X = (X_1, ..., X_N)$ est de Markov et les variables $Y_1, ..., Y_N$ sont indépendantes conditionnellement à X. De plus, pour tout $t = 1, ..., N$, $p(y_t|x) = p(y_t|x_t)$. On a donc

$$p(y|x) = \prod_{t=1}^{T} p(y_t|x_t) \qquad (1)$$

Il s'avère que dans de nombreux cas (1) représente une simplification relativement brutale de la réalité physique, ce qui peut être préjudiciable dans l'estimation de X. Afin d'y remédier, des modèles plus généraux, dits « chaînes de Markov couples » (CM Couple) ont été proposés et leur intérêt pratique, par rapport aux chaînes de Markov cachées (CM Cachées) classiques, s'est avéré non négligeable en segmentation d'images [49]. Dans une CM Couple on suppose directement la markovianité du couple $Z = (X, Y)$, ce qui implique que $(Y|X)$ et $(X|Y)$ sont de Markov. Cependant, bien que X ne soit pas nécessairement de Markov (on trouvera dans [49] des CNS pour que X soit markovien), on peut néanmoins utiliser les CM Couples pour faire du filtrage du type Kalman ou du particulaire malgré que les bruits ne soient plus nécessairement indépendants. Comme on vient de le voir les CM Couples généralisent les CM Cachées, dans ce même sens on pourra généraliser les CM Couples avec des chaînes de Markov triplets (CMT) où ni X, ni même $Z = (X, Y)$, ne sont pas nécessairement de Markov. Malgré la présence de ce troisième processus, qu'on appellera « intermédiaire », il peut tout de même être encore possible de développer des outils pour faire du filtrage du type Kalman ou du particulaire.

Le premier objectif des travaux dans le cadre de cette thèse est de proposer des méthodes de filtrage originales en utilisant les modèles classiques. Nous proposons deux nouvelles méthodes d'approximation dans le cas des systèmes linéaires gaussiens à sauts Markoviens. La première est fondée sur l'utilisation des chaînes de Markov cachées par du bruit à mémoire longue, récemment proposées dans [40][47]. On obtient une méthode « partiellement non

supervisée » dans laquelle certains paramètres, comme la matrice des transitions dans le processus markovien des sauts, peut être estimés. En comparant son efficacité avec celle des méthodes classiques (Kalman, particulaire), les résultats obtenus avec la nouvelle méthode sont comparables et très satisfaisant en temps de calcul. La deuxième exploite l'idée de ne garder que les trajectoires les plus probables ; là encore, on obtient une méthode très rapide donnant des résultats intéressants.

Enfin, nous proposons deux familles de modèles à sauts originaux. La première, où $Z = (X, Y)$ est une CM Couple conditionnellement aux sauts, est très générale, et nous proposons une extension du filtrage particulaire à cette famille. La deuxième, où le couple (sauts, observations) est markovien, est un cas particulier de la première avec la propriété intéressante selon laquelle le filtrage optimal exact est possible avec une complexité linéaire en temps. L'utilisation de la deuxième famille en tant qu'approximation de la première est alors testée et les premiers résultats semblent très encourageants.

L'organisation de la thèse est la suivante.

Le premier chapitre est consacré aux chaînes de Markov à espace d'état fini, particulièrement le cas des chaînes à deux états sera étudié en détails. On donnera des exemples d'algorithmes permettant de simuler des variables aléatoires à support fini et ensuite la simulation des trajectoires de chaîne de Markov pour d'éventuelles applications respectivement en dimension un, deux et trois.

Dans le deuxième chapitre on s'intéresse au modèle de chaînes de Markov cachées. On y rappelle les modèles classiques où le processus observé conditionnellement au processus caché est sans mémoire (les variables sont indépendantes conditionnellement à l'état caché) ainsi que les modèles récents où ce processus est à mémoire longue. Nous rapellons deux algorithmes d'inférence bayésienne le MPM et le MAP et deux algorithmes d'estimation de paramètres EM et ECI. Notre contribution ici concerne l'estimation « adaptative », qui permettra de mettre en place des filtrages « partiellement » non supervisés dans le chapitre 5.

Le troisième chapitre est consacré aux rappels de la problématique classique du filtrage optimal. Nous y traitons des systèmes linéaires gaussiens, des systèmes linéaires non gaussiens, des systèmes gaussiens non linéaires, ainsi que des différents filtrages optimaux ou sous-optimaux associés (filtre de Kalman, filtre particulaire, …).

Dans le quatrième chapitre nous proposons deux approximations du filtre optimal dans les systèmes classiques, qui constituent des contributions originales.

Dans le cinquième et dernier chapitre on propose deux contributions originales. D'abord, on étend le modèle classique du chapitre 4 à un modèle très général et une extension du filtrage particulaire à ce nouveau modèle. Ensuite, on étudie un cas particulier de cette extension dans laquelle le filtrage optimal exact est possible avec une complexité linéaire en temps.

Chapitre 1

Chaînes de Markov à espace d'états fini

Ce premier chapitre est consacré aux modélisations markoviennes les plus simples, que sont les chaînes à espace d'état fini. Dans le cadre de cette thèse ces modèles seront utilisés dans les chapitres 2, 4 et 5, où ils modéliseront la présence des sauts dans certains modèles de Markov triplets. Par ailleurs, ils sont utiles pour comprendre les algorithmes de simulations que nous utiliserons dans ce travail. Nous présentons quelques développements de base de ces modèles, la plupart sans démonstrations. Ces différents développements ne sont pas tous indispensables pour la suite de la thèse, mais peuvent être utiles pour mieux comprendre certains comportements des algorithmes présentés dans un cadre plus complexe, celui des modèles de Markov triplets. Nous mettons un accent particulier sur les chaînes à deux états, qui seront beaucoup utilisés dans les différentes expérimentations.
Historiquement, c'est Andrei Markov qui a publié les premiers résultats sur les chaînes de Markov à espace d'état fini en 1906 et 30 ans après, une généralisation aux espaces d'états infinis dénombrables a été publiée par Kolmogorov. Cette famille de processus, malgré sa simplicité, s'est avérée très efficace pour modéliser et étudier des phénomènes aléatoires très variés, d'où son utilisation dans de nombreuses applications. Citons la génétique, finance, économie, informatique, météorologie, traitement de la parole, traitement d'image, ….

1.1 Introduction

Soit $\aleph$ l'ensemble des nombres naturels et $X = (X_t)_{t\in\aleph}$ une suite de variables aléatoires à valeurs dans un ensemble fini $\Omega = \{\lambda_1, \lambda_2, ..., \lambda_K\}$, appelé « espace d'état », dont les éléments seront appelés « classes ». Pour $t \in \aleph$ donné, on peut représenter l'évolution de la chaîne X en fonction du temps avec un arbre déterministe de toutes les réalisations possibles, en considérant les espaces produits $\Delta_t = \left\{ x_0^t \;\middle|\; x_0^t = (x_0, x_1, ..., x_t) \in \Omega^{t+1} \right\}$. La construction naturelle de cet arbre consiste à tracer pas à pas, depuis un nœud originel, tous les chemins conduisant aux résultats envisageables. Le chemin $x_0^t = (x_0, x_1, ..., x_t) \in \Delta_t$ correspond ainsi à une branche, allant du nœud initial à l'un des nœuds à la hauteur 't'. Par ailleurs, on affecte à chacun des chemins élémentaires $(x_s \to x_{s+1})$ la probabilité de réalisation. Comme on va le voir par la suite dans le cas d'une chaîne de Markov la probabilité de passage à un nœud x_{s+1}, conditionnelle au chemin parcouru, dépend uniquement du nœud d'avant x_s.

Définition 1.1

On dira que la suite de variables aléatoires $(X_t)_{t\in\aleph}$ est une chaîne de Markov à valeurs dans $\Omega=\{\lambda_1,\lambda_2,...,\lambda_K\}$ si et seulement si pour tout $t\in\aleph$ et pour tout $(x_0,x_1,...,x_t,x_{t+1})\in\Omega^{t+2}$, on a:

$$\mathrm{P}(X_{t+1}=x_{t+1}\,|\,X_t=x_t,X_{t-1}=x_{t-1},\ldots,X_0=x_0)=\mathrm{P}(X_{t+1}=x_{t+1}\,|\,X_t=x_t) \quad (1.1)$$

Afin de simplifier, s'il n'y a pas d'ambigüité, les différentes lois associées à la chaîne $(X_t)_{t\in\aleph}$ comme $\mathrm{P}(X_0=x_0,X_1=x_1,\ \ldots,X_t=x_t)$, $\mathrm{P}(X_{t+1}=x_{t+1}\,|\,X_t=x_t)\ldots$ seront parfois notées $p(x_0,x_1,\ \ldots,x_t)$, $p(x_{t+1}\,|\,x_t)\ldots$

Définition 1.2

Une matrice stochastique est une matrice carrée dont chaque élément est un réel compris entre 0 et 1 et dont la somme des éléments de chaque ligne vaut 1.

La loi d'une chaîne de Markov est donc définie par la loi de X_0 et une suite des matrices stochastiques donnant les lois $\mathrm{P}(X_{t+1}=x_{t+1}\,|\,X_t=x_t)$. En effet, soit une famille de matrices stochastiques $Q_t=(Q_t(\lambda_i,\lambda_j))_{(i,j)\in\{1,\ldots,K\}^2}$. Si π_0 est la loi de X_0, et si on interprète les entrées de chaque ligne comme les probabilités de passage de l'état λ_i vers l'un des états possibles $\lambda_j\in\Omega$ à l'instant 't' : $Q_t(\lambda_i,\lambda_j)=\mathrm{P}(X_{t+1}=\lambda_j\,|\,X_t=\lambda_i)$, on a

$$p(x_0,x_1,...,x_T)=\pi_0(x_0)\times Q_1(x_0,x_1)\times Q_2(x_1,x_2)\times...\times Q_T(x_{T-1},x_T), \quad (1.2)$$

Cette formule découle directement de la formule des probabilités composées et de la propriété de Markov.

Exemple 1.1

L'arbre des trajectoires d'une chaîne de Markov à deux états et à horizon fixe T est présenté à la Figure 1.1.

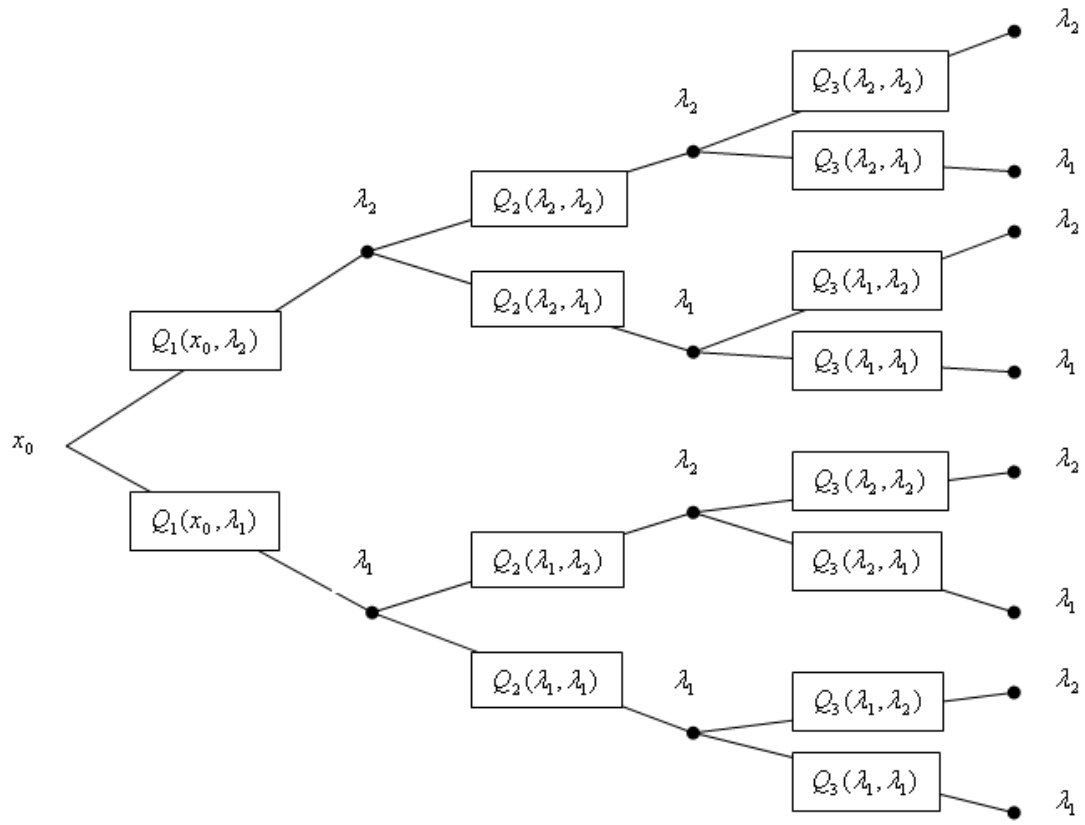


Figure 1.1 Représentation de l'arbre des trajectoires d'une chaîne de Markov pour $K = 2$ et $T = 3$.

1.2 Chaînes de Markov homogènes

Définition 1.3

Une chaîne de Markov $(X_t)_{t\in\aleph}$ est dite homogène (dans le temps) si $\forall t \in \aleph$, $\mathrm{P}(X_{t+1} = x_{t+1} \mid X_t = x_t)$ ne dépend pas de t.

Ainsi la loi d'une chaîne de Markov homogène est donnée par une probabilité π_0 et une seule matrice stochastique $Q = (q_{i,j})_{(i,j)\in\{1,\ldots,K\}^2}$, dite « matrice de transitions » ou « matrice de Markov».

Il est souvent pratique d'associer à une matrice de transition un graphe orienté, où les sommets sont les états de la chaîne et l'orientation est donnée par les probabilités $(q_{i,j})_{(i,j)\in\{1,\ldots,K\}^2}$. Deux sommets λ_i, λ_j sont alors reliés par une flèche allant de λ_i vers λ_j si et seulement si la probabilité de transition $q_{i,j}$ de λ_i vers λ_j est non nulle.

Exemple 1.2

Soient $K = 2$, $\Omega = \{\lambda_1, \lambda_2\}$ et $Q = \begin{pmatrix} 1-a & a \\ b & 1-b \end{pmatrix}$.

Le graphe orienté inter-états correspondant est le suivant:

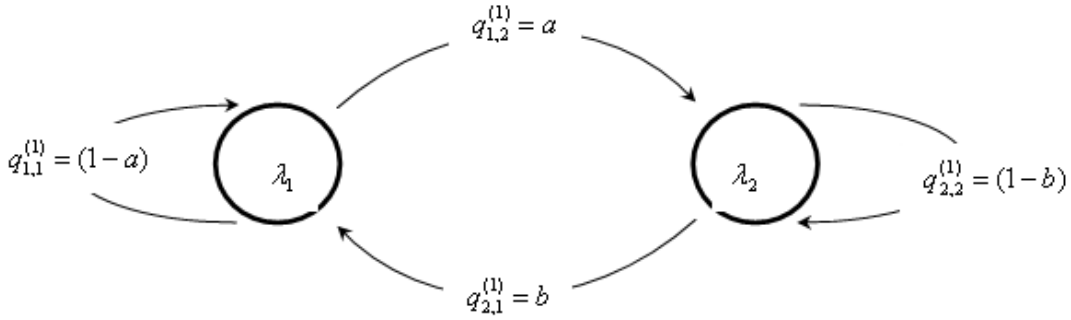


Figure 1.2 Exemple de graphe orienté associé à une chaîne à deux états.

Considérons une chaîne de Markov homogène avec une matrice de transitions Q. Pour tout $n \in \aleph$ on note $Q^{(n)} = (q_{i,j}^{(n)})_{(i,j)\in\{1,\dots,K\}^2}$ la matrice de transitions définie par $q_{i,j}^{(n)} = \mathrm{P}(X_{t+n} = \lambda_j \,|\, X_t = \lambda_i)$, et on note Q^n la nième puissance de Q. On a alors l'important résultat suivant, appelé théorème de « Chapman-Kolmogorov » :

Théorème 1.1

La matrice de transition Q d'une chaîne de Markov homogène vérifie:

$$\forall (n,m) \in \aleph \qquad Q^{(n+m)} = Q^{(n)} \times Q^{(m)} \tag{1.3}$$

$$\forall n \in \aleph \qquad Q^{(n)} = Q^n . \tag{1.4}$$

La démonstration de **(1.3)** découle de la propriété des probabilités totales, et celle de **(1.4)** est fondée sur **(1.3)**, elle peut être utilisée pour faire un raisonnement par récurrence.

Le lemme et la proposition qui suivent donnent un moyen pratique pour simuler une chaîne de Markov.

Lemme 1.1

Soit $U = (U_t)_{t\in\aleph}$ une suite de variables aléatoires indépendantes de même loi à valeurs dans un espace mesurable Λ. Soit X_0 une variable aléatoire à valeurs dans Ω et de loi π_0 indépendante de la suite U. Soit f une fonction mesurable définie sur $\Omega \times \Lambda$ à valeur dans Ω.

Alors la suite de variables aléatoires $X=(X_t)_{t\in\aleph}$ définie pour tout $t\in\aleph$ par $X_{t+1}=f(X_t,U_{t+1})$ est une chaîne de Markov homogène de distribution initiale π_0 et de matrice de transition

$$Q=(\mathrm{P}(f(\lambda_i,U_1)=\lambda_j))_{(i,j)\in\{1,...,K\}^2}=(q_{i,j})_{(i,j)\in\{1,...,K\}^2}. \qquad (1.5)$$

La preuve de ce lemme est immédiate, pour $X_t=x_t$ la variable $X_{t+1}=f(x_t,U_{t+1})$ dépend de U_{t+1}, et le vecteur $(X_0,...,X_{t-1})$ dépend du vecteur $(U_1,...,U_{t-1})$. Sachant que U_{t+1} et $(U_1,...,U_{t-1})$ sont indépendantes, $X_{t+1}=f(x_t,U_{t+1})$ et $(X_0,...,X_{t-1})$ le sont également (conditionnellement à $X_t=x_t$), d'où la markovianité.

Les transitions $q_{i,j}=\mathrm{P}(X_{t+1}=\lambda_j \mid X_t=\lambda_i)$ ne dépendent pas de 't' donc la chaîne X est bien une chaîne de Markov homogène de distribution initiale π_0 et de matrice de transitions $Q=(q_{i,j})_{(i,j)\in\{1,...,K\}^2}$.

On a également la réciproque, que nous donnons dans la forme suivante :

Proposition 1.1

Soit $X=(X_t)_{t\in\aleph}$ est une chaîne de Markov homogène à valeurs dans Ω, de matrice de transitions $Q=(q_{i,j})_{(i,j)\in\{1,...,K\}^2}$. Il existe une suite de variables aléatoires (i.i.d) $U=(U_t)_{t\in\aleph}$ à valeurs dans $[0,1]$ et une fonction f mesurable définie sur $\Omega\times[0,1]$ à valeurs dans Ω, indépendante de X_0 et telle que $X_{t+1}=f(X_t,U_{t+1})$.

Remarque 1.1

Si on se donne une matrice de transition Q, on construit une fonction f de mise à jour associée, on simule une suite $U=(U_t)_{t\in\aleph}$ de variables aléatoires (i.i.d) suivant une loi régulière donnée et on simule X_0 selon la distribution initiale π_0, alors la simulation de la chaîne de Markov $X=(X_t)_{t\in\aleph}$ se fait pas à pas suivant la relation récurrente $X_{t+1}=f(X_t,U_{t+1})$. Nous donnerons la constriction d'une telle fonction f dans la section 6.

Il est possible de calculer des espérances ou des lois conditionnelles pour une chaîne de Markov homogène en fonction des puissances de sa matrice de transition. Nous avons le résultat suivant :

Proposition 1.2

Soit $X=(X_t)_{t\in\aleph}$ une chaîne de Markov homogène de matrice de transition Q. On note π_t la loi de X_t et on considère f une fonction bornée définie sur Ω à valeur dans $\Re$. On a pour tout $t\in\aleph^*$:

1 $\pi_t = \pi_0\, Q^t$;

2 $\mathrm{E}[f(X_t)\,|\,X_0] = (Q^t\, f)(X_0)$;

3 $\mathrm{E}[f(X_t)\,|\,X_{t-1},...,X_0] = (Q\, f)(X_{t-1})$;

4 $\mathrm{E}[f(X_t)] = <\pi_t, f> = <(\pi_0\, Q^t), f> = <\pi_0, (Q^t f)>$.

Preuve:

- Le point 1 découle de la règle des causes totales ; on a :

$\forall t \in \aleph^*,\ \forall\ j \in \{1,...,K\},$

$$\pi_t^{(j)} = \sum_{i=1}^{K} \mathrm{P}(X_{t-1} = \lambda_i, X_t = \lambda_j) = \sum_{i=1}^{K} \mathrm{P}(X_t = \lambda_j \,|\, X_{t-1} = \lambda_i) \times \mathrm{P}(X_{t-1} = \lambda_i) = \sum_{i=1}^{K} \pi_{t-1}^{(i)} \times q_{i,j}$$

Sous forme matricielle on peut donc écrire $\pi_t = \pi_{t-1}\, Q$. Par itération de cette égalité on a: $\forall t \in \aleph,\ \pi_t = \pi_0\, Q^t$.

- Le point 2 découle de la définition de l'espérance conditionnelle et de l'équation de (***Chapman-Kolmogorov***), on a:

$$\forall t \in \aleph,\ \forall i \in \{1,...,K\},\ \mathrm{E}[f(X_t)\,|\,X_0 = \lambda_i] = \sum_{j=1}^{K} \mathrm{P}(X_t = \lambda_j \,|\, X_0 = \lambda_i) \times f(\lambda_j)$$

D'autre part $\forall t \in \aleph,\ \forall (i,j) \in \{1,...,K\}^2$

$$\mathrm{P}(X_t = \lambda_j \,|\, X_0 = \lambda_i) = (Q^t)_{i,j} = q_{i,j}^{(t)}$$

donc $\forall t \in \aleph,\ \forall i \in \{1,...,K\}$

$$\mathrm{E}[f(X_t)\,|\,X_0 = \lambda_i] = \sum_{j=1}^{K} q_{i,j}^{(t)} \times f(\lambda_j) = (Q^t f)^{(i)} = (Q^t f)(\lambda_i)$$

d'où le résultat.

- Le point 3 peut être déduit du résultat précédent, combiné avec la propriété de Markov. En effet, $\forall t \in \aleph,\ \forall (i,j) \in \{1,...,K\}^2$, et $\forall \omega_0^{t-2} = \{\omega_{t-2},...,\omega_0\} \in \Omega^{t-1}$, on a:

$$\mathrm{E}[f(X_t)\,|\,X_{t-1} = \lambda_i, X_0^{t-2} = \omega_0^{t-2}] = \sum_{j=1}^{K} \mathrm{P}(X_t = \lambda_j \,|\, X_{t-1} = \lambda_i,\ X_0^{t-2} = \omega_0^{t-2}) \times f(\lambda_j)$$

Or:

$$\mathrm{P}(X_t = \lambda_j \,|\, X_{t-1} = \lambda_i, X_0^{t-2} = \omega_0^{t-2}) = \mathrm{P}(X_t = \lambda_j \,|\, X_{t-1} = \lambda_i) = q_{i,j},$$

donc:

$$\mathrm{E}[f(X_t) \,|\, X_{t-1} = \lambda_i, X_0^{t-2} = \omega_0^{t-2}] = \sum_{j=1}^{K} q_{i,j} \times f(\lambda_j) = (Q\,f)^i = (Q\,f)(\lambda_i),$$

d'où le résultat.

- Le dernier point 4 se démontre à partir des points précédents, on a:

$$\forall t \in \aleph,\ \mathrm{E}[f(X_t)] = \sum_{j=1}^{K} \mathrm{P}(X_t = \lambda_j) \times f(\lambda_j) = <\pi_t, f>$$

Du point 1 on déduit directement que pour tout $t \in \aleph : \mathrm{E}[f(X_t)] = <(\pi_0\, Q^t), f>$;
on a alors pour tout $t \in \aleph$:

$$\mathrm{E}[f(X_t)] = \sum_{j=1}^{K} \pi_t^{(j)} \times f(\lambda_j) = \sum_{j=1}^{K} \left(\sum_{i=1}^{K} \pi_0^{(i)} q_{i,j}^{(t)} \right) \times f(\lambda_j)$$

$$= \sum_{i=1}^{K} \pi_0^{(i)} \left(\sum_{j=1}^{K} q_{i,j}^{(t)} \times f(\lambda_j) \right) = \sum_{i=1}^{K} \pi_0^{(i)} (Q^t f)^{(i)} = <\pi_0, (Q^t f)>$$

Remarque 1.2

Notons le résultat suivant, très utile en pratique, qui découle du premier point :

$$\forall (s,t) \in \aleph,\ \pi_{t+s} = \pi_0 \times Q^{t+s} = (\pi_0 \times Q^s) \times Q^t = \pi_s \times Q^t$$

1.3 Chaînes de Markov irréductibles

L'irréductibilité est une notion qui nous permet de s'assurer que tous les états possibles peuvent être atteints par la chaîne. Pour traduire cette notion on parle également des états qui « communiquent », sachant que deux états communiquent si la probabilité d'aller de l'un à l'autre est strictement positive. On dira qu'une chaîne de Markov est irréductible si tous les états possibles communiquent entre eux. Pour chercher les états qui communiquent il est souvent plus commode de travailler sur le graphe de la chaîne plutôt que sur la matrice de transitions. Une chaîne de Markov est alors irréductible si et seulement si il existe un chemin fermé dans le graphe passant au moins une fois par tous les états de la chaîne. Un tel graphe est dit « fortement connexe ». Ainsi un graphe (ou un sous-graphe) est fortement connexe si tous ses sommets communiquent entre eux.
On peut décomposer alors un graphe quelconque des états d'une chaîne de Markov en une partition de sous-graphes, qui contient deux types qui sont fortement connexes :

1. Ceux qui ne communiquent pas avec l'extérieur : aucun chemin du sous-graphe ne permet d'atteindre un sommet externe à ce dernier. Ce sont des classes dites fermées, on les appelle aussi "classes finales".

2. Ceux qui communiquent avec l'extérieur : à partir desquelles on peut atteindre au moins un sommet externe.

Définition 1.4

Soit $(X_t)_{t\in\aleph}$ une chaîne de Markov homogène à valeurs dans $\Omega = \{\lambda_1, \lambda_2, ..., \lambda_K\}$.

(i) Un état $\lambda_j \in \Omega$ est dit « accessible », « atteignable », à partir de l'état $\lambda_i \in \Omega$ si et seulement si il existe $n \in \aleph$ tel que $\mathrm{P}(X_n = \lambda_j | X_0 = \lambda_i) > 0$;

(ii) On dit que les états λ_i et λ_j communiquent, et on note $\lambda_i \leftrightarrow \lambda_j$, si et seulement si λ_j est accessible à partir de λ_i (on note $\lambda_i \to \lambda_j$) et λ_i est accessible à partir de λ_j ($\lambda_j \to \lambda_i$) ;

(iii) Les classes d'équivalence correspondantes à la relation $\lambda_i \leftrightarrow \lambda_j$ sont dites « classes d'états communicants ».

En introduisant dans l'ensemble des classes d'états communicants la relation d'ordre suivante : $C_1 \le C_2$ s'il existe $\lambda_i \in \mathrm{C}_1$ et $\lambda_j \in \mathrm{C}_2$ tels que $\lambda_i \to \lambda_j$, on peut montrer qu'avec la probabilité égale à un la chaîne finit, après un temps fini, dans une des classes terminales pour cette relation d'ordre partiel.
En pratique, nous aurons à faire aux chaînes « irréductibles », où tous les états communiquent. Nous avons la définition suivante

Définition 1.5

Une chaîne de Markov dont tous les éléments de l'espace d'état communiquent est dite « irréductible ».

Notons qu'une définition équivalente consiste à dire que l'espace Ω est réduit à une seule classe d'états communicants. On peut affirmer que la notion d'irréductibilité dépend uniquement des probabilités de transition et non de la loi initiale.

1.4 Stationnarité et comportement asymptotique

Comme précisé précédemment on s'intéresse exclusivement aux chaînes de Markov homogènes à espace d'états fini. Dans ce paragraphe on va étudier la stationnarité de ces chaînes ainsi que leur « comportement asymptotique »: le but est de décrire la loi de X_t lorsque $t \to +\infty$.

1.4.1 Stationnarité stricte

Une chaîne aléatoire $X = (X_t)_{t\in\aleph}$ à valeurs dans un ensemble fini $\Omega = \{\lambda_1, \lambda_2, ..., \lambda_K\}$ est dite « stationnaire d'ordre $n+1$ » si

$\forall (x_0, x_1, ..., x_n) \in \Omega^{n+1}$, $\forall (t_0, t_1, ..., t_n) \in \aleph^{n+1}$ et $\forall\, s \in \aleph$ on a

$$\mathrm{P}(X_{t_0+s} = x_0, X_{t_1+s} = x_1, ..., X_{t_n+s} = x_n) = \mathrm{P}(X_{t_0} = x_0, X_{t_1} = x_1, ..., X_{t_n} = x_n)$$

La loi conjointe du vecteur aléatoire $(X_{t_0+s}, X_{t_1+s}, ..., X_{t_n+s})$ est ainsi la même que celle du vecteur $(X_{t_0}, X_{t_1}, ..., X_{t_n})$.

Une chaîne est stationnaire au sens strict (ou fortement stationnaire) si $\forall\, n \in \aleph$ elle est stationnaire à l'ordre $n+1$.

On note qu'une chaîne de Markov stationnaire d'ordre 2 est fortement stationnaire.

1.4.2 Distribution stationnaire et mesure invariante

Soit un espace d'état fini $\Omega = \{\lambda_1, \lambda_2, ..., \lambda_K\}$; une loi de probabilité sur Ω sera considéré comme étant un vecteur ligne $\pi = (\pi^{(1)}, ..., \pi^{(K)})$ composé des probabilités des différents états possibles.

Définition 1.6

Soit $(X_t)_{t\in\aleph}$ une chaîne de Markov homogène à valeurs dans $\Omega = \{\lambda_1, \lambda_2, ..., \lambda_K\}$, de matrice des transitions Q. Une probabilité $\pi = (\pi^{(1)}, ..., \pi^{(K)})$ est invariante par Q (ou est une « distribution stationnaire » de Q) si et seulement si elle vérifie l'équation matricielle $\pi Q = \pi$.

Le vecteur ligne π est donc un vecteur propre à gauche pour la valeur propre 1.
On vérifie aisément la propriété suivante :

Proposition 1.4

Une chaîne de Markov homogène de matrice de transition Q est fortement stationnaire si et seulement si la loi X_0 est invariante par Q.

Nous avons alors le résultat important suivant :

Théorème 1.2

Une chaîne de Markov homogène irréductible admet une probabilité invariante.

Précisons la notion de réversibilité. Soit π une probabilité invariante par $Q = (q_{k,l})_{(k,l)\in\Omega^2}$. Pour tout $k \in \{1, ..., K\}$, on a :

$$\pi_l = \sum_{k=1}^{K} \pi_k \, q_{k,l}$$

La matrice $\widetilde{Q} = (\widetilde{q}_{k,l})_{(k,l)\in\Omega^2} = (\mathrm{P}(X_t = \lambda_k \mid X_{t+1} = \lambda_l))_{(k,l)\in\Omega^2}$ est alors également une matrice stochastique. En effet, $\forall t \in \aleph$, et $\forall (k,l) \in \{1, ..., K\}^2$ nous avons,

$$\widetilde{q}_{l,k} = \mathrm{P}(X_t = \lambda_k \mid X_{t+1} = \lambda_l) = \frac{\mathrm{P}(X_t = \lambda_k, X_{t+1} = \lambda_l)}{\mathrm{P}(X_{t+1} = \lambda_l)} =$$

$$= \frac{\mathrm{P}(X_{t+1} = \lambda_l \mid X_t = \lambda_k)\mathrm{P}(X_t = \lambda_k)}{\mathrm{P}(X_{t+1} = \lambda_l)} = \frac{\mathrm{P}(X_t = \lambda_k)}{\mathrm{P}(X_{t+1} = \lambda_l)} \mathrm{q}_{k,l} = \frac{\pi_k}{\pi_l} q_{k,l};$$

On a bien $\sum_{k=1}^{K} q_{l,k} = 1$, donc $\widetilde{Q}$ est une matrice de Markov.

La matrice $\widetilde{Q}$ s'interprète comme la matrice de transition de la chaîne X après « retournement du temps ».

Définition 1.7

On dit qu'une chaîne de Markov homogène de matrice de transition $Q = (q_{k,l})_{(k,l)\in\Omega^2}$, est réversible par rapport à la probabilité π si $\forall (k,l) \in \{1, ..., K\}^2 \quad \pi_k \, q_{k,l} = \pi_l \, q_{l,k}$.

Remarque 1.3

Il découle directement de la dernière définition que si une chaîne de Markov est réversible par rapport à une probabilité π, alors cette probabilité est invariante.

1.4.3 Théorème ergodique pour chaînes irréductible finies

Soit $(X_t)_{t\in\aleph}$ une chaîne de Markov homogène et irréductible, à valeurs dans $\Omega = \{\lambda_1, \lambda_2, ..., \lambda_K\}$. On dit qu'elle est de période $d \in \aleph^*$ si, en partant d'un $\lambda \in \Omega$, les seules possibilités de retour à λ sont aux instants multiples de d. On montre que d ne dépend pas de λ (plus généralement, lorsqu'il y a plusieurs classes d'équivalence d'états communicants, on montre que chaque classe admet une période qui est la même pour tous ses éléments).

Lorsque $d = 1$ la chaîne est dite « apériodique ».

Sachant que $X_0 = \lambda \in \Omega$; on notera τ_λ la variable aléatoire « temps du premier retour à λ » donnée par :

$$\tau_\lambda = Inf\left\{ t \geq 1 \ / \ X_t = \lambda \right\}.$$

L'espérance $\mathrm{E}\left[\tau_\lambda \,\middle|\, X_0 = \lambda \right]$ donne le temps moyen du premier retour à λ.

On notera N_λ la variable aléatoire « nombre de visite en λ » donnée par :

$$N_\lambda = \sum_{t=1}^{+\infty} 1_{\{X_t = \lambda\}}.$$

Définition 1.8

Soit $(X_t)_{t \in \aleph}$ une chaîne de Markov homogène à valeurs dans Ω fini.

(i) L'état $\lambda \in \Omega$ est dit « absorbant » si et seulement si la probabilité de le quitter est nulle ;

(ii) Un état $\lambda \in \Omega$ est dit « transitoire » si et seulement si $\mathrm{P}\left[\tau_\lambda < +\infty \,\middle|\, X_0 = \lambda \right] < 1$;

(iii) Un état $\lambda \in \Omega$ est dit « récurrent » si et seulement si $\mathrm{P}\left[\tau_\lambda < +\infty \,\middle|\, X_0 = \lambda \right] = 1$.

La démonstration des trois résultats suivants peut être consultée dans [39].

Théorème 1.3

Soit X une chaîne de Markov homogène irréductible. Alors

(i) Pour tout $\lambda \in \Omega$, $\dfrac{1}{T} \sum_{t=1}^{T} 1_{\{X_t = \lambda\}} \xrightarrow[T \to +\infty]{} \pi(\lambda) = \dfrac{1}{\mathrm{E}\left[\tau_\lambda \,\middle|\, X_0 = \lambda \right]}$ *p.s.* ;

(ii) Le vecteur π défini dans ***(i)*** est l'unique probabilité invariante de la chaîne. De plus, $\pi(\lambda) > 0$ pour tout $\lambda \in \Omega$;

(iii) Pour toute fonction f positive définie sur Ω, si $\langle \pi, f \rangle = \sum_{\lambda \in \Omega} \pi(\lambda) f(\lambda) < \infty$, alors:

$$\frac{1}{T} \sum_{t=1}^{T} f(X_t) \xrightarrow[T \to +\infty]{} \langle \pi, f \rangle = \sum_{\lambda \in \Omega} \pi(\lambda) f(\lambda) \quad p.s.$$

Théorème 1.4: ***(Théorème de Kolmogorov cas ergodique)***

Soit X une chaîne de Markov ergodique (irréductible de période $d = 1$) de probabilité invariante unique π. Alors, on a

$$\forall (k,l) \in \left\{1, ..., K\right\}^2, \ \lim_{t \to +\infty} Q^t(\lambda_k, \lambda_l) = \lim_{t \to +\infty} q_{k,l}^{(t)} = \pi(\lambda_l)$$

Remarque 1.4

Dans le cas où la chaîne est irréductible à espace d'état fini (récurrente positive) mais de période $d > 1$, l'espace d'état admet une répartition $\{C_0,...,C_{d-1}\}$ en d classes disjointes « classes cycliques ». Dans ce cas $Q^{t\times d}(\lambda_k,\lambda_l) \neq 0$ si et seulement si λ_k et λ_l appartiennent à la même classe. On peut montrer également que :

$$\sum_{\lambda \in C_0} \pi(\lambda) = \frac{1}{d}, \text{ et } \forall (k,l) \in \{1,...,K\}^2 \ \lim_{t\to+\infty} Q^{t\times d}(\lambda_k,\lambda_l) = d \times \pi(\lambda_k)$$

1.5 Chaîne de Markov à deux états

Dans le cas d'une chaîne $(X_t)_{t\in\aleph}$ à deux états ($K = 2$) considérée dans ce paragraphe nous adoptons les notations suivantes :

- $\Omega = \{\lambda_1,\lambda_2\}$, $Q = \begin{pmatrix} 1-a & a \\ b & 1-b \end{pmatrix}$, avec $(a,b) \in [0,1]\times[0,1]$;
- pour tout $k \in \{1,2\}$, $\tau_{\lambda_k} = \tau_k$ et $N_{\lambda_k} = N_k$.

1.5.1 Irréductibilité

La chaîne $(X_t)_{t\in\aleph}$ est irréductible pour tout $(a,b) \in]0,1]\times]0,1]$. En effet, si $0 < a\,b < 1$ alors Ω est une classe communicante fermée irréductible et on a:

$$p_{k,k} = \mathrm{P}(T_k < \infty \mid X_0 = \lambda_k) = 1, \mathrm{P}(N_k = \infty \mid X_0 = \lambda_k) = 1$$

Les deux états de la chaîne sont alors récurrents positifs, et donc la chaîne est irréductible récurrente positive. Dans ce cas la chaîne est également apériodique, donc ergodique. On peut également le constater en notant que la matrice de transition est régulière.

Si $a \in [0,1]$ et $b = 0$, l'état λ_2 est un état absorbant (donc récurrent), et il s'agit d'une chaîne absorbante. L'état λ_1 admet une classification selon les valeurs prise par a. Nous avons:

$$p_{k,k} = \begin{cases} 0 & si \quad k=1 \ et \ a=1 & (\lambda_1 \ est \ transitoire) \\ \ell_1 \in]0,1[& si \quad k=1 \ et \ a \in]0,1[& (\lambda_1 \ est \ transitoire) \\ 1 & si \quad k=1 \ et \ a=0 & (\lambda_1 \ est \ absorbant) \\ 1 & si \quad k=2 \ et \ a \in [0,1] & (\lambda_2 \ est \ absorbant) \end{cases}$$

D'autre part

$$\mathrm{P}(N_k = \infty \,|\, X_0 = \lambda_k) = \begin{cases} \ell_2 \in [0,1[& si \quad k = 1 \\ 1 & si \quad k = 2 \end{cases}$$

Les composantes a et b jouent des rôles symétriques dans la matrice de transition, donc on pourra conclure de manière analogue si $a = 0$ et $b \in [0,1]$.

Remarque 1.5

Si $a = 1$ et $b = 1$ alors les deux états λ_1 et λ_2 sont récurrents positifs et la chaîne est dite « oscillante ». Elle est dans ce cas périodique de période $d = 2$, avec $Q^2 = Q^4 = Q^6 = \cdots = I_2$.

1.5.2 Comportement asymptotique et distribution invariante

1.5.2.1 Puissances de la matrice de transition

On montre aisément (par exemple par récurrence) que pour $(a+b) \neq 0$ on a

$$\forall n \in \aleph, \quad Q^n = \begin{pmatrix} q_{1,1}^{(n)} & q_{1,2}^{(n)} \\ q_{2,1}^{(n)} & q_{2,2}^{(n)} \end{pmatrix} = \frac{1}{a+b} \begin{pmatrix} b + a\,(1-a-b)^n & a - a\,(1-a-b)^n \\ b - b\,(1-a-b)^n & a + b\,(1-a-b)^n \end{pmatrix}$$

$$= \frac{1}{a+b} \begin{pmatrix} b & a \\ b & a \end{pmatrix} + \frac{(1-a-b)^n}{a+b} \begin{pmatrix} a & -a \\ -b & b \end{pmatrix}$$

On peut également démontrer ce résultat en diagonalisant Q. La matrice Q admet comme valeurs propres 1 et $(1-a-b)$, avec les vecteurs propres $(1,\ 1)'$ et $(a,\ -b)'$. En posant $D = \begin{pmatrix} 1 & 0 \\ 0 & 1-a-b \end{pmatrix}$ et $P = \begin{pmatrix} 1 & a \\ 1 & -b \end{pmatrix}$, on a, pour $(a+b) \neq 0$, $Q = P^{-1} D P$ et donc pour tout $n \in \aleph$ on a $Q^n = P^{-1} D^n P$, ce qui mène au résultat.

L'hypothèse $(a+b) \neq 0$ n'est pas forte car si $(a+b) = 0$, cela correspond à $a = b = 0$. Alors $Q = I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ et $Q^n = I_2^n = I_2$.

Un autre cas particulier à préciser est $a = b = 1$, ce qui correspond à $Q = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, et qui donne $Q^n = \begin{cases} Q & si\ n\ impaire \\ I_2 & si\ n\ paire \end{cases}$.

Les deux derniers cas extrêmes qu'on vient d'exposer correspondent à des chaînes de Markov à comportement déterministe pour un état initial donné.

1.5.2.2 Calcul de la loi et comportement à l'infini $(t \to +\infty)$

Le premier point de la Proposition 1.3 nous permet de calculer explicitement la loi π_t de X_t. En effet, on a $\pi_t = \pi_0 Q^t$ pour tout $t \in \aleph$. Donc, pour $(a+b) \neq 0$ on a:

$$\pi_t = \left(\pi_t^{(1)} \quad \pi_t^{(2)}\right) = \frac{1}{a+b}\left(\pi_0^{(1)} \quad \pi_0^{(2)}\right)\begin{pmatrix} b + a(1-a-b)^t & a - a(1-a-b)^t \\ b - b(1-a-b)^t & a + b(1-a-b)^t \end{pmatrix},$$

ce qui donne

$$\begin{cases} \pi_t^{(1)} = \dfrac{1}{a+b}\left[\left(\pi_0^{(1)} + \pi_0^{(2)}\right)b + \left(a\,\pi_0^{(1)} - b\,\pi_0^{(2)}\right)(1-a-b)^t\right] \\ \pi_t^{(2)} = \dfrac{1}{a+b}\left[\left(\pi_0^{(1)} + \pi_0^{(2)}\right)a - \left(a\,\pi_0^{(1)} - b\,\pi_0^{(2)}\right)(1-a-b)^t\right] \end{cases}$$

Sachant que $\pi_0^{(1)} + \pi_0^{(2)} = 1$, il vient:

$$\begin{cases} \pi_t^{(1)} = \dfrac{b}{a+b} + (1-a-b)^t\left(\pi_0^{(1)} - \dfrac{b}{a+b}\right) \\ \pi_t^{(2)} = \dfrac{a}{a+b} + (1-a-b)^t\left(\pi_0^{(2)} - \dfrac{a}{a+b}\right) \end{cases}$$

Nous en déduisons le comportement de la loi lorsque $(t \to +\infty)$:

1 Si $0 < a+b < 2$ on a $|1-(a+b)| < 1$ alors $\begin{cases} Q^t \xrightarrow[t\to+\infty]{} \overline{Q} = \dfrac{1}{a+b}\begin{pmatrix} b & a \\ b & a \end{pmatrix} \\ \pi_t \xrightarrow[t\to+\infty]{} \overline{\pi} = \left(\dfrac{b}{a+b}, \dfrac{a}{a+b}\right) \end{cases}$

Remarques

- La limite de la loi lorsque $(t \to +\infty)$ est indépendante de la loi initiale ;
- La matrice limite est composée de deux lignes identiques ;
- On a l'équivalence suivante : $[0 < a+b < 2] \Leftrightarrow [0 < a\,b < 1]$.

Pour les deux cas extrêmes $a+b=0$ et $a+b=2$ (qui équivalent, respectivement, à $a=b=0$ et $a=b=1$) nous avons :

2 Si $a+b=0$ alors $\begin{cases} Q^t = I_2 = \overline{Q} \\ \pi_t = \pi_0 = \overline{\pi} \end{cases} \quad \forall\, t \in \aleph$

3 Si $a+b=2$, alors on n'a pas de convergence (il s'agit d'une chaîne oscillante): on a pour tout $t \in \aleph$:

$$\begin{cases} Q^{2\times t} = I_2 \\ Q^{2\times t+1} = Q \end{cases}, \begin{cases} \pi_{2\times t} = \pi_0 \\ \pi_{2\times t+1} = (1-\pi_0^{(1)}, 1-\pi_0^{(2)}) \end{cases}$$

On remarque que lorsque $\pi_0 = (0.5, 0.5)$, on a $\pi_{2\times t} = \pi_{2\times t+1} = \pi_0$.

1.5.2.3 Loi stationnaire

Dans le cas où $0 < a\,b < 1$ la chaîne est ergodique ; on peut alors appliquer le théorème de Kolmogorov et la probabilité invariante n'est autre que la probabilité limite $\overline{\pi} = \left(\frac{b}{a+b}, \frac{a}{a+b}\right)$. On peut vérifier ce résultat en utilisant la définition de la distribution invariante en résolvant le système suivant:

$$\begin{cases} \pi = \pi \times Q \\ \pi^{(1)} + \pi^{(2)} = 1 \end{cases}, \text{ avec } \pi = (\pi^{(1)}, \pi^{(2)})$$

On montre aisément que la solution unique de ce système n'est autre que $\overline{\pi}$.

Dans les deux autres cas nous avons:

- Si $a = b = 0$ toutes les distributions de probabilité sont stationnaires.
- Si $a = b = 1$ l'unique distribution de probabilité stationnaire est $\pi = (0.5, 0.5)$.

1.5.2.4 Chaîne réversible

On va chercher sous quelle condition la chaîne de Markov est réversible, c'est-à-dire sous quelle condition il existe une distribution de probabilité $\nu = (\nu^{(1)}, \nu^{(2)})$ telle que:

$$\forall (k,l) \in \{1,\ 2\}^2 \qquad \nu^{(k)} \times q_{k,l} = \nu^{(l)} \times q_{l,k}$$

Si une telle distribution de probabilité existe elle est alors solution du système

$$\begin{cases} a\times\nu^{(1)} = b\times\nu^{(2)} \\ \nu^{(1)} + \nu^{(2)} = 1 \end{cases} \Leftrightarrow \begin{cases} (a+b)\times\nu^{(1)} = b \\ (a+b)\times\nu^{(2)} = a \end{cases} \Leftrightarrow \begin{cases} \nu = \left(\frac{b}{a+b}, \frac{a}{a+b}\right) & si \quad 0 < a\,b < 1 \\ \nu: \text{ arbitraire} & si \quad a = b = 0 \end{cases}$$

Donc la chaîne est réversible par rapport à sa distribution de probabilité stationnaire $\nu = \overline{\pi}$.

1.6 Algorithmes de simulation

La simulation des réalisations des chaînes de Markov est un puissant outil de calcul. Nous supposons que l'on dispose d'un générateur de variables aléatoires indépendantes de loi uniforme sur $[0,1]$. Nous allons expliciter l'une des méthodes qui permet d'opérer des transformations sur ces variables afin de produire une variable aléatoire possédant une loi donnée. Elle permet alors la simulation des réalisations des chaînes de Markov finies. En pratique on possède un générateur de variables « pseudo-aléatoires », qu'on peut dire approximativement aléatoire. On donnera par la suite un exemple d'algorithme pour générer des nombres pseudo-aléatoires dans $[0,1]$, qui seront considérés comme un échantillon issu d'une distribution uniforme [38].

1.6.1 Lois discrètes à support fini

Soit X une variable aléatoire à valeurs dans $\Omega = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$, de loi

$$p_k = \mathrm{P}(X = \lambda_k), \quad \forall\, k \in \{1, \ldots, K\}$$

On pose $s_0 = 0$ et $s_l = \sum_{k=1}^{l} p_k$ pour $1 \le l \le K$. On a, en particulier, $s_K = \sum_{k=1}^{K} p_k = 1$. Les points $s_0, s_1, \ldots, s_K$ induisent une partition de l'intervalle $[0,1]$. Par ailleurs, soit U une variable aléatoire de loi uniforme sur $[0,1]$. On a alors

$$\forall\, k \in \{1, \ldots, K\},\ \mathrm{P}\left(U \in [s_{k-1}, s_k[\right) = s_k - s_{k-1} = p_k\,,$$

la variable $\widetilde{X} = \sum_{k=1}^{K} \lambda_k \, 1_{[s_{k-1},\, s_k[}(U)$ est donc de même loi que X.

Comme on a supposé qu'on peut générer une variable aléatoire de loi uniforme sur $[0,1[$, alors par la transformation proposée ci-dessus X peut l'être aussi à partir de U. Notons que l'indice de l'état qu'on choisira comme réalisation pour X n'est autre que l'indice de l'intervalle contenant la réalisation de la variable U.

L'algorithme se déroule de la manière suivante.

Algorithme 1.1: Générer des nombres pseudo-aléatoires $u \sim \boldsymbol{U}_{[0,1[}$

- Choisir $u_0 \in [0,1[$ comme valeur initiale (germe).
- Fixer des nombres entiers e_0, e_1 et d ;
- On construit la suite de nombres $(w_n)_{n \in \mathbb{N}}$ itérativement:

$$w_{n+1} = e_1\, w_n + e_0 \qquad \text{Modulo} \quad d$$

Alors :

$$w_{n+1} \in \{1, ..., d-1\}$$

- On affecte à u_{n+1} la valeur $\frac{w_{n+1}}{d}$.

Alors:

$$u_{n+1} \in [0,1[$$

Approximation

On considère $u_{n+1} \sim \boldsymbol{U}_{[0,1[}$

Fin

L'algorithme qui suit est un moyen efficace de simulation de loi à support fini

---

Algorithme 1.2: Méthode de découpage d'intervalle pour simuler $x \sim \sum_{k=1}^{K} \pi^{(k)} \delta_{\lambda_k}$.

- Construire le vecteur $s = [s_0, s_1, \ldots, s_K]$;

$$s_0 = 0, \qquad s_l = \sum_{k=1}^{l} \pi^{(k)} \quad \text{pour} \quad 1 \le l \le K$$

- Générer $u \sim \boldsymbol{U}_{[0,1[}$.
- Pour $k = 1$ jusqu'à $k = K$ faire

Si $u \in [s_{k-1}, s_k[$ alors:

$x = \lambda_k$

Fin-(si)

Fin-(pour)

Fin

1.6.2 Simulation de Trajectoire d'une chaîne de Markov fini

Soit $X = (X_t)_{t \in \mathbb{N}}$ chaîne de Markov homogène à valeurs dans $\Omega = \{\lambda_1, \lambda_2, ..., \lambda_K\}$, de matrice de transition $Q = (q_{i,j})_{i,j \in \{1,...,K\}^2}$ et de distribution initiale $\pi_0 = \left(\pi_0^{(1)}, ..., \pi_0^{(K)}\right)$.

La méthode qu'on va utiliser repose sur la Proposition 1.2 ; elle fournie un moyen de simuler une chaîne à partir d'une suite de nombres aléatoires compris entre 0 et 1.

On commence par simuler la variable initiale X_0. On pose:

$$s_0 = 0, \qquad s_l = \sum_{k=1}^{l} \pi_0^{(k)} \qquad \text{pour} \quad 1 \le l \le K$$

$$S_{k,0} = 0, \qquad S_{k,l} = \sum_{k=1}^{l} q_{k,l} \qquad \text{pour} \quad (k,l) \in \{1,...,K\}^2$$

On définit, pour tous $(k,l) \in \{1,...,K\}^2$ et $u \in [0,1[$ une fonction f dite de mise à jour associée à Q, tel que:

$$f_0(u) = \lambda_k \qquad \forall u \in [s_{k-1}, s_k[\qquad (1.6)$$

$$f(\lambda_k, u) = \lambda_l \qquad \forall u \in [S_{k,(l-1)}, S_{k,l}[\qquad (1.7)$$

Proposition 1.5:

Soient $U_0, U_1, \ldots, U_T$ des variables aléatoires indépendantes et identiquement distribuées sur $[0,1[$. Le processus $X = (X_t)_{t \in \mathbb{N}}$ définie par:

$$X_0 = f_0(U_0), \text{ et } X_{t+1} = f(X_t, U_{t+1}) \text{ pour } t = 1,...,N-1,$$

avec f_0 et f vérifiant respectivement (1.6) et (1.7), est une chaîne de Markov à valeurs dans Ω.

Remarque 1.6

Soit $U \sim \boldsymbol{U}_{[0,1[}$ un résultat qui découle directement de ce dernier théorème:

$$\forall\, k \in \{1,...,K\} \qquad f(\lambda_k, U) \sim \sum_{l=1}^{K} q_{k,l}\, \delta_{\lambda_l}$$

Algorithme 1.3: Simulation d'une chaîne de Markov jusqu'à un horizon fixe T.

- On simule $(T+1)$ variables aléatoires $(u_0, u_1, ..., u_T)$ distribuées uniformément sur l'intervalle $[0,1[$;
- On construit les deux fonctions f_0 et f comme définit dans (1.6) et (1.7).
- La simulation de la variable initiale correspond à $x_0 = f_0(u_0)$.

- Pour $t = 1$ jusqu'à $t = T$ poser

$$x_t = f(x_{t-1}, u_t)$$

Fin-(pour)

Fin

Une version fondée sur l'algorithme ***(1.2)*** correspond à l'algorithme qui suit.

Algorithme 1.4: Simulation d'une chaîne de Markov jusqu'à un horizon fixe T.

- On simule la variable initiale $x_0 \sim \sum_{k=1}^{K} \pi_0^{(k)} \delta_{\lambda_k}$.

- Pour $t = 1$ jusqu'à $t = T$ faire

  Pour $k = 1$ jusqu'à $k = K$ faire

  Si $x_{t-1} = \lambda_k$ alors:

$$x_t \sim \sum_{l=1}^{K} q_{k,l} \delta_{\lambda_l}$$

  Fin-(si)

  Fin-(pour)

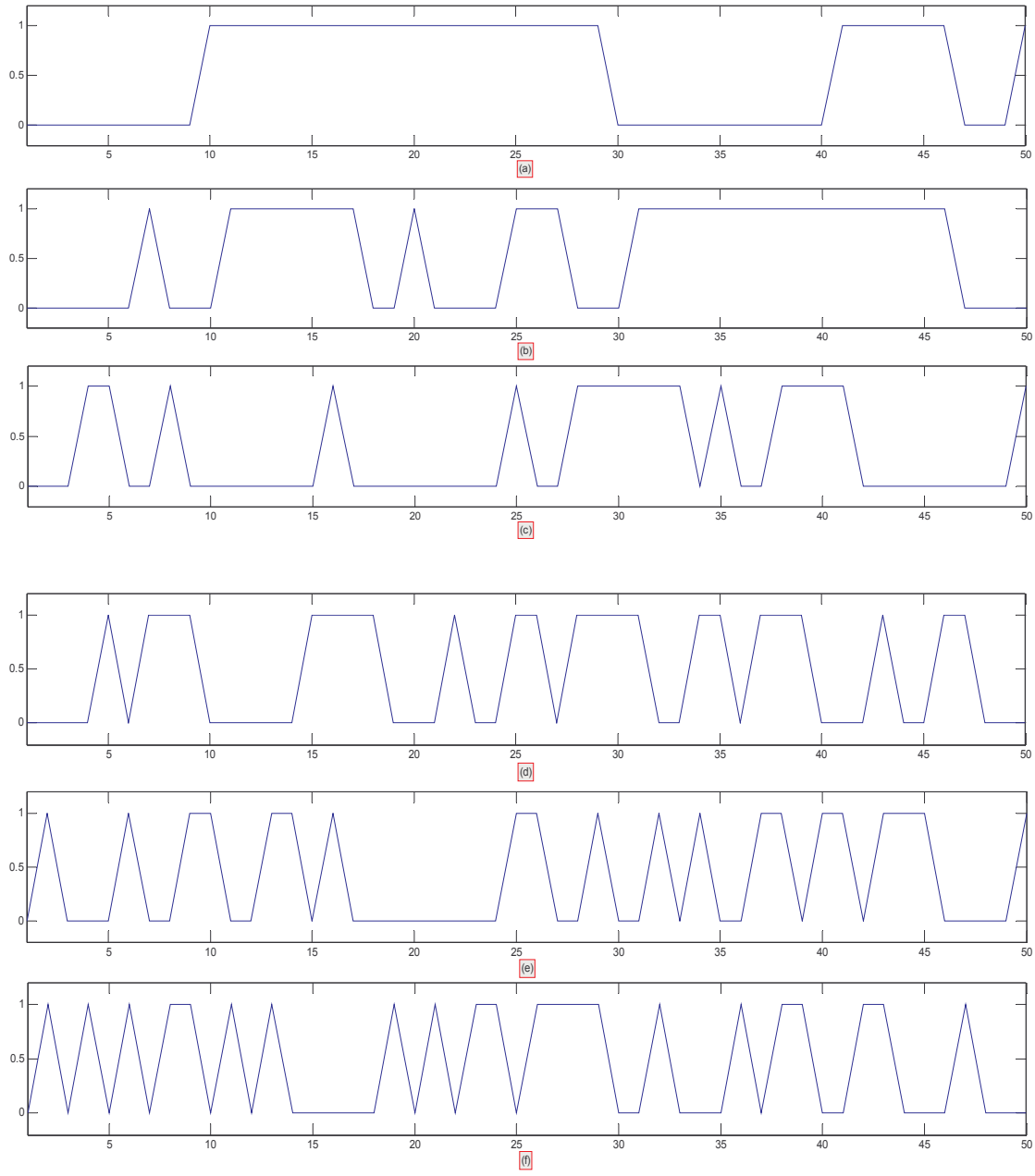
Fin-(pour)

Fin

Exemple 1.4:

Soit $\Omega = \{0, 1\}$ et $Q = \begin{pmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{pmatrix}$. La taille de l'échantillon est fixée à $T = 50$. Nous donnons quelques exemples de simulation de chaînes de Markov stationnaires (la probabilité initiale est ainsi toujours une probabilité stationnaire pour la chaîne).

→ Le nombre de trajectoires possibles pour chaque valeur prise par ρ est K^T. Nous montrons à la Figure 1.2 des exemples de trajectoires dans l'ordre croissant pour les valeurs prisent par $\rho \in \Xi = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.


1
0.5
0
5
10
15
20
25
30
35
40
45
50
(a)
(b)
(c)
(d)
(e)
(f)

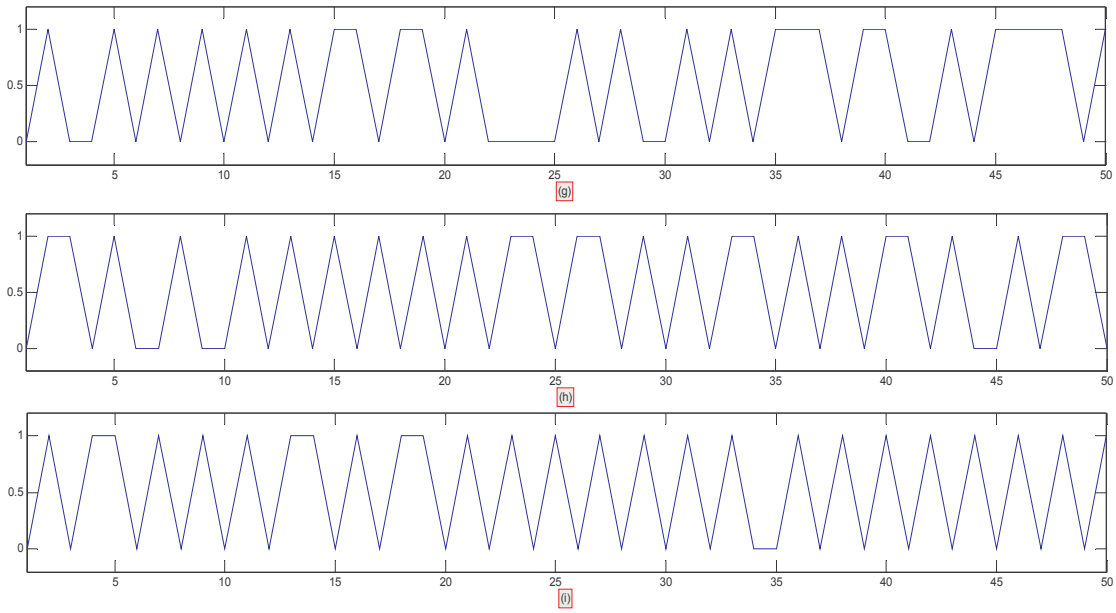
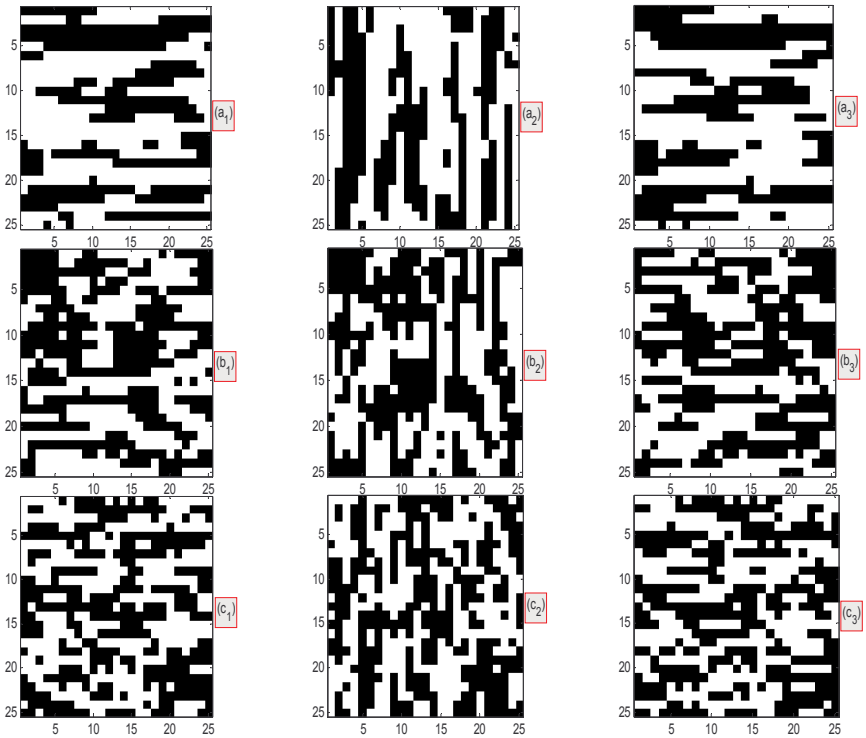


Figure 1.2 : Une simulation de trajectoire (a) pour $\rho = 0.1$, (b) pour $\rho = 0.2$, (c) pour $\rho = 0.3$, (d) pour $\rho = 0.4$, (e) pour $\rho = 0.5$, (f) pour $\rho = 0.6$, (g) pour $\rho = 0.7$, (h) pour $\rho = 0.8$ et (k) pour $\rho = 0.9$.

Nous donnons ci-dessous des images résultant des transformations sous forme matricielle des vecteurs simulés de taille $T = 625 = 25 \times 25$ (balayage de l'image horizontale, vertical, et en zigzag) pour des applications en imagerie et pour $\rho \in \Xi = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. En effet, diverses transformations de l'ensemble bidimensionnel des pixels en une suite monodimensionnelle permettent d'utiliser les différentes méthodes développées dans cette thèse en traitement d'images.



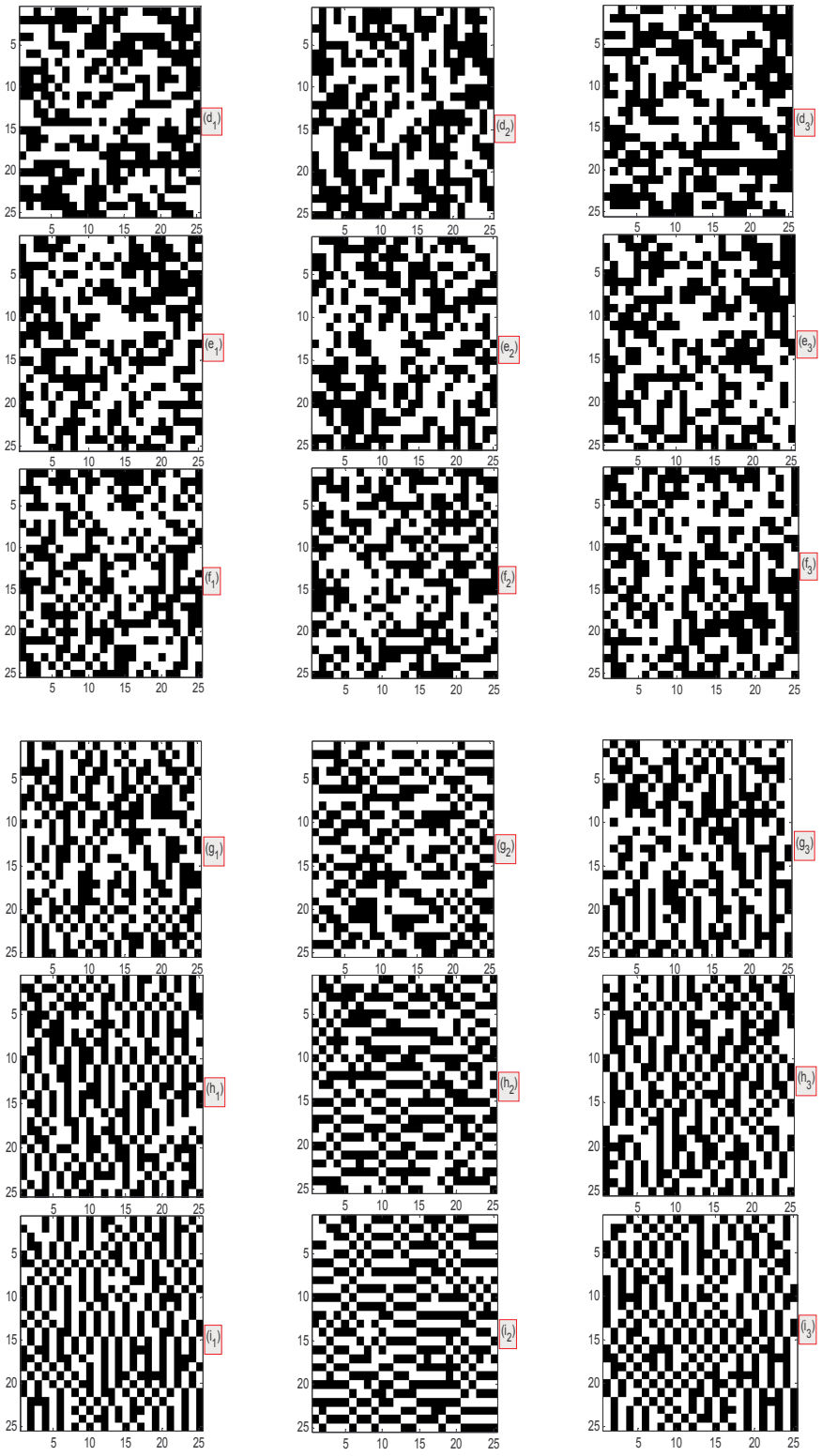


Figure 1.3 : Transformation des trajectoires simulées en images de taille (25x25) à $K=2$ niveaux de gris selon trois parcours différents : Le balayage horizontale donne les figures $((a_1),(b_1),(c_1),(d_1),(e_1),(f_1),(g_1),(h_1),(k_1))$, le balayage verticale donne les figures $((a_2),(b_2),(c_2),(d_2),(e_2),(f_2),(g_2),(h_2),(k_2))$ et pour le balayage en zigzague $((a_3),(b_3),(c_3),(d_3),(e_3),(f_3),(g_3),(h_3),(k_3))$.

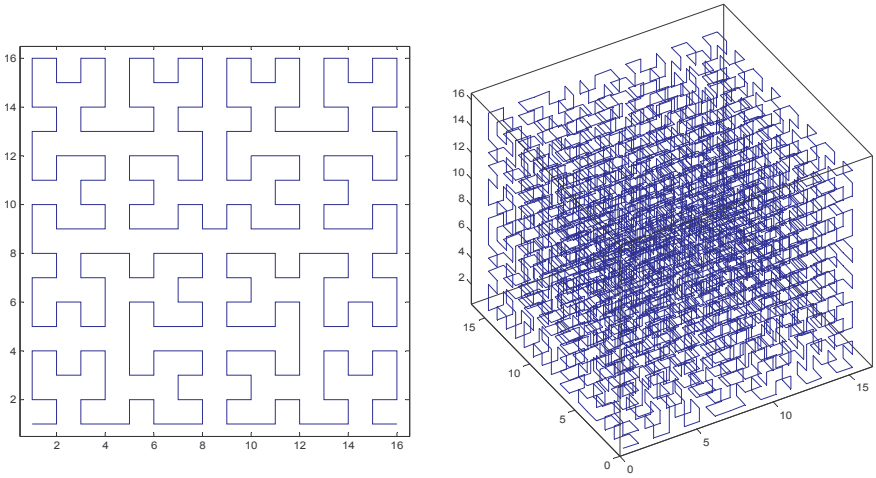


Figure 1.4 : La courbe de gauche représente le parcours de Hilbert-Peano en dimension 2 passant par 256 points de la surface du carré de taille (16x16). La courbe de droite est un parcours du type Hilbert-Peano en dimension 3 passant par 4096 points du volume du cube de taille (16x16x16).

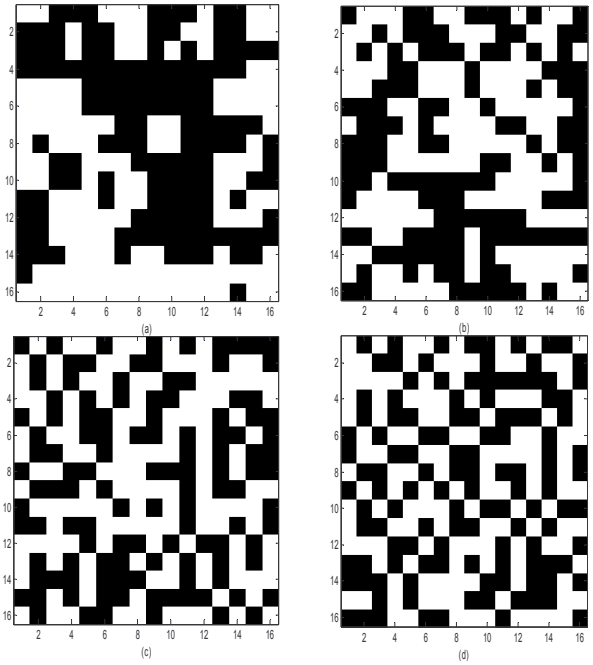


Figure 1.5 : Transformation des trajectoires simulées en images en dimension 2 de taille (16x16) à deux niveaux de gris selon le parcours de Hilbert-Peano : (a) pour $\rho = 0.2$, (b) pour $\rho = 0.4$, (c) pour $\rho = 0.6$ et (d) pour $\rho = 0.8$.

Transformation en 2D et 3D des vecteurs simulés en utilisant le parcours continu du type Hilbert-Peano pour des applications en imagerie ($T = T_{2\times d} = 256 = 16 \times 16$ en dimension 2 et $T = T_{3\times d} = 4096 = 16 \times 16 \times 16$ en dimension 3) sont présentées à la Figure 1.4.

On représente à la Figure 1.6. la transformation suivant le parcours de Peano de quatre vecteurs simulés correspondants à $\rho \in \{0.2, 0.4, 0.6, 0.8\}$.

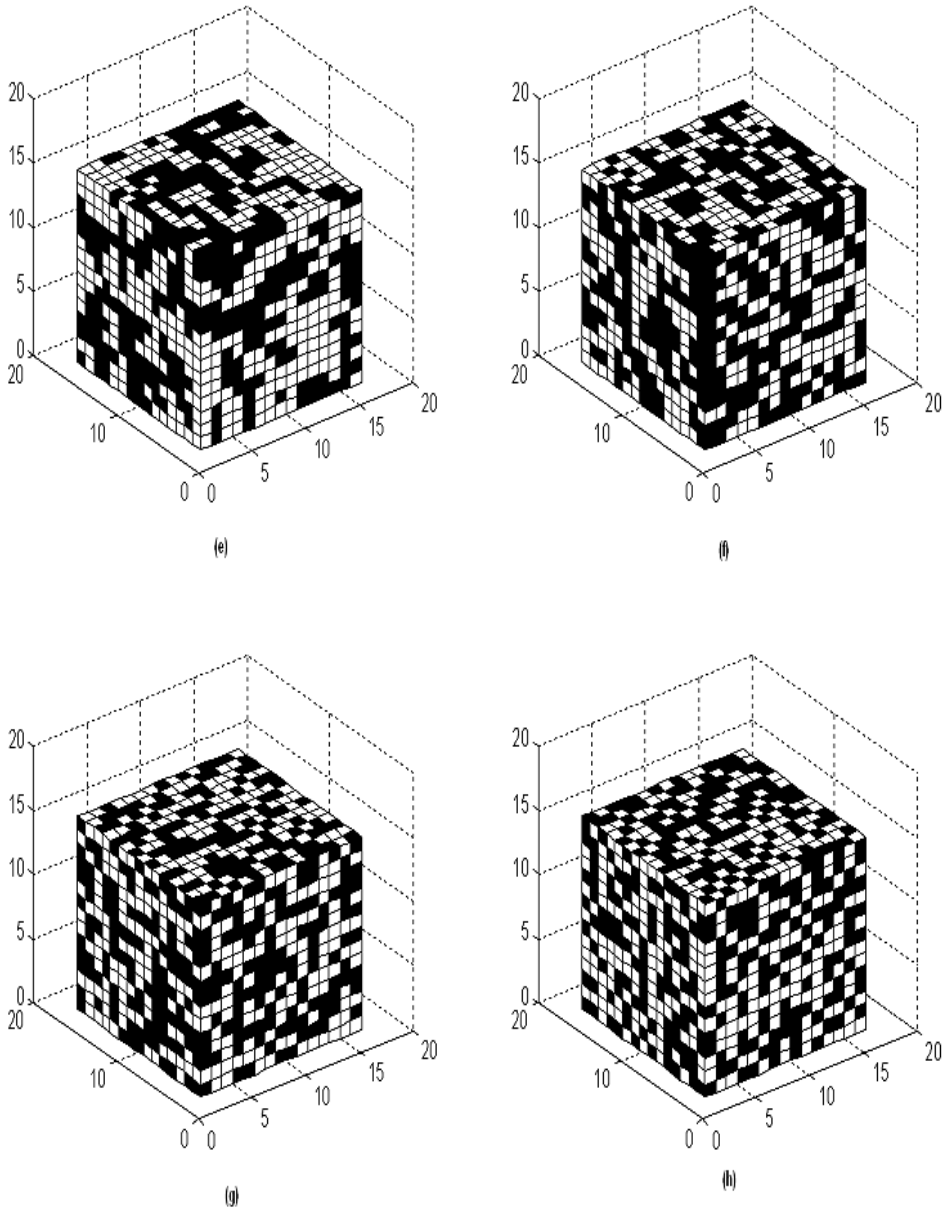


Figure 1.6 : Transformation des trajectoires simulées en images en dimension 3 de taille (16x16x16) à deux niveaux de gris selon le parcours de Hilbert-Peano : (f) pour $\rho = 0.2$, (g) pour $\rho = 0.4$, (h) pour $\rho = 0.6$ et (k) pour $\rho = 0.8$.

Conclusion

Nous avons rappelé dans ce chapitre différentes propriétés classiques des chaînes de Markov à espace d'états fini. Bien que le modèle soit parmi les plus simples permettant de prendre en compte la dépendance temporelle, nous constatons déjà sa richesse. Comme on va le voir au chapitre 2, dans le cas des chaînes de Markov cachées, la markovianité permet de calculer aussi bien les lois marginales a posteriori des variables que de connaître la réalisation de probabilité maximale de l'ensemble des variables. L'hypothèse de Markovianité sera alors cruciale. Par ailleurs, nous avons présenté quelques méthodes simples de simulation des variables, qui seront utilisées dans la suite.

Chapitre 2

Chaînes de Markov cachées

Les modèles de Markov cachés (MMC) jouissent d'une grande notoriété liée à leur efficacité, en particulier, dans le traitement du problème de la segmentation d'image, ou encore pour leur application dans le domaine de la reconnaissance de la parole. D'autres domaines comme la génétique, la finance, l'économie, l'informatique, la météorologie, la biologie … sont également concernés. Le modèle le plus simple est le modèle de chaîne de Markov caché (CMC). Dans un tel modèle les données cachées, qui représentent par exemple l'image segmentée (recherchée), sont considérées comme la réalisation d'une chaîne de Markov alors que le processus observé est une fonction composée du processus caché et d'un bruit aléatoire (stochastique). Un certain nombre de propriétés propres aux chaînes peuvent être étendues naturellement aux champs et aux arbres de Markov cachés.
Dans ce paragraphe on s'intéresse au problème de filtrage et de lissage dans le cas des données cachées discrètes, mais aussi aux algorithmes d'estimation de paramètre avec pour objectif des traitements non supervisés. Nous abordons deux types de modèles. Les CMC classiques, où la loi de l'observation conditionnelle au processus caché est très simple (variables indépendantes), et les CMC par du bruit « à mémoire longue », qui sont des modèles récents proposés dans [40] [46].
Ces derniers modèles seront utilisés dans le chapitre 4 pour approcher le filtre statistique optimal dans les systèmes linéaires à sauts.

2.1 Introduction

Soit $x = (x_1, x_2, ..., x_T)$ des états cachés qu'on cherche à déterminer à partir de données observées $y = (y_1, y_2, ..., y_T)$. Une approche probabiliste consiste à considérer le couple (x, y) comme une réalisation d'un processus couple donné par $(X, Y) = (X_1, X_2, ..., X_T, Y_1, Y_2, ..., Y_T)$. En absence de liens déterministes entre les données cachées et celles observées, ou dans le cas où ces liens sont d'une grande complexité, les méthodes de recherches stochastiques peuvent s'avérer efficaces.

Dans toute la suite de ce chapitre on considère que X_t prend ses valeurs dans l'ensemble $\Omega = \{\lambda_1, \lambda_2, ..., \lambda_K\}$ (Ω est l'espace d'état), et chaque Y_t prend ses valeurs dans $\Re$. La relation $p(x, y) = p(x)p(y \mid x)$, où $p(x)$ est une loi sur Ω^T et $p(y \mid x)$ sont des densités de probabilité sur $\Re^T$ par rapport à la mesure de Lebesgue, nous permet de constater la difficulté à la quelle on est confronté si on souhaite calculer la loi jointe dans toute sa généralité. En effet, le nombre de trajectoires possibles pour le processus caché jusqu'à

l'horizon fixe T est de K^T, qui croît rapidement avec T et devient impossible à manipuler. Pour faire face à ce problème d'ordre pratique il s'avère nécessaire de se restreindre à des lois $p(x,y)$ particulières. Dans la suite de ce chapitre on va décrire deux exemples : les chaînes de Markov cachées avec bruit indépendant (CMC-BI), et les chaînes de Markov cachées avec bruit à mémoire longue (CMC-BML). Par ailleurs, on va considérer que les bruits suivent des lois gaussiennes.

Notons que la loi jointe du couple $Z=(X,Y)$ peut être donnée en toute généralité sous la forme factorisée suivante:

$$p(z) = p(z_T \mid z_1^{T-1}) p(z_1^{T-1}) = p(z_1) \prod_{t=2}^{T} p(z_t \mid z_1^{t-1}) \qquad (2.1)$$

Cette loi s'exprime ainsi par la loi de probabilité initiale $p(z_1)$ et des transitions $\left\{ p(z_t \mid z_1^{t-1}) \right\}_{t=2}^{T}$.

De manière similaire, la loi de la chaîne X conditionnelle à $Y=y$, dite « loi a posteriori », peut être écrite sous une forme factorisée suivante:

$$p(x \mid y) = p(x_T \mid x_1^{T-1}, y) p(x_1^{T-1} \mid y) = p(x_1 \mid y) \prod_{t=2}^{T} p(x_t \mid x_1^{t-1}, y) \qquad (2.2)$$

Cette loi dépend de la marginale a posteriori initiale $p(x_1 \mid y)$ et des lois de transitions a posteriori $\left\{ p(x_t \mid x_1^{t-1}, y) \right\}_{t=2}^{T}$.

Enfin, le calcul de la loi marginale a posteriori $p(x_t \mid y)$ peut se faire en deux étapes:

- Formule récursive pour le calcul de la loi jointe a posteriori de l'origine jusqu'à l'instant t.

$$p(x_1^t \mid y) = p(x_t \mid x_1^{t-1}, y)\, p(x_1^{t-1} \mid y)$$

- Marginalisation

$$p(x_{t+1} \mid y) = \sum_{x_1^t \in \Omega^t} p(x_{t+1} \mid x_1^t, y)\, p(x_1^t \mid y)$$

La loi marginale dépend également des lois de transition $\left\{ p(x_t \mid x_1^{t-1}, y) \right\}_{t=2}^{T}$. Ainsi que précisé ci-dessus ces calculs deviennent rapidement impossible à effectuer lorsque T croît. Dans le paragraphe suivant on va exprimer ces quantités par les probabilités dites « rétrogrades » et « progressives », dont on va montrer l'intérêt dans les modèles particuliers CMC-BI et CMC-BML.

2.2 Probabilités rétrograde et progressive

2.2.1 Probabilités rétrogrades

On appelle probabilités « rétrogrades » (ou coefficients « Backward ») les quantités définies par:

$$\forall t \in \{1,...,T-1\} \qquad \beta_t(z_1^t) = p(y_{t+1}^T \,|\, z_1^t) \tag{2.3}$$

Le calcul de ces probabilités peut s'effectuer récursivement en passe arrière. Nous avons:

$$\begin{aligned} \forall t \in \{1,...,T-1\} \qquad \beta_t(z_1^t) = p(y_{t+1}^T \,|\, z_1^t) &= \sum_{x_{t+1}\in\Omega} p(y_{t+2}^T, x_{t+1}, y_{t+1} \,|\, z_1^t) \\ &= \sum_{x_{t+1}\in\Omega} p(y_{t+2}^T \,|\, z_1^{t+1}) p(z_{t+1} \,|\, z_1^t) \\ &= \sum_{x_{t+1}\in\Omega} p(z_{t+1} \,|\, z_1^t)\, \beta_{t+1}(z_1^{t+1}) \end{aligned} \tag{2.4}$$

On remarque que:

$$\beta_{T-1}(z_1^{T-1}) = p(y_T \,|\, z_1^{T-1}) = \sum_{x_T\in\Omega} p(z_T \,|\, z_1^{T-1}) \tag{2.5}$$

La formule de récurrence est valable pour $t = T$ si on pose $\beta_T(z_1^T) = 1$.

L'introduction des probabilités rétrogrades nous permet de calculer les probabilités de transitions $\left\{ p(x_t \,|\, x_1^{t-1}, y) \right\}_{t=2}^T$ en fonction des probabilités de transitions du processus Z, c'est-à-dire les quantités $\left\{ p(z_t \,|\, z_1^{t-1}) \right\}_{t=2}^T$. On a:

$$\begin{aligned} p(x_1 \,|\, y) = \frac{p(x_1, y)}{\sum_{x_1\in\Omega} p(x_1, y)} &= \frac{p(y_2^T | x_1, y_1)\, p(x_1, y_1)}{\sum_{x_1\in\Omega} p(y_2^T | x_1, y_1)\, p(x_1, y_1)} \\ &= \frac{p(z_1)\, \beta_1(z_1)}{\sum_{x_1\in\Omega} p(z_1)\, \beta_1(z_1)} \end{aligned}, \tag{2.6}$$

et pour tout $t \in \{1,...,T-1\}$:

$$p(x_{t+1}\,|\,x_1^t, y) = \frac{p(x_{t+1}, y_{t+1}^T\,|\,x_1^t, y_1^t)}{\sum_{x_{t+1}\in\Omega} p(x_{t+1}, y_{t+1}^T\,|\,x_1^t, y_1^t)} = \frac{p(y_{t+2}^T, z_{t+1}\,|\,z_1^t)}{\sum_{x_{t+1}\in\Omega} p(y_{t+2}^T, z_{t+1}\,|\,z_1^t)}$$

$$= \frac{p(z_{t+1}\,|\,z_1^t)\,\beta_{t+1}(z_1^{t+1})}{\sum_{x_{t+1}\in\Omega} p(z_{t+1}\,|\,z_1^t)\,\beta_{t+1}(z_1^{t+1})} = p(z_{t+1}\,|\,z_1^t)\frac{\beta_{t+1}(z_1^{t+1})}{\beta_t(z_1^t)} \tag{2.7}$$

Alors:

$$p(x\,|\,y) = \prod_{t=1}^{T} p(z_t\,|\,z_0^{t-1})\frac{\beta_t(z_0^t)}{\beta_{t-1}(z_0^{t-1})}, \tag{2.8}$$

avec $\beta_0(z_0) = \beta_0 = p(y) = \sum_{x_1\in\Omega} p(z_1)\beta_1(z_1)$, $\beta_T(z_0^T) = \beta_T(z_1^T) = 1$, et $p(z_1\,|\,z_0) = p(z_1)$, $p(z_{t+1}\,|\,z_0^t) = p(z_{t+1}\,|\,z_1^t)$ pour $t \geq 1$.

2.2.2 Probabilités progressives

On appelle probabilités « progressives » (ou coefficients « Forward) » les quantités définies par:

$$\forall t \in \{1,...,T\} \qquad \alpha_t(x_1^t) = p(z_1^t) = p(x_1^t, y_1^t) \tag{2.9}$$

Le calcul de ces probabilités peut être effectué récursivement en passe avant. Nous avons:

$$\alpha_1(x_1) = p(z_1) = p(x_1, y_1) = p(x_1)\,p(y_1\,|\,x_1), \tag{2.10}$$

et pour tout $t \in \{1,...,T-1\}$:

$$\begin{aligned}\alpha_{t+1}(x_1^{t+1}) = p(x_1^{t+1}, y_1^{t+1}) &= p(x_{t+1}, y_{t+1}\,|\,x_1^t, y_1^t)\,p(x_1^t, y_1^t)\\ &= p(z_{t+1}\,|\,z_1^t)\,\alpha_t(x_1^t)\end{aligned} \tag{2.11}$$

On a alors pour tout $t \in \{1,...,T\}$:

$$p(x_1^t\,|\,y) = \sum_{x_{t+1}^T\in\Omega^{T-t}} p(x\,|\,y) \tag{2.12}$$

La formule (2.8) ne permet pas un calcul récursif des probabilités marginales, pour le permettre on va réécrire la probabilité jointe a posteriori sous la forme suivante. Pour tout $t \in \{1,...,T\}$:

$$p(x_1^t, y) = p(y_{t+1}^T\,|\,x_1^t, y_1^t)\,p(x_1^t, y_1^t) = \beta(z_1^t)\,\alpha_t(x_1^t). \tag{2.13}$$

D'autre part:

$$p(x,y) = p(x_{t+1}^{T} \mid x_1^t, y)\; p(x_1^t, y) = p(x_{t+1}^{T} \mid x_1^t, y)\; \beta(z_1^t)\; \alpha_t(x_1^t), \tag{2.14}$$

et

$$p(y) = \sum_{x \in \Omega^T} p(x,y) \;= \sum_{x_1^t \in \Omega^t} \beta_t(z_1^t)\; \alpha_t(x_1^t) \tag{2.15}$$

La probabilité a posteriori est alors donnée par:

$$\forall t \in \{1,...,T-1\}, \quad p(x \mid y) = \frac{\beta_t(z_1^t)\; \alpha_t(x_1^t)}{\sum_{x_1^t \in \Omega^t} \beta_t(z_1^t)\; \alpha_t(x_1^t)}\, p(x_{t+1}^{T} \mid x_1^t, y), \tag{2.16}$$

Les marginales a posteriori sont donc données par:

$$\forall t \in \{1,...,T\},\; p(x_t \mid y) = \sum_{x_1^{t-1} \in \Omega^{t-1}} \; \frac{\beta_t(z_1^t)\; \alpha_t(x_1^t)}{\sum_{x_1^t \in \Omega^t} \beta_t(z_1^t)\; \alpha_t(x_1^t)} \tag{2.17}$$

Remarque 2.1

Le calcul de la loi jointe a posteriori ou marginale a posteriori dépend des lois de transition du processus couple. Dans les formules (2.16) et (2.17) les deux densités d'intérêt sont calculables en fonction des coefficients progressifs et rétrogrades qui eux-mêmes sont calculés récursivement en fonction des transitions $\left\{ p(z_t \mid z_1^{t-1}) \right\}_{t=2}^{T}$.

2.3 Chaîne de Markov cachée à bruit indépendant

Définition 2.1

Le couple de processus aléatoires $(X,Y) = (X_t, Y_t)_{t \in \aleph^*}$ sera appelé une « chaîne de Markov cachée à bruit indépendant » (CMC-BI) si $(X_t)_{t \in \aleph^*}$ est une chaîne de Markov à espace d'état Ω fini et les variables aléatoires Y_t sont à valeur dans $\Re$, vérifiant les deux propriétés suivantes:

(H1) $\quad \forall T \in \aleph^* \quad p(y_1^T \mid x_1^T) = \prod_{t=1}^{T} p(y_t \mid x_1^T)$;

(H2) $\quad \forall T \in \aleph^* \quad \forall t \in \{1,...,T\} \quad p(y_t \mid x_1^T) = p(y_t \mid x_t)$.

2.3.1 Inférence Bayésienne

Dans le cas des chaînes de Markov cachées à bruit indépendant le processus X est donc une chaîne de Markov. D'autre part, on montre classiquement en utilisant les hypothèses (H1) et (H2) que le processus couple Z est également un processus de Markov. Nous avons pour tout $t \in \{1,...,T-1\}$:

$$\begin{aligned} p(z_{t+1} \mid z_1^t) = p(x_{t+1}, y_{t+1} \mid x_1^t, y_1^t) &= p(y_{t+1} \mid x_1^{t+1}, y_1^t)\, p(x_{t+1} \mid x_1^t, y_1^t) \\ &= p(y_{t+1} \mid x_{t+1})\, p(x_{t+1} \mid x_t) \\ &= p(z_{t+1} \mid z_t) \end{aligned} \tag{2.18}$$

En partant de la forme générale des probabilités a posteriori vue au paragraphe précédent, on peut en déduire leurs formes dans le cas particulier des chaînes de Markov cachées à bruit indépendant. Le processus couple étant un processus de Markov sa loi pour un horizon fixe $T > 1$ se factorise de la manière suivante:

$$p(z) = p(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \tag{2.19}$$

Les probabilités rétrogrades prennent alors la forme suivante :

$$\forall t \in \{1,...,T-1\} \qquad \beta_t(z_1^t) = p(y_{t+1}^T \mid z_1^t) = p(y_{t+1}^T \mid x_t) = \beta_t(x_t), \tag{2.20}$$

On a:

$$\forall t \in \{1,...,T-1\} \qquad \beta_t(x_t) = \sum_{x_{t+1} \in \Omega} p(x_{t+1} \mid x_t) p(y_{t+1} \mid x_{t+1})\ \beta_{t+1}(x_{t+1}) \tag{2.21}$$

Concernant les probabilités progressives, nous avons :

$$\forall t \in \{1,...,T-1\} \qquad \alpha_{t+1}(x_1^{t+1}) = p(x_{t+1} \mid x_t)\, p(y_{t+1} \mid x_{t+1})\, \alpha_t(x_1^t) \tag{2.22}$$

Posons:

$$\forall t \in \{1,...,T-1\} \qquad \alpha_{t+1}(x_{t+1}) = \sum_{x_1^t \in \Omega^t} \alpha_{t+1}(x_1^{t+1}) \tag{2.23}$$

Alors:

$$\forall t \in \{1,...,T-1\}, \qquad \alpha_{t+1}(x_{t+1}) = p(y_{t+1} \mid x_{t+1}) \sum_{x_t \in \Omega} p(x_{t+1} \mid x_t)\, \alpha_t(x_t) \tag{2.24}$$

On déduit de (2.16) la loi marginale a posteriori :

$$\forall t \in \{1,...,T\}, \qquad p(x_t \mid y) = \frac{\beta_t(x_t)\,\alpha_t(x_t)}{\sum_{x_t \in \Omega} \beta_t(x_t)\,\alpha_t(x_t)} \tag{2.25}$$

Introduit pour la première fois par Baum et Petri en 1966 [70] le modèle de chaîne de Markov cachée à bruit indépendant est très utilisé en traitement de signaux monodimensionnels. En 1970 ces mêmes chercheurs en compagnie de Soules et Weiss [71] ont proposé une procédure récursive dite « progressive-rétrograde » (Forward-Backward) pour le calcul des probabilités marginales a posteriori. Dans le cas où les transitions de la chaîne X et les lois $\left(p(y_t \mid x_t)\right)_{t \in \{1,...,T\}}$ sont données, les lois marginales a posteriori sont ainsi calculables via la procédure décrite plus haut. Il en résulte des possibilités de restauration bayésienne abordée ci-après.

L'algorithme qui va suivre correspond à celui de Baum-Welsh conditionnel ; proposé par Devijver en 1985, son principal atout est d'assurer la stabilité numérique lors du calcul des probabilités rétrogrades et progressives [72]. En effet, lorsque T devient grand ces dernières peuvent tendre vers 0 et poser ainsi un problème d'ordre numérique..

On pose:

$$\widetilde{\alpha}_{t+1}(x_{t+1}) = p(x_{t+1} \mid y_1^{t+1}) = \frac{\alpha_{t+1}(x_{t+1})}{\sum_{x_{t+1} \in \Omega} \alpha_{t+1}(x_{t+1})}$$

$$= \frac{p(y_{t+1} \mid x_{t+1}) \sum_{x_t \in \Omega} p(x_{t+1} \mid x_t)\, p(x_t \mid y_1^t)}{\sum_{x_{t+1} \in \Omega} p(y_{t+1} \mid x_{t+1}) \sum_{x_t \in \Omega} p(x_{t+1} \mid x_t)\, p(x_t \mid y_1^t)} \tag{2.26}$$

$$= \frac{p(y_{t+1} \mid x_{t+1}) \sum_{x_t \in \Omega} p(x_{t+1} \mid x_t)\, \widetilde{\alpha}_t(x_t)}{\sum_{x_{t+1} \in \Omega} p(y_{t+1} \mid x_{t+1}) \sum_{x_t \in \Omega} p(x_{t+1} \mid x_t)\, \widetilde{\alpha}_t(x_t)},$$

et

$$\widetilde{\beta}_t(x_t) = \frac{p(y_{t+1}^T \mid x_t)}{p(y_{t+1}^T \mid y_1^t)} = \frac{p(y_{t+1}^T \mid x_t)}{\sum_{x_t \in \Omega} p(y_{t+1}^T \mid x_t)\, p(x_t \mid y_1^t)} = \frac{\beta_t(x_t)}{\sum_{x_t \in \Omega} \beta_t(x_t)\, \widetilde{\alpha}_t(x_t)}$$

$$= \frac{\sum_{x_{t+1} \in \Omega} p(y_{t+2}^T \mid x_{t+1}) p(y_{t+1} \mid x_{t+1}) p(x_{t+1} \mid x_t)}{p(y_{t+2}^T \mid y_1^{t+1}) p(y_{t+1} \mid y_1^t)}$$

(2.27)

$$= \frac{1}{p(y_{t+1}\,|\, y_1^t)} \sum_{x_{t+1}\in\Omega} p(y_{t+1}\,|\, x_{t+1}) p(x_{t+1}\,|\, x_t) \widetilde{\beta}_{t+1}(x_{t+1})$$

$$= \frac{\sum_{x_{t+1}\in\Omega} p(y_{t+1}\,|\, x_{t+1})\, p(x_{t+1}\,|\, x_t)\, \widetilde{\beta}_{t+1}(x_{t+1})}{\sum_{x_{t+1}\in\Omega} p(y_{t+1}\,|\, x_{t+1}) \sum_{x_t\in\Omega} p(x_{t+1}\,|\, x_t)\, \widetilde{\alpha}_t(x_t)}$$

On a alors:

$$p(x_t\,|\, y) = \widetilde{\alpha}_t(x_t)\, \widetilde{\beta}_t(x_t) \tag{2.28}$$

Nous présentons ci-après l'algorithme de calcul des marginales a posteriori dans une CMC-BI classique [49]. Afin de simplifier l'écriture nous adoptons les notations suivantes. Soit $\Omega = \{\lambda_1,...,\lambda_K\}$ l'espace d'état de à la chaîne X. Alors pour tout $(k,l) \in \{1,...,K\}^2$, on notera:

$$p_1(k) = P(X_1 = \lambda_k)\,, \tag{2.29}$$

$$a_t(k,l) = P(X_{t+1} = \lambda_l\,|\, X_t = \lambda_k)\,, f_k(y,t) = P(Y_t = y\,|\, X_t = \lambda_k) \tag{2.30}$$

$$\alpha_t(\lambda_k) = \alpha_t(k),\ \ \beta_t(\lambda_k) = \beta_t(k),\ \ \gamma_t(k) = P(X_t = \lambda_k\,|\, Y = y) \tag{2.31}$$

Algorithme 2.1: Calcul des marginales a posteriori

Procédure Forward *(filtrage)* :

- $t = 1$ on calcule $\overline{\alpha}_1 = \sum_{i=1}^{K} f_i(y_1,1)\, p_1(i)$;

  Pour tout $k \in \{1,...,K\}$ $\widetilde{\alpha}_1(k) = \frac{1}{\overline{\alpha}_1} f_k(y_1,1)\, p_1(k)$

- $1 < t \le T$ on calcul $\overline{\alpha}_t = \sum_{j=1}^{K} f_j(y_t,t) \sum_{i=1}^{K} a_{t-1}(i,j)\, \widetilde{\alpha}_{t-1}(i)$

  Pour tout $l \in \{1,...,K\}$ $\widetilde{\alpha}_t(l) = \frac{1}{\overline{\alpha}_t} f_l(y_t,t) \sum_{k=1}^{K} a_{t-1}(k,l)\, \widetilde{\alpha}_{t-1}(k)$

Procédure Backward

- $t = T$ pour tout $l \in \{1,...,K\}$ $\widetilde{\beta}_T(l) = 1$

- $t < T$ pour tout $k \in \{1,...,K\}$ $\quad \widetilde{\beta}_t(k) = \dfrac{1}{\overline{\alpha}_{t+1}} \sum_{l=1}^{K} a_t(k,l) f_l(y_{t+1}, t+1)\, \widetilde{\beta}_{t+1}(l)$

Marginale a posteriori (Lissage)

- $1 \le t \le T$ pour tout $k \in \{1,...,K\}$ $\quad \gamma_t(k) = \widetilde{\alpha}_t(k)\, \widetilde{\beta}_t(k)$

Fin

2.3.2 Estimation Bayésienne de la trajectoire cachée

On s'intéresse à deux types d'estimateur : l'estimateur MAP *(Maximum A Posteriori)* et l'estimateur MPM *(Maximum de Marginales a Posteriori)*.

Pour le second estimateur (MPM), que l'on utilisera dans les simulations, l'*Algorithme (2.1)* permet donc un calcul récursif des marginales a posteriori. On pose alors :

$$\hat{x}_t = \arg \underset{x_t \in \Omega}{Max} \left[p(x_t \mid y) \right] = \arg \underset{x_t \in \Omega}{Max} \left[\widetilde{\alpha}_t(x_t)\, \widetilde{\beta}_t(x_t) \right] \tag{2.32}$$

Comme son nom l'indique l'estimateur du MAP est obtenu en maximisant la loi de probabilité a posteriori :

$$\hat{x}_{MAP} = \arg \underset{x \in \Omega^T}{Max} \left[p(x \mid y) \right] \tag{2.33}$$

La solution du MAP est également calculable, avec une complexité raisonnable, dans les CMC-BI : on peut utiliser l'algorithme de Viterbi [31] dont le déroulement est le suivant.

Commençons par le calcul de manière récursive de la probabilité du chemin le plus probable allant d'un instant t jusqu'à l'horizon fixé T . Nous avons

$$\forall t \in \{2,...,T-1\} \qquad p(z_t^T \mid z_1^{t-1}) = p(z_t \mid z_1^{t-1})\, p(z_{t+1}^T \mid z_1^t) \tag{2.34}$$

On pose:

$$\forall t \in \{1,...,T-1\} \qquad \delta_t(z_1^t) = \max_{x_{t+1}^T \in \Omega^{T-t}} [p(z_{t+1}^T \mid z_1^t)] \tag{2.35}$$

Alors:

$$\forall t \in \{2,...,T\} \qquad \delta_{t-1}(z_1^{t-1}) = \max_{x_t \in \Omega} [p(z_t \mid z_1^{t-1})\, \delta_t(z_1^t)] \tag{2.36}$$

Dans le cas des chaînes de Markov cachées à bruit indépendant on a:

$$p(z_t \mid z_1^{t-1}) = p(z_t \mid x_{t-1}) = p(y_t \mid x_t)\, p(x_t \mid x_{t-1}) \tag{2.37}$$

$$p(z_t^T \mid z_1^{t-1}) = p(z_t^T \mid x_{t-1}) = p(z_t \mid x_{t-1})\, p(z_{t+1}^T \mid x_t)\,. \tag{2.38}$$

Donc

$$\forall t \in \{1,...,T-1\} \qquad \delta_t(z_1^t) = \delta_t(x_t) = \max_{x_{t+1}^T \in \Omega^{T-t}} [p(z_{t+1}^T \mid x_t)]\ , \tag{2.39}$$

ce qui donne

$$\forall t \in \{2,...,T\},\ \delta_{t-1}(z_1^{t-1}) = \delta_{t-1}(x_{t-1}) = \max_{x_t \in \Omega} [p(y_t \mid x_t)\, p(x_t \mid x_{t-1})\ \delta_t(x_t)] \tag{2.40}$$

On stocke alors les valeurs d'état réalisant les maxima dans ce sens rétrograde.

$$\psi_t(x_{t-1}) = \arg\left[\max_{x_t \in \Omega} [p(y_t \mid x_t)\, p(x_t \mid x_{t-1})\ \delta_t(x_t)]\right] \tag{2.41}$$

Ensuite une étape en passe avant est suffisante pour récupérer les composantes du MAP.

$$\hat{x}_1 = \psi_1 = \arg\left[\max_{x_1 \in \Omega} [p(y_1 \mid x_1)\, p(x_1)\ \delta_1(x_1)]\right] \tag{2.42}$$

$$\hat{x}_t = \psi_t(\hat{x}_{t-1}) = \arg\left[\max_{x_t \in \Omega} [p(y_t \mid x_t)\, p(x_t \mid \hat{x}_{t-1})\ \delta_t(x_t)]\right] \tag{2.43}$$

On va résumer ces deux étapes rétrograde et progressive dans l'algorithme qui suit.

Algorithme 2.2: Calcul de l'estimateur MAP (algorithme de Viterbi)

Initialisation

- $t = T$ on calcule

$$\delta_{T-1}(x_{T-1}) = \underset{x_T \in \Omega}{Max}\left[p(y_T \mid x_T)\, p(x_T \mid x_{T-1})\right]$$

$$\psi_T(x_{T-1}) = \arg \underset{x_T \in \Omega}{Max}\left[p(y_T \mid x_T)\, p(x_T \mid x_{T-1})\right]$$

Procédure rétrograde

- $t = (T-1)...1$ on calcul

$$\psi_t(x_{t-1}) = \arg \underset{x_t \in \Omega}{Max}\left[p(y_t \mid x_t)\, p(x_t \mid x_{t-1})\, \delta_t(x_t)\right]$$

$$\delta_{t-1}(x_{t-1}) = \underset{x_t \in \Omega}{Max}\left[p(y_t \mid x_t)\, p(x_t \mid x_{t-1})\, \delta_t(x_t)\right]$$

Procédure progressive

- $t = 1$ Initialisation au sens direct $\hat{x}_1 = \arg\left[\max_{x_1 \in \Omega} [p(y_1 | x_1)\, p(x_1)\ \delta_1(x_1)] \right]$
- $1 < t \leq T$ on calcul $\hat{x}_t = \psi_t(\hat{x}_{t-1})$

Fin

Conclusion

Le modèle de chaîne de Markov cachée à bruit indépendant est pratique au niveau du calcul des lois nécessaires pour des estimations bayésiennes des états cachés, qui peut être fait avec une complexité linéaire en temps. Cependant, ce modèle est fondé sur des hypothèses fortes, particulièrement celle de l'indépendance des variables observées conditionnellement au processus caché. Pour ces raisons ce modèle a connu plusieurs généralisations selon l'application traitée. On peut citer les modèles semi-markoviens cachés (MSMC), les modèles hiérarchiques. Par ailleurs, il apparaît que seule la markovianité du processus caché conditionnellement au processus observé est indispensable pour pouvoir faire des traitements bayésiens, ce qui permet d'introduire les modèles markoviens « couples » [52] et « triplets » [51]. Une autre généralisation consiste à considérer les chaînes « partiellement » de Markov, qui sont markoviennes par rapport aux variables cachées mais ne le sont pas nécessairement par rapport aux variables observées [53]. Les chaînes de Markov cachées par du bruit à mémoire longue considérées ci-après font partie de ces deniers modèles. Ainsi que nous allons le voir, X sera de Markov mais le couple (X, Y) ne le sera pas [46] ; cependant, les traitements bayésiens restent possibles avec une complexité raisonnable.

2.4 Chaîne de Markov cachée avec bruit à mémoire longue

Dans ce paragraphe nous étudions le cas où le processus des observations est à dépendance longue conditionnellement au processus caché, ce dernier étant une chaîne de Markov. La corrélation au sein du processus peut modéliser des phénomènes fractals ou saisonniers comme cela peut être le cas dans des images naturelles ou des données financières. Introduits et étudiés récemment [40] [46], ces modèles font partie des modèles « partiellement de Markov » [53].

Définition 2.2 (Dépendance longue)

Soit $Y = (Y_t)_{t \in \mathbb{N}^*}$ un processus réel stationnaire du second ordre, de famille de covariance $(\gamma(n))_{n \in \mathbb{N}}$. On dira que Y est à dépendance longue s'il existe un $\upsilon \in]0,1]$ et une constante $C^{te} \in \Re$ tels que:

$$\lim_{n \to +\infty} n^{\upsilon} \gamma(n) = C^{te} \qquad (2.44)$$

Remarque 2.2

- La covariance d'un processus à mémoire longue satisfait la condition $\sum_{n\in\aleph}\gamma(n)=+\infty$;
- Lorsque $\gamma(n)\sim_{+\infty} C^{te}n^{-\upsilon}$ pour $\upsilon>1$, on dit que Y est à dépendance intermédiaire.

On va se restreindre dans la suite de ce mémoire au processus à mémoire longue dont la famille de covariance est donnée par

$$\gamma(n)=\sigma^2(n+1)^{-\upsilon},\ \forall n\in\aleph,\ \ \forall\upsilon\in\Re^+. \tag{2.45}$$

Définition 2.3

Le couple de processus aléatoires $(X,Y)=(X_t,Y_t)_{t\in\aleph^*}$ sera appelé « chaîne de Markov cachée à bruit à mémoire (ou dépendance, ou corrélation) longue » si :

(i) $(X_t)_{t\in\aleph^*}$ est une chaîne de Markov à espace d'état Ω fini ;

(ii) les variables aléatoires Y_t sont à valeurs dans $\Re$ et vérifient la propriété suivante:

$$\forall T\in\aleph^*\ \ \forall t\in\{1,...,T\}\quad p(y_t\,|\,x_1^T,y_1^{t-1})=p(y_t\,|\,x_t,y_1^{t-1}), \tag{2.46}$$

avec les lois conditionnelles $p(y_t\,|\,x_t,y_1^{t-1})$ définies, à x_t donnés, par des processus à mémoire longue.

2.4.1 Inférence Bayésienne

Dans le cas des chaînes de Markov cachées à dépendance longue le processus X est une chaîne de Markov; par contre le processus couple $Z=(X,Y)$ n'est pas nécessairement un processus de Markov. Cependant, les probabilités rétrogrades et les probabilités progressives sont calculables dans certaines situations. Nous avons :

$$\begin{aligned}\forall t\in\{1,...,T-1\},\quad p(z_{t+1}\,|\,z_1^t)&=p(x_{t+1},y_{t+1}\,|\,x_1^t,y_1^t)\\&=p(y_{t+1}\,|\,x_1^{t+1},y_1^t)\,p(x_{t+1}\,|\,x_1^t,y_1^t)\\&=p(y_{t+1}\,|\,x_{t+1},y_1^t)\,p(x_{t+1}\,|\,x_t)\end{aligned} \tag{2.47}$$

- Les probabilités rétrogrades s'écrivent:

$$\forall t\in\{1,...,T-1\}\qquad \beta_t(z_1^t)=p(y_{t+1}^T\,|\,z_1^t)=p(y_{t+1}^T\,|\,x_t,y_1^t)=\beta_t(x_t) \tag{2.48}$$

Nous avons donc:

$$\forall t\in\{1,...,T-1\}\qquad \beta_t(x_t)=\sum_{x_{t+1}\in\Omega}p(x_{t+1}\,|\,x_t)p(y_{t+1}\,|\,x_{t+1},y_1^t)\ \beta_{t+1}(x_{t+1}) \tag{2.49}$$

Nous voyons que ce calcul est possible si les transitions $p(y_{t+1}|\, x_{t+1}, y_1^t)$ sont manipulables. L'intérêt du modèle considéré est que dans le cas gaussien ces transitions sont effectivement calculables avec une complexité raisonnable.

- Les probabilités progressives s'écrivent:

$$\forall t \in \{1,...,T-1\} \qquad \alpha_{t+1}(x_1^{t+1}) = p(x_{t+1}|\, x_t)\; p(y_{t+1}|\, x_{t+1}, y_1^t)\; \alpha_t(x_1^t) \tag{2.50}$$

Posons:

$$\forall t \in \{1,...,T-1\} \qquad \alpha_{t+1}(x_{t+1}) = \sum_{x_1^t \in \Omega^t} \alpha_{t+1}(x_1^{t+1}) \tag{2.51}$$

Alors nous avons:

$$\forall t \in \{1,...,T-1\} \qquad \alpha_{t+1}(x_{t+1}) = p(y_{t+1}|\, x_{t+1}, y_1^t) \sum_{x_t \in \Omega} p(x_{t+1}|\, x_t)\; \alpha_t(x_t) \tag{2.52}$$

Comme ci-dessus, ces récursions sont calculables si les transitions $p(y_{t+1}|\, x_{t+1}, y_1^t)$ sont calculables, ce qui sera le cas dans le cas gaussien.

On déduit de (2.14) la loi marginale a posteriori :

$$\forall t \in \{1,...,T\} \qquad p(x_t|\, y) = \frac{\beta_t(x_t)\; \alpha_t(x_t)}{\sum_{x_t \in \Omega} \beta_t(x_t)\; \alpha_t(x_t)} \tag{2.53}$$

Comme dans le cas des chaînes de Markov cachées à bruit indépendant, le calcul des marginales a posteriori reste possible pour les chaînes de Markov cachés à dépendance longue toujours de manière récursive, à partir du moment où les transitions $(p(y_{t+1}|\, x_{t+1}, y_1^t))_{t\in\{1,...,T-1\}}$ sont données ou calculables. En utilisant le même algorithme que celui décrit dans le paragraphe précédent on peut donc résoudre le problème de l'estimation de l'état caché. Si on connaît les paramètres de transition de la chaîne cachée et les densités $p(y_{t+1}|\, x_{t+1}, y_1^t)$ on peut donc effectuer les deux procédures filtrage et lissage.

2.5 Estimation des paramètres

Dans plusieurs applications la loi du processus couple $Z = (X, Y)$ n'est pas connue ; les paramètres de la loi $p(z)$ sont alors à estimer, entièrement ou partiellement. Dans ce mémoire on va s'intéresser à deux algorithmes généraux que sont « Espérance Maximisation » (EM) et « Estimation Conditionnelle Itérative » (ECI). Ces méthodes ont fait leurs preuves dans le contexte des différents modèles de Markov cachés et leurs différentes extensions [49] [28] [29] [33] [40] [41] [46] [47]. Ces méthodes permettent une segmentation efficace dans un cadre non supervisé. Nous précisons ci-après la méthode EM dans le cadre des CMC-BI, et la méthode ECI dans le cadre des CMC-BML, la méthode EM étant difficile à utiliser dans ce dernier cadre.

Nous allons utiliser les notations suivantes :

Θ : L'espace des paramètres du modèle ;
$P_\Theta = \{ p(x,y \mid \theta),\ \theta \in \Theta \}$: L'ensemble des lois paramétriques de $p(x,y \mid \theta)$;
$\hat{\theta}(z)$: Un estimateur de θ calculable à partir de données complètes $z = (x,y)$;
$\overline{\theta} = E[\hat{\theta}(X,Y) \mid Y = y, \theta^*]$: L'espérance conditionnelle de l'estimateur choisi ;
$L_\theta(x,y) = \log[p(x,y \mid \theta)]$: La log-vraisemblance complète du modèle ;
$\Im(\theta,\theta^*) = E[L_\theta(X,Y) \mid Y = y, \theta^*]$: L'espérance conditionnelle de la log-vraisemblance.

2.5.1 L'algorithme EM

EM, ainsi que sa version stochastique SEM, sont deux algorithmes itératifs. Le premier a été proposé par Dempster et al en 1977 [73], et Wu en a publié une généralisation. Pour ce qui concerne la convergence, on pourra consulter la référence [35]. Le deuxième est une approximation stochastique de l'EM, dont l'objectif, au départ, a été d'éviter des maximas locaux.

L'algorithme EM se déroule alternativement entre deux étapes:

- Etape (E): calcul de l'espérance de la log-vraisemblance complète, en utilisant les paramètres courants ;

- Etape (M): extraction des paramètres θ maximisant l'espérance de la log-vraisemblance.

Dans la mise en œuvre de cette méthode on peut être confronté à deux difficultés majeures, chacune étant propre à l'une des deux étapes de l'algorithme. Si la maximisation de l'espérance de log-vraisemblance est difficile, on peut se contenter d'une valeur de paramètres qui permet simplement d'accroître la log-vraisemblance (algorithme GEM). Par ailleurs, si le calcul de l'espérance de la log-vraisemblance est lui même impossible ou trop compliqué, l'une des solutions proposées est de l'approcher par la moyenne empirique en utilisant des simulations selon la loi a posteriori fondée sur les paramètres courants (algorithme MCEM).

Dans sa version originale, lorsque l'on n'utilise pas de simulations, la suite des paramètres produite par EM est déterministe et alors l'initialisation est d'une grande importance. En effet, comme on va le voir dans la partie applications, la convergence vers le maximum global n'est pas assurée et l'algorithme EM peut garantir uniquement la convergence vers un point stationnaire de la fonction de vraisemblance, et donc rester piégé dans un maximum local ou un point selle. Broniatowski *et al.* [69] ont proposé d'introduire une phase stochastique entre le calcul de l'espérance et sa maximisation, ce qui peut permettre d'éviter les minima locaux ; la méthode ainsi obtenue a été nommée « stochastique » EM (SEM) [24] [25] [69].

Remarque 1.3

Une combinaison entre les deux algorithmes EM et SEM a été proposé par Celeux *et al.* [7]. Cette nouvelle version appelée SAEM est un compromis qui est peut permettre d'éviter les minima locaux au début des itérations et se comporte comme EM vers la fin des itérations.

Algorithme 2.3: Estimation des paramètres : ***EM***, ***GEM*** et ***MCEM***.

Initialisation

- On se donne une valeur initiale des paramètres $\theta^{(0)} = \theta_0 \in \Theta$.

Etape (E)

- Calculer $\Im(\theta, \theta^{(n)}) = E[L_\theta(X,Y) \mid Y = y, \theta^{(n)}]$;

→ Si le calcul de $\Im(\theta, \theta^{(n)})$ n'est pas possible (MCEM) ou dans le but d'introduire une perturbation stochastique pour éviter les minima locaux (SEM), on adopte l'approximation suivante:

$$\Im(\theta, \theta^{(n)}) \approx \tfrac{1}{I} \sum_{i=1}^{I} L_\theta(x^{(n,i)}, y) \ ,$$

Où $x^{(n,1)}, x^{(n,2)}, ..., x^{(n,I)}$ est un échantillon (i.i.d) des réalisations de X simulé selon la loi $p(x \mid y, \theta^{(n)})$.

Etape (M)

- Calcul de la nouvelle valeur du paramètre $\theta \in \Theta$ par

$$\theta^{(n+1)} = \arg[\max_{\theta \in \Theta}[\Im(\theta, \theta^{(n)})]]$$

→ Si la maximisation est complexe, on choisit une valeur des paramètres qui permet d'accroitre l'espérance :

$$\theta^{(n+1)} \in \left\{ \theta \in \Theta \mid \Im(\theta, \theta^{(n)}) \geq \Im(\theta^{(n)}, \theta^{(n)}) \right\}$$

Critère d'arrêt

- Si $\left| \log(p(y \mid \theta^{(n+1)}) - \log(p(y \mid \theta^{(n)})) \right| \leq \varepsilon$ alors $\hat{\theta}_{EM}(y) = \theta^{(n+1)}$.

Fin

2.5.2 L'algorithme ECI

L'algorithme ECI, proposé par Pieczynski [68], est également itératif et permet, comme EM, d'estimer les paramètres dans les modèles de Markov cachés. Pour le mettre en œuvre il faut vérifier deux conditions. Premièrement, la possibilité de choisir un estimateur du paramètre d'intérêt calculable à partir de données complètes. Deuxièmement, être en mesure de simuler des réalisations du processus caché selon sa loi conditionnelle aux observations et à

paramètres du modèle connus. Le principe de l'ECI est fondé sur l'idée suivante. On suppose que l'estimateur à partir des données complètes présente de bonnes propriétés en termes de l'erreur quadratique moyenne. On doit l'approcher par une fonction des seules valeurs du processus observé (données incomplètes). La meilleure approximation au sens de l'erreur quadratique moyenne est l'espérance conditionnelle à l'observation y . Comme cette espérance conditionnelle dépend justement du paramètre qu'on cherche à estimer, l'algorithme ECI propose de contourner cette difficulté en utilisant une procédure itérative qui consiste à calculer la valeur suivante des paramètres (l'espérance conditionnelle de l'estimateur à partir des données complètes) en fonction de leurs valeurs courante et des observations.
Si l'espérance conditionnelle de l'estimateur choisi n'est pas calculable pour tous les paramètres, on peut l'approcher par sa moyenne empirique, en obtenant une version stochastique d'ECI, comme c'est le cas pour EM avec sa version stochastique MCEM.

Algorithme 2.4: Estimation des paramètres : ***ECI***.

Initialisation

- On se donne une valeur initiale du paramètre $\theta^{(0)} = \theta_0 \in \Theta$.

Etape Itérative

- Calculer $\theta^{(n+1)} = E[\,\hat{\theta}(X,Y) \,\big|\, Y = y, \theta^{(n)}\,]$;

→ Si le calcul de l'espérance conditionnelle n'est pas possible explicitement pour toutes les composantes de $\hat{\theta}(X,Y)$ on adopte, pour ces composantes $\hat{\theta}_m(X,Y)$, l'approximation suivante:

$$\theta_m^{(n+1)} = \tfrac{1}{I} \sum_{i=1}^{I} \hat{\theta}_m(x^{(n,i)}, y)$$

Où $x^{(n,1)}, x^{(n,2)}, ..., x^{(n,I)}$ un échantillon **(i.i.d)** simulé selon la loi $p(x \,\big|\, y, \theta^{(n)})$.

Critère d'arrêt

- Si $\left\|\theta^{(n+1)} - \theta^{(n)}\right\| \le \varepsilon$ alors $\hat{\theta}_{ICE}(y) = \theta^{(n+1)}$.

Fin

Remarque 1.5

Dans l'algorithme ECI comme pour EM, le critère d'arrêt peut être différent selon le choix de l'utilisateur et la précision souhaitée. On peut consulter dans [48] des comparaisons, au plan théorique, entre EM et ECI. Par ailleurs, on trouvera dans [54] [75] des résultats théoriques concernant la convergence de l'ECI.

2.6 Estimation adaptative des paramètres dans le cas partiellement supervisé

Nous proposons dans cette section des estimateurs adaptatifs, qui peuvent être utilisés en filtrage, en partant de deux méthodes générales EM et ICE. Nous étudions le cas des CMC et nous comparons « EM adaptative » (AEM) avec ECI « adaptative » (AECI). Nous proposons également, ce qui est une contribution originale, une AECI valable dans le cas du bruit à corrélation longue.
On va se restreindre dans nos études au cas où X est une chaîne de Markov à deux états, stationnaire au sens strict et homogène, et on considérera que toutes les densités des lois marginales du processus observé conditionnellement au processus caché sont gaussiennes. La comparaison s'effectuera dans un premier temps dans un cadre entièrement supervisé (paramètres du modèle connus) et puis partiellement supervisé (paramètres des transitions inconnus, moyennes et variances connues).
Les algorithmes adaptatifs AEM et AECI pour l'estimation des paramètres latents ont un intérêt particulier dans le cas du problème de filtrage ; nous allons donner un exemple montrant l'évolution de la restauration par filtrage en fonction du paramètre estimé au cours du temps.

Soit $\Omega = \{\lambda_1, \lambda_2\}$ l'espace d'état de la chaîne de Markov X et Q sa matrice de transitions, supposée symétrique. La taille maximale de l'échantillon est $T = 200$.

On considère les notations suivantes :

La loi initiale et la matrice de transitions seront notées respectivement $p(x_1) = \pi_1$ et $Q = \begin{pmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{pmatrix}$.

Cas des CMC-BI : $\forall t \in \{1, \ldots, T-1\}$, $p(y_t | x_t)$ est une gaussienne de moyenne μ_{x_t} et de variance $\sigma^2_{x_t}$.

Cas des CMC-ML : $\forall t \in \{1, \ldots, T-1\}$, $p(y_t | x_t, y_1^{t-1})$ est une gaussienne de moyenne $\widetilde{\mu}_{x_t}$ et de variance $\widetilde{\gamma}_{x_t}$.

2.6.1 Chaînes de Markov cachées à bruit indépendant

Les algorithmes adaptatifs AEM et AECI sont obtenus à partir des algorithmes EM et ECI de la manière suivante. Soient $\hat{\theta}^t_{EM}$ et $\hat{\theta}^t_{ECI}$ les deux estimateurs donnés respectivement par l'algorithme EM et ECI connaissant les observations y_1^t. Alors $\hat{\theta}^{t+1}_{EM}$ et $\hat{\theta}^{t+1}_{ECI}$ sont calculés en prenant en compte les observations y_1^{t+1} et en initialisant les deux algorithmes avec les estimées données par les deux méthodes à l'instant précédent. Bien entendu, cela engendre une grande instabilité au début des algorithmes, lorsqu'il y a peu de données.

Exemple de simulation

Considérons $\pi_1 = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$, $\mu = \begin{pmatrix} \mu_{\lambda_1} \\ \mu_{\lambda_2} \end{pmatrix} = \begin{pmatrix} -0.5 \\ 0.5 \end{pmatrix}$ et $\sigma = \begin{pmatrix} \sigma_{\lambda_1} \\ \sigma_{\lambda_2} \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/2 \end{pmatrix}$ donnés, et ρ appelé à varier.

Les résultats des estimations avec EM et ECI (estimation du paramètre de transition et restauration du signal connaissant toute l'observation y_1^T jusqu'à l'horizon final T) pour plusieurs valeurs de $\rho \in \{0.1, 0.2, 0.3, 0.4\}$ sont présentés dans la Table 2.1. Les taux d'erreur de la restauration du signal caché sont présentés dans la Table 2.2.

ρ	0.10	0. 20	0.30	0.40
EM $\hat{\rho}$	0.10	0.19	0.36	0.51
ECI $\hat{\rho}$	0.09	0.18	0.39	0.52

Table 2.1. L'estimateur du paramètre de transition avec les deux algorithmes

ρ	0,1	0.2	0.3	0.4
EM	8.5%	6.8%	9.0%	17.1%
ECI	13.0%	6.0%	11.3%	22.0%

Table 2.2. Les taux d'erreurs dans la restauration du signal caché.

On s'intéresse ici au cas partiellement supervisé où le paramètre de transition est inconnu car par la suite on va étudier dans les deux derniers chapitres les modèles triplet à sauts Markovien, ou le cas où tous les paramètres du modèle seront connus sauf le paramètre de transition dans la chaîne des sauts. Comme dans le cas du filtrage les observations s'accumulent au cours du temps ; il sera intéressant de voir comment évolue l'estimation de ce paramètre progressivement dans le temps.

Nous présentons un exemple, qui montre l'évolution de l'estimation de ρ au cours du temps, (le vrai paramètre est $\rho = 0.2$) à la Figure 2.1.

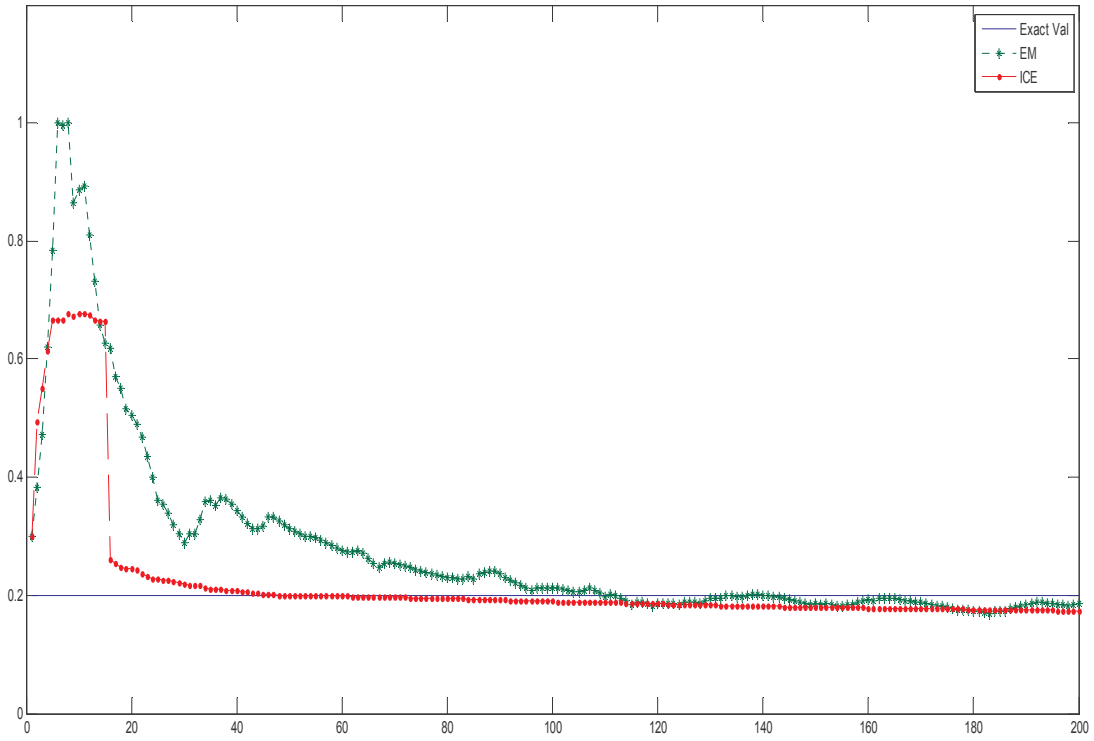


Figure 2.1 : Les courbes représentent l'évolution de l'estimation du paramètre de transition au cours du temps. En rouge (traits longs) l'estimation avec AECI, en vert (traits courts) l'estimation avec AEM, et en bleu la vraie valeur du paramètre.

L'exemple présenté dans la Figure 2.1 est représentatif des différentes simulations effectuées ; on obtient des courbes similaires pour d'autres valeurs du paramètre ρ. Les deux méthodes sont instables au début, aucune de deux n'étant systématiquement supérieure à l'autre. Globalement, on peut conclure de notre étude deux résultats suivants. Le premier est que les méthodes AEM et AECI présentent de bonnes propriétés de stabilité et de convergence. Le deuxième est l'importance de l'initialisation. En effet, dans les cas classiques de EM et ECI la convergence peut dépendre fortement de l'initialisation, plus particulièrement pour EM. Dans le cas adaptatif un autre problème aggrave cette dépendance. Même pour une valeur initiale proche de la vraie valeur du paramètre on constate souvent une divergence dans le calcul des premières estimées, ce qui est dû au nombre insuffisant d'observations.

2.6.2 Chaînes de Markov cachées à mémoire longue

Ici on s'intéressera exclusivement à l'algorithme ECI ; en effet, étant donné la complexité de la vraisemblance dans le cas de la mémoire longue aucune méthode fondée sur EM n'a encore, à notre connaissance, été proposée. Nous proposons une méthode de type AECI en utilisant l'extension récente, et non triviale, de l'algorithme ECI au cas des chaîne de Markov cachée à mémoire longue [46] [47].

Comme on l'a précisé tout au début de ce paragraphe on s'intéresse au cas partiellement supervisé, où seul le paramètre de transition est inconnu.

On rappelle les formules qui permettent le calcul paramètres dans le cas de la mémoire longue [46] :

$$\tilde{\mu}_{x_{t+1}} = \mu_{x_{t+1}} + \Gamma_{x_{t+1}}^{2,1} (\Gamma_{x_{t+1}}^{t+1})^{-1} (y_1^t - \mu_{x_{t+1}}^t); \tag{2.54}$$

$$\tilde{\gamma}_{x_{t+1}} = \sigma_{x_{t+1}}^2 - \Gamma_{x_{t+1}}^{2,1} (\Gamma_{x_{t+1}}^{t+1})^{-1} \Gamma_{x_{t+1}}^{1,2}, \tag{2.55}$$

Avec $\Gamma_{x_{t+1}}^{t+1} = \begin{pmatrix} \Gamma_{x_t}^t & \Gamma_{x_{t+1}}^{2,1} \\ \Gamma_{x_{t+1}}^{1,2} & \sigma_{x_{t+1}}^2 \end{pmatrix}$, $\mu_{x_{t+1}}^t = \underbrace{(\mu_{x_{t+1}}, ..., \mu_{x_{t+1}})}_{t \ \ fois}$ et $\Gamma_{x_{t+1}}^{2,1} = \left(\Gamma_{x_{t+1}}^{1,2}\right)' = (\gamma_{x_{t+1}}(t), ..., \gamma_{x_{t+1}}(0))$

On rappelle l'estimateur du paramètre d'intérêt ρ donné par :

$$\begin{cases} \hat{\rho}_{t+1}^{(0)} = \hat{\rho}_t \\ \hat{\rho}_{t+1}^{(n+1)} = \dfrac{1}{t+1} \displaystyle\sum_{s=1}^{t} p(x_{s+1} \mid x_s, y_1^{s+1}, \hat{\rho}_{t+1}^{(n)}) \delta_{\{x_s \neq x_{s+1}\}} \end{cases} \tag{2.56}$$

Pour le même jeu de paramètre mêmes valeurs de π_1, μ, σ, et $\alpha = (1/3, 1/4)$
On obtient les résultats suivant:

ρ	0.02	0.1	0.25	0.35
ECI	2%	15%	21%	36%

Table 2.3 : Les taux d'erreurs dans la restauration en fonction du paramètre estimé.

Les courbes de restauration dans le cas $\rho = 0.1$ se trouvent à la Figure 2.2.

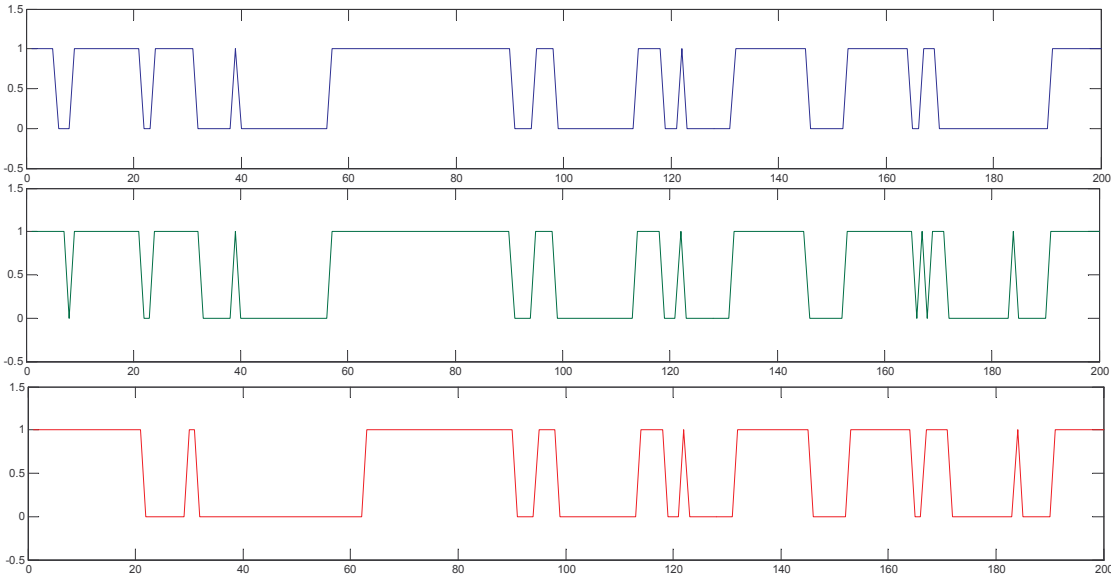


Figure 2.2 Les courbes représentent la restauration du signal caché ; en vert (signal du milieu) à paramètres connus et en rouge (signal du bas) sans les moyennes, les variances et le paramètre de covariance.

Comme dans le cas des CMC, la méthode proposée présente globalement un comportement de bonne qualité et produit des résultats encourageants.

Conclusion

Le modèle de Markov caché à mémoire longue à un intérêt particulier car il prend en compte la corrélation au sein du processus d'observation. Malgré cette complexité supplémentaire, il est possible d'estimer les paramètres par une extension récente de la méthode générale ECI. On utilisera, dans le chapitre 4, ce modèle pour approcher le modèle triplet à sauts Markoviens.

Chapitre 3

Filtrage statistique

L'estimation d'un signal d'intérêt à partir de l'observation d'un autre signal qui lui est corrélé est l'un des problèmes centraux dans le traitement du signal. Généralement le signal observé est considéré comme une version bruitée du signal recherché. On a vu dans le chapitre précédent que dans le cas où le signal caché est modélisé par une chaîne de Markov à espace d'état fini des algorithmes du type MAP ou MPM permettent de calculer l'estimée de l'état latent en maximisant respectivement la probabilité a posteriori et ses marginales. Dans le cas où le signal caché est à espace d'état continu (réel) d'autres algorithmes ont été développés du type filtre de Kalman dans le cas linéaire, EKF [44] et UKF [76] dans le cas non linéaire, ou encore des algorithmes du type filtre particulaire [37]. Le plus souvent dans ces algorithmes l'estimée de l'état latent correspond à la moyenne (l'espérance) de la loi a posteriori marginale, ce qui est optimal au sens de l'erreur quadratique moyenne.

Dans ce chapitre on va se restreindre au problème de filtrage : on suppose que les observations s'accumulent en progressant dans le temps et on a besoin de calculer l'estimée de l'état caché à chaque instant, en prenant en considérations toutes les observations de l'origine jusqu'à l'instant courant. Ce chapitre est consacré aux rappels des méthodes existantes et ne contient pas de contributions originales.

3.1 Systèmes linéaires gaussiens

On appelle processus aléatoire réel en temps discret une famille $\{X_t, t \in \aleph\}$, qui sera également notée $\{X_t\}$ pour simplifier, de vecteurs aléatoires à valeur dans $\Re^{n_x}$. Un processus aléatoire réel en temps discret gaussien $\{X_t\}$ est un processus aléatoire tel que pour tout $t \in \aleph^*$ le vecteur $(X_1, X_2, ..., X_t)$ est un vecteur aléatoire gaussien (de taille $n_x \times t$).

Un bruit blanc $\{X_t\}$ est un processus aléatoire centré (i.e. $E[X_t] = 0 \ \forall t \in \aleph^*$) tel que $E[X_t X_s^{'}] = 0$ pour tout $t \neq s$.

On dit que deux processus aléatoires $\{X_t^{(1)}\}$, $\{X_t^{(2)}\}$ sont indépendants si pour tout $t \in \aleph^*$, les vecteurs aléatoires $(X_1^{(1)}, X_2^{(1)}, ..., X_t^{(1)})$ et $(X_1^{(2)}, X_2^{(2)}, ..., X_t^{(2)})$ sont indépendants.

3.1.1 Equation d'état

On considère l'équation dynamique suivante :

$$X_{t+1} = m_{t+1} + F_{t+1} X_t + W_{t+1} \tag{3.1}$$

Où $\{X_t\}$ et $\{W_t\}$ prennent leurs valeurs dans $\Re^{n_x}$ et $F_t \in \Re^{n_x \times n_x}$.

La suite $(m_t)_{t\in\aleph^*}$ est une suite réelle non aléatoire.
On suppose également:

- Le bruit $\{W_t\}$ est un bruit blanc gaussien de covariance Q_t^W ;
- La variable X_0 est gaussienne, de moyenne $\overline{m}_0$ et de matrice var-covariance Q_0^X ;
- Le bruit $\{W_t\}$ et la variable X_0 sont indépendants.

Le processus $\{X_t\}$ vérifiant (3.1) est alors un processus gaussien. En particulier, X_t est une variable gaussienne, de moyenne $\overline{m}_t$ et de matrice var-covariance Q_t^X avec:

$$\overline{m}_t = m_t + F_t\,\overline{m}_{t-1}, \quad Q_t^X = F_t\,Q_{t-1}^X\,F_t^{'} + Q_t^W$$

3.1.2 Equation d'état et d'observation

On considère le système dynamique suivant :

$$\begin{cases} X_{t+1} = m_{t+1} + F_{t+1}\,X_t + W_{t+1} \\ Y_t = \mu_t + G_t\,X_t + V_t \end{cases} \tag{3.2}$$

Où les deux processus $\{Y_t\}$ et $\{V_t\}$ prennent leurs valeurs dans $\Re^{n_y}$ et $G_t \in \Re^{n_y \times n_x}$ est une suite de matrice déterministe.

La suite $(\mu_t)_{t\in\aleph^*}$ est une suite réelle et non aléatoire. On suppose également que :

- Le processus $\{V_t\}$ est un bruit blanc gaussien de matrice var-covariance Q_t^V ;
- Les bruits $\{W_t\}$ et $\{V_t\}$, et la variable X_0 sont mutuellement indépendants.

Dans **(3.2)**, X_t modélise l'état d'un système non observé directement à l'instant t, mais l'on dispose d'une observation y_t qui correspond à une réalisation de la variable Y_t.
Le système **(3.2)** est linéaire gaussien ; on en déduit que le processus couple $\{(X_t, Y_t)\}$ est également gaussien. En particulier, $(X_t, Y_t)^{'}$ est un vecteur gaussien dont la moyenne et la matrice de var-covariance sont données par:

$$\begin{pmatrix} \overline{m}_t \\ \overline{\mu}_t \end{pmatrix} = \begin{pmatrix} m_t + F_t\,\overline{m}_{t-1} \\ \mu_t + G_t\,\overline{m}_t \end{pmatrix}, \quad \begin{pmatrix} Q_t^X & Q_t^{XY} \\ Q_t^{YX} & Q_t^Y \end{pmatrix} = \begin{pmatrix} F_t\,Q_{t-1}^X\,F_t^{'} + Q_t^W & Q_t^X\,G_t^{'} \\ G_t\,Q_t^X & G_t\,Q_t^X\,G_t^{'} + Q_t^V \end{pmatrix}$$

Remarque 3.1

On peut trouver dans la littérature des modèles plus complexes où :

- Les coefficients F_t et Q_t^W peuvent dépendre de $(Y_1, Y_2, ..., Y_t)$;
- Les coefficients G_t et Q_t^V peuvent dépendre de $(Y_1, Y_2, ..., Y_{t-1})$.

3.1.3 Filtre de Kalman Bucy

Le modèle défini dans les deux paragraphes précédents est donné sous forme Markovienne, d'autre part les matrices F_t et H_t, qui définissent la représentation d'état Markovienne, dépendent du temps et le signal observé est une combinaison linéaire du signal caché. Le filtre de Kalman permet alors de calculer l'estimée optimale au sens de l'erreur quadratique moyenne.

Dans la suite on va supposer, afin de simplifier les écritures, que $\forall t \in \aleph^*$, $F_t = F$, $G_t = G$, $Q_t^W = Q^W$, et $Q_t^V = Q^V$.

On a alors :

$$\begin{cases} X_{t+1} = m_t + F\, X_t + W_{t+1} \\ Y_t = \mu_t + G\, X_t + V_t \end{cases} \tag{3.3}$$

Le filtre de Kalman Bucy [11][12] [13] qu'on va décrire est un algorithme à deux étapes, dites « correction » et « prédiction ». Il est possible de commencer indifféremment par l'une ou l'autre ; il suffit de choisir la bonne convention pour l'état initial.
Nous avons la propriété classique suivante :

Proposition 3.1

Soit $Z = (X, Y)'$ un vecteur aléatoire gaussien avec la matrice des covariances $Q^{YY} = E[(Y - E[Y])(Y - E[Y])']$, supposée inversible. Alors le vecteur aléatoire X sachant Y est gaussien de moyenne et de matrice var-covariance

$$E[X \mid Y] = E[X] + Q^{XY} (Q^{YY})^{-1} (Y - E[Y]), \quad Q^{X|Y} = Q^X - Q^{XY} (Q^{YY})^{-1} Q^{YX}$$

Nous allons adopter les notations suivantes

$$\hat{m}_t = E[X_t \mid y_0^t], \quad \hat{P}_t = E[(X_t - \hat{m}_t)(X_t - \hat{m}_t)' \mid y_0^t]$$

$$\widetilde{m}_{t+1} = E[X_{t+1} \mid y_0^t], \quad \widetilde{P}_{t+1} = E[(X_{t+1} - \widetilde{m}_{t+1})(X_{t+1} - \widetilde{m}_{t+1})' \mid y_0^t]$$

$$\widetilde{\mu}_{t+1} = E[Y_{t+1} \mid y_0^t], \quad \widetilde{\Gamma}_{t+1} = E[(Y_{t+1} - \widetilde{\mu}_{t+1})(Y_{t+1} - \widetilde{\mu}_{t+1})' \mid y_0^t]$$

L'étape de prédiction ne dépend pas de l'observation y_{t+1} mais du dernier estimé, ce qui rend les calculs plus simples par rapport à l'étape de filtrage. La moyenne et la matrice de var-covariance sont données par:

$$\widetilde{m}_{t+1} = E[X_{t+1} \mid y_0^t] = m_{t+1} + F\, E[X_t \mid y_0^t] = m_{t+1} + F\, \hat{m}_t,$$

$$\begin{aligned}
\widetilde{P}_{t+1} &= E[(X_{t+1} - \widetilde{m}_{t+1})(X_{t+1} - \widetilde{m}_{t+1})' \mid y_0^t] \\
&= E[X_{t+1}\, X_{t+1}' \mid y_0^t] - \widetilde{m}_{t+1}\, \widetilde{m}_{t+1}' \\
&= E[(m_{t+1} + F\, X_t + W_{t+1})\,(m_{t+1} + F\, X_t + W_{t+1})' \mid y_0^t] - (m_{t+1} + F\, \hat{m}_t)\,(m_{t+1} + F\, \hat{m}_t)' \\
&= F\, E[X_t\, X_t' \mid y_0^t]\, F' + Q^W - F\, \hat{m}_t\, \hat{m}_t'\, F' \\
&= F\,(E[X_t\, X_t' \mid y_0^t] - \hat{m}_t\, \hat{m}_t')\, F' + Q^W \\
&= F\, \hat{P}_t\, F' + Q^W
\end{aligned}$$

L'étape de filtrage, appelée aussi étape de correction, prend en compte la dernière observation y_{t+1}.
Nous avons :

$$\begin{aligned}
\widetilde{\mu}_{t+1} = E[Y_{t+1} \mid y_0^t] &= \mu_{t+1} + G\, E[X_{t+1} \mid y_0^t] \\
&= \mu_{t+1} + G\, \widetilde{m}_{t+1} \\
&= \mu_{t+1} + G\, m_{t+1} + G\, F\, \hat{m}_t,
\end{aligned}$$

$$\begin{aligned}
\widetilde{\Gamma}_{t+1} &= E[(Y_{t+1} - \widetilde{\mu}_{t+1})(Y_{t+1} - \widetilde{\mu}_{t+1})' \mid y_0^t] \\
&= E[(\mu_{t+1} + G\, X_{t+1} + V_{t+1})\,(\mu_{t+1} + G\, X_{t+1} + V_{t+1})' \mid y_0^t] - (\mu_{t+1} + G\, \widetilde{m}_{t+1})\,(\mu_{t+1} + G\, \widetilde{m}_{t+1})' \\
&= G\, \widetilde{P}_t\, G' + Q^V \\
&= G\,(F\, \hat{P}_t\, F' + Q^W)\, G' + Q^V = G\, F\, \hat{P}_t F' G' + G\, Q^W G' + Q^V.
\end{aligned}$$

D'autre part le vecteur couple $(\widetilde{X}_{t+1}, \widetilde{Y}_{t+1}) = (X_{t+1} - \widetilde{m}_{t+1},\ Y_{t+1} - \widetilde{\mu}_{t+1})$ est gaussien ; en utilisant la proposition 3.1 nous avons :

$$E[\widetilde{X}_{t+1} \mid \widetilde{Y}_{t+1}] = Q_{t+1}^{\widetilde{X}\widetilde{Y}}\, (Q_{t+1}^{\widetilde{Y}\widetilde{Y}})^{-1}\, \widetilde{Y}_{t+1},$$

$$E[(\widetilde{X}_{t+1} - E[\widetilde{X}_{t+1} \mid \widetilde{Y}_{t+1}])(\widetilde{X}_{t+1} - E[\widetilde{X}_{t+1} \mid \widetilde{Y}_{t+1}])' \mid \widetilde{Y}_{t+1}] = Q_{t+1}^{\widetilde{X}\widetilde{X}} - Q_{t+1}^{\widetilde{X}\widetilde{Y}}\ (Q_{t+1}^{\widetilde{Y}\widetilde{Y}})^{-1}\, Q_{t+1}^{\widetilde{Y}\widetilde{X}}$$

avec :

$$Q_{t+1}^{\widetilde{X}\widetilde{X}} = E[\widetilde{X}_{t+1}\, \widetilde{X}_{t+1}'] = \widetilde{P}_{t+1},$$

$$Q_{t+1}^{\widetilde{X}\widetilde{Y}} = (Q_{t+1}^{\widetilde{Y}\widetilde{X}})' = E[\widetilde{X}_{t+1}\, \widetilde{Y}_{t+1}'] = E[(X_{t+1} - \widetilde{m}_{t+1})\,(Y_{t+1} - \widetilde{\mu}_{t+1})'] = \widetilde{P}_{t+1}\, G',$$

$$Q_{t+1}^{\tilde{Y}\tilde{Y}} = E[\tilde{Y}_{t+1}\,\tilde{Y}_{t+1}^{'}] = E[(Y_{t+1} - \tilde{\mu}_{t+1})\,(Y_{t+1} - \tilde{\mu}_{t+1})^{'}] = G\,\tilde{P}_{t+1}\,G^{'} + Q^{V}.$$

Les deux vecteurs aléatoires gaussiens $\tilde{X}_{t+1}$ et $(Y_1, Y_2, ..., Y_t)$ sont indépendants, on en conclut que :

$$E[\tilde{X}_{t+1} \,|\, y_1^{t+1}] = E[\tilde{X}_{t+1} \,|\, y_1^{t}, \tilde{y}_{t+1}] = E[\tilde{X}_{t+1} \,|\, \tilde{y}_{t+1}]$$

De même on a :

$$E[(\tilde{X}_{t+1} - E[\tilde{X}_{t+1} \,|\, y_1^{t+1}])(\tilde{X}_{t+1} - E[\tilde{X}_{t+1} \,|\, y_1^{t+1}])^{'} \,|\, y_1^{t+1}]$$
$$= E[(\tilde{X}_{t+1} - E[\tilde{X}_{t+1} \,|\, \tilde{y}_{t+1}])(\tilde{X}_{t+1} - E[\tilde{X}_{t+1} \,|\, \tilde{y}_{t+1}])^{'} \,|\, \tilde{y}_{t+1}]$$

Alors :

$$\hat{m}_{t+1} = E[X_{t+1} \,|\, y_1^{t+1}] = E[\tilde{X}_{t+1} \,|\, \tilde{y}_{t+1}] + \tilde{m}_{t+1}$$
$$= \tilde{m}_{t+1} + \tilde{P}_{t+1}\,G^{'}(G\,\tilde{P}_{t+1}\,G^{'} + Q^{V})^{-1}\,(y_{t+1} - \tilde{\mu}_{t+1}),$$

$$\hat{P}_{t+1} = E[(X_{t+1} - E[X_{t+1} \,|\, y_1^{t+1}])(X_{t+1} - E[X_{t+1} \,|\, y_1^{t+1}])^{'} \,|\, y_1^{t+1}]$$
$$= E[(\tilde{X}_{t+1} - E[\tilde{X}_{t+1} \,|\, \tilde{y}_{t+1}])(\tilde{X}_{t+1} - E[\tilde{X}_{t+1} \,|\, \tilde{y}_{t+1}])^{'} \,|\, \tilde{y}_{t+1}]$$
$$= \tilde{P}_{t+1} - \tilde{P}_{t+1}\,G^{'}(G\,\tilde{P}_{t+1}\,G^{'} + Q^{V})^{-1}\,G\,\tilde{P}_{t+1}.$$

En notant $\Delta_{t+1} = \tilde{P}_{t+1}\,G^{'}(G\,\tilde{P}_{t+1}\,G^{'} + Q^{V})^{-1}$ la matrice de « gain » du filtre de Kalman on a alors :

$$\hat{m}_{t+1} = \tilde{m}_{t+1} + \Delta_{t+1}\,(y_{t+1} - \tilde{\mu}_{t+1})$$

$$\hat{P}_{t+1} = \tilde{P}_{t+1} - \Delta_{t+1}\,G\,\tilde{P}_{t+1} = (I - \Delta_{t+1}\,G)\,\tilde{P}_{t+1}$$

On résume les deux étapes de prédiction et de filtrage dans l'algorithme qui suit :

Algorithme 3.1 : Filtre de Kalman Bucy

➔ $t = 0$

- Initialisation

$$X_0 \sim N(m_0, P_0) \quad \text{On pose} \quad \begin{cases} \hat{m}_0 = m_0 \\ \hat{P}_0 = P_0 \end{cases}$$

➔ $t \in \{1, ..., T\}$

- Prédiction

$$\begin{cases} \widetilde{m}_t = m_t + F\,\hat{m}_{t-1} \\ \widetilde{P}_t = F\,\hat{P}_t\,F' + Q^W \end{cases}$$

- Innovation

$$\begin{cases} \widetilde{\mu}_t = \mu_t + G\,m_t + G\,F\,\hat{m}_{t-1} \\ \widetilde{\Gamma}_t = G\,F\,\hat{P}_t F' G' + G\,Q^W G' + Q^V \end{cases}$$

- Filtrage

$$\begin{cases} \hat{m}_t = \widetilde{m}_t + \widetilde{P}_t\,G'(G\,\widetilde{P}_t\,G' + Q^V)^{-1}\,(y_t - \widetilde{\mu}_t) \\ \hat{P}_t = \widetilde{P}_t - \widetilde{P}_t\,G'(G\,\widetilde{P}_t\,G' + Q^V)^{-1}\,G\,\widetilde{P}_t \end{cases}$$

Fin

3.2 Systèmes linéaires non gaussiens

Dans le système **(3.3)** du paragraphe précédent on se trouvait dans un cadre gaussien et l'évolution du système est régie par une dynamique simple (linéaire). Le filtre de Kalman Bucy donne une solution optimale en minimisant la variance. Dans le cas non gaussien, la solution donnée par l'algorithme de Kalman, exposée ci-après, est sous-optimale.
On se place dans un modèle linéaire de la forme suivante :

$$\begin{cases} X_{t+1} = F\,X_t + W_{t+1} \\ Y_t = G\,X_t + V_t \end{cases} \qquad (3.5)$$

La problématique ne change pas, on suppose que le processus $\{X_t\}$ n'est pas observable directement, on souhaite estimer à chaque instant t la valeur prise par X_t. On dispose pour cela d'une observation, qui correspond à une réalisation y_1^t des variables $\{Y_1^t\}$. Les paramètres du système dynamique sont supposés connus (les matrices F et G, ainsi que les moyennes et les matrices de var-covariances des bruits sont connues).

Ce modèle est largement employé en navigation et radioguidage. Dans ce type d'applications, on étend souvent l'équation d'évolution à :

$$X_{t+1} = m_{t+1} + F\,X_t + W_{t+1},$$

où m_{t+1} est un vecteur déterministe. L'état X_t est un vecteur aléatoire qui rassemble les coordonnées (position et vitesse) à l'instant t d'un objet volant ; l'équation d'évolution est alors une discrétisation de la relation fondamentale de la dynamique, elle représente la façon dont X_t évolue dans le temps en fonction de l'état à l'instant précédent et des forces qui

s'exercent sur l'objet volant, que sont la poussée des moteurs m_{t+1} et les perturbations incontrôlées W_{t+1}. L'état n'est pas disponible mais on dispose des observations radar $Y_t = y_t$, qui sont reliées à X_t, avec V_t bruit de mesure. Le but est d'estimer à tout instant t la position et la vitesse de l'objet volant, en corrigeant si besoin la trajectoire grâce à la commande des moteurs m_{t+1}.
La présence de m_{t+1} ne modifie pas le problème de l'estimation de X_t ; afin de simplifier nous garderons pour la suite la forme (3.5).

Nous complèterons les hypothèses faites, en précisant que W et V sont des processus centrés et composés des variables décorrélées entre elles (ce sont des bruits blancs). On considère également qu'ils sont décorrélés de X_0 et mutuellement décorrélés. Ces hypothèses sont résumées dans :

$$E\left(\begin{bmatrix} X_0 \\ W_t \\ V_t \end{bmatrix}\begin{bmatrix} 1 & X_0^{'} & W_s^{'} & V_s^{'} \end{bmatrix}\right) = \begin{pmatrix} m_0 & P_0 + m_0\, m_0^{'} & 0 & 0 \\ 0 & 0 & Q^W \delta_{\{t=s\}} & Q^{WV} \delta_{\{t=s\}} \\ 0 & 0 & (Q^{WV})^{'} \delta_{\{t=s\}} & Q^V \delta_{\{t=s\}} \end{pmatrix} \qquad (3.6)$$

Un tel modèle est physiquement pertinent dans un problème de navigation ; par ailleurs les hypothèses d'orthogonalité de W et de V et conjointement de W et V permettent de simplifier considérablement les équations du filtre. Nous supposerons également que la matrice Q^V est inversible pour tout t.

L'idée exposée ci-dessus était de calculer un estimateur de X_t sachant $Y_1^t = y_1^t$ en utilisant le critère de minimisation de l'erreur quadratique moyenne sous contrainte linéaire. Cependant, la solution dans ce cas, qui correspond à l'espérance conditionnelle $\hat{m}_t = E(X_t \mid y_1^t)$, n'a aucune raison d'être une fonction linéaire des données y_1^t, sauf dans des cas particuliers comme dans le paragraphe précédent où les bruits sont gaussiens. Par ailleurs, le calcul de l'espérance conditionnelle nécessite la connaissance de la loi jointe de X_t et de Y_1^t, ce qui n'est pas le cas dans le modèle considéré. Par conséquent on considère une solution sous-optimale qui est donnée par

$$\hat{m}_t = \mathrm{Pr}_{H_t(y)}(X_t) = \underset{h(.)linéaire}{Min}\left[\left(E\left(X_t - h\left(Y_1^t\right)\right)\right)^2\right] = \underset{A_1, A_2, \ldots, A_t}{Min}\left[\left(E\left(X_t - \sum_{t=1}^{T_0} A_t\, Y_t\right)\right)^2\right] \quad \textbf{(3.7)}$$

La solution $\hat{m}_t = \mathrm{Pr}_{H_t(y)}(X_t)$ est donc la projection de X_t sur l'espace $H_t(Y) = \{$ensemble des combinaisons linéaires des $Y_1, \ldots, Y_t\}$.

Pour décrire le filtre nous aurons besoin des notations suivantes :

$$\hat{\chi}_{t+1} = X_{t+1} - \hat{m}_{t+1} = X_{t+1} - P_{H_{t+1}(Y)}(X_{t+1}),$$

$$\tilde{\chi}_{t+1} = X_{t+1} - \tilde{m}_{t+1} = X_{t+1} - P_{H_t(Y)}(X_{t+1}),$$

$$\hat{P}_{t+1} = \left\langle \hat{\chi}_{t+1} \middle| \hat{\chi}_{t+1} \right\rangle_{H_{t+1}}, \quad \widetilde{P}_{t+1} = \left\langle \widetilde{\chi}_{t+1} \middle| \widetilde{\chi}_{t+1} \right\rangle_{H_t} ,$$

$$\widetilde{\varepsilon}_{t+1} = Y_{t+1} - \widetilde{\mu}_{t+1} = Y_{t+1} - P_{H_t(Y)}(Y_{t+1}), \quad \text{et} \quad \widetilde{\Gamma}_{t+1} = \left\langle \widetilde{\varepsilon}_{t+1} \middle| \widetilde{\varepsilon}_{t+1} \right\rangle_{H_t} .$$

Les équations du filtre de Kalman s'obtiennent comme dans le cas gaussien en deux étapes : étape de filtrage et l'étape de prédiction. L'algorithme nous permet de commencer indifféremment par l'une ou l'autre. Dans ce paragraphe on va donner la version inverse à celle donnée dans le cas gaussien c'est-à-dire on commence par la mise à jour (l'actualisation dite étape de filtrage) ensuite l'étape de propagation (prédiction). Nous allons nous contenter de donner l'algorithme ; pour les démonstrations on peut consulter les références [11]. On peut y trouver une version détaillée pour les différentes équations du filtre.

L'étape dite de filtrage nécessite le calcul au préalable de l'innovation $\widetilde{\varepsilon}_t$ et $\widetilde{\Gamma}_t$.
Nous avons

$$\begin{aligned}\widetilde{\varepsilon}_t &= Y_t - P_{H_{t-1}(Y)}(Y_t) = Y_t - P_{H_{t-1}(Y)}(G\,X_t + V_t) = Y_t - G\,P_{H_{t-1}(Y)}(X_t) \\ &= Y_t - G\ \widetilde{m}_t ,\end{aligned}$$

$$\begin{aligned}\widetilde{\Gamma}_t &= \left\langle \widetilde{\varepsilon}_t \middle| \widetilde{\varepsilon}_t \right\rangle_{H_{t-1}} = \left\langle Y_t - G\ \widetilde{m}_t \middle| Y_t - G\ \widetilde{m}_t \right\rangle_{H_{t-1}} \\ &= \left\langle G\,X_t + V_t - G\ \widetilde{m}_t \middle| G\,X_t + V_t - G\ \widetilde{m}_t \right\rangle_{H_{t-1}} \\ &= G \left\langle X_t - \widetilde{m}_t \middle| X_t - \widetilde{m}_t \right\rangle_{H_{t-1}} G' + \left\langle V_t \middle| V_t \right\rangle_{H_{t-1}} \\ &= G\,\widetilde{P}_t\,G' + Q^V .\end{aligned}$$

On définit la matrice de gain Δ_t de filtrage de Kalman telle que :

$$\hat{m}_t = P_{H_t(Y)}(X_t) = P_{H_{t-1}(Y)}(X_t) + \Delta_t(Y_t - P_{H_{t-1}(Y)}(Y_t)) = \widetilde{m}_t + \Delta_t\,\widetilde{\varepsilon}_t ,$$

$$\begin{aligned}\hat{P}_t &= \left\langle \hat{\chi}_t \middle| \hat{\chi}_t \right\rangle_{H_t} = \left\langle X_t - \hat{m}_t \middle| X_t - \hat{m}_t \right\rangle_{H_t} = \left\langle X_t - \widetilde{m}_t - \Delta_t\,\widetilde{\varepsilon}_t \middle| X_t - \widetilde{m}_t - \Delta_t\,\widetilde{\varepsilon}_t \right\rangle_{H_t} \\ &= \left\langle X_t - \widetilde{m}_t \middle| X_t - \widetilde{m}_t \right\rangle_{H_{t-1}} + \Delta_t \left\langle \widetilde{\varepsilon}_t \middle| \widetilde{\varepsilon}_t \right\rangle_{H_{t-1}} \Delta_t' = \left\langle \widetilde{\chi}_t \middle| \widetilde{\chi}_t \right\rangle_{H_{t-1}} + \Delta_t \left\langle \widetilde{\varepsilon}_t \middle| \widetilde{\varepsilon}_t \right\rangle_{H_{t-1}} \Delta_t' \\ &= \widetilde{P}_t + \Delta_t\,\widetilde{\Gamma}_t\,\Delta_t' ,\end{aligned}$$

où

$$\Delta_t = \widetilde{P}_t\,G'\,\widetilde{L}_t^{-1} = \widetilde{P}_t\,G'(G\,\widetilde{P}_t\,G' + Q^V)^{-1}$$

L'étape de prédiction est directe et relativement simple :

$$\widetilde{m}_{t+1} = P_{H_t(Y)}(X_{t+1}) = P_{H_t(Y)}(F\,X_t + W_{t+1}) = F\,P_{H_t(Y)}(X_t) = F\,\hat{m}_t ,$$

$$\widetilde{P}_{t+1} = \left\langle \widetilde{\chi}_{t+1} \middle| \widetilde{\chi}_{t+1} \right\rangle_{H_t} = \left\langle X_{t+1} - \widetilde{m}_{t+1} \middle| X_{t+1} - \widetilde{m}_{t+1} \right\rangle_{H_t} = \left\langle X_{t+1} - F\,\hat{m}_t \middle| X_{t+1} - F\,\hat{m}_t \right\rangle_{H_t}$$
$$= F \left\langle X_t - \hat{m}_t \middle| X_t - \hat{m}_t \right\rangle_{H_{t-1}} F' + \left\langle W_{t+1} \middle| W_{t+1} \right\rangle_{H_{t-1}} = F\,\hat{P}_t\,F' + Q^W$$

Les équations retrouvées sont similaires à celles retrouvées dans le cas gaussien.

Remarque 3.2

En pratique, ces équations dites du filtre de Kalman sont rarement implémentées directement sur ordinateur telles qu'elles sont parce que la propagation d'erreur d'arrondi peut conduire à une perte du caractère symétrique semi-défini positif des matrices de covariance. Il est cependant possible de remédier à ce problème d'ordre numérique en remplaçant l'algorithme de Kalman par un algorithme dit « à factorisation » ou « en racine carrée ». Les principes des algorithmes à factorisation consiste à remplacer la propagation des matrices de var-covariances par celle de leurs racines carrées.

3.3 Système gaussien non linéaire

On considère le système stochastique non linéaire suivant :

$$\begin{cases} X_{t+1} = f(X_t) + W_{t+1} \\ Y_t = g(X_t) + V_t \end{cases} \qquad \textbf{(3.8)}$$

Où f et g sont deux fonctions non linéairse, définies sur $\Re^{n_x}$, à valeurs respectivement dans $\Re^{n_x}$ et $\Re^{n_y}$.
On suppose que les fonctions f et g sont dérivables. Les bruits $\{W_t\}$ et $\{V_t\}$ sont des bruits blancs gaussiens (de var-covariances respectives Q^W et Q^V) indépendants entre eux et indépendants de l'état initial du système X_0.
La plupart des propriétés obtenues précédemment ne sont plus vraies. En particulier, le calcul des espérances et des matrices var-covariances devient compliqué. Du fait de la linéarité des systèmes vus précédemment le filtre se calcule de manière récursive dans le temps et les équations sont relativement simples. Dans le cas non linéaire d'autres algorithmes ont vu le jour comme le filtre de Kalman linéarisé, des variantes comme EKF, ou encore le célèbre filtre particulaire.

3.3.1 Filtre de Kalman étendu (EKF)

La méthode consiste à calculer l'estimateur $\hat{m}_{t+1}$ à l'instant $(t+1)$ en linéarisant les fonctions f et g autour de l'estimé $\hat{m}_t$ à l'instant précédent. On va se contenter dans la description de cette méthode d'un développement à l'ordre 1 ; un ordre plus élevé est nécessaire pour une approximation plus précise.
Nous avons

$$f(x) \approx f(\hat{m}_t) + df(\hat{m}_t)\,(x - \hat{m}_t) \qquad (3.9)$$

$$g(x) \approx g(\hat{m}_t) + dg(\hat{m}_t)\,(x - \hat{m}_t) \tag{3.10}$$

Les fonctions df et dg désignent les dérivées à l'ordre 1 respectivement de f et g.

Le système (3.8) est remplacé par le système linéaire composé de (3.9), (3.10) suivant :

$$\begin{cases} X_{t+1} = m_{t+1} + F_t\, X_t + W_{t+1} \\ Y_t = \mu_t + G_t\, X_t + V_t \end{cases}, \tag{3.11}$$

Où $F_t = df(\hat{m}_t),\ G_t = dg(\hat{m}_t),\ m_{t+1} = f(\hat{m}_{t+1}) - F_t\,\hat{m}_t$ et $\mu_{t+1} = g(\hat{m}_{t+1}) - G_t\,\hat{m}_t$.

On applique alors le filtre de Kalman à ce nouveau système linéaire.

Remarques 3.3

- On peut s'attendre à de bons résultats avec cette technique de filtrage lorsque l'on est proche d'une situation « linéaire » ou lorsque le rapport signal/bruit est grand ;

- Pour vérifier si le filtre de Kalman linéarisé est bien adapté, on peut, en sortie, tester le processus de « pseudo innovation » $\tilde{\varepsilon}_t = Y_t - \tilde{\mu}_t$, et vérifier s'il est « proche » d'un bruit blanc ;

- Le choix du système de coordonnées dans lequel on exprime le problème influence beaucoup le comportement du filtre de Kalman linéarisé.

3.3.2 UKF (Unscented Kalman Filter)

Le filtre UKF est également récursif. Contrairement au filtre de Kalman linéarisé ici on n'impose pas le calcul de la dérivée et on ne fait pas d'hypothèse de régularité sur les fonctions f et g. Le calcul de l'estimée à l'instant présent repose sur le calcul de la moyenne et la matrice de var-covariance dans les deux étapes de prédiction et filtrage en considérant un développement linéaire pondéré autour des estimés respectifs à l'instant précédent.
On explicite le principe du filtre en dimension $n_x = 1$; on pourra consulter [76] pour le cas général. Soit $\hat{m}_t$ et $\hat{P}_t$ les estimés respectifs de la moyenne et la matrice var-covariance à l'instant courant.
On commence par le calcul des sigma-points :

$$z_x^{(0)} = \hat{m}_t,\quad z_x^{(i)} = \hat{m}_t + \sqrt{(n_x + \lambda)}\left(\hat{P}_t^{1/2}\right)^{(i)} \text{ et } \quad z_x^{(j)} = \hat{m}_t - \sqrt{(n_x + \lambda)}\left(\hat{P}_t^{1/2}\right)^{(j-n_x)},$$

où $i \in \{1,...,n_x\},\ j \in \{1+n_x,...,2n_x\}$ et $\left(\hat{P}_t^{1/2}\right)^{(i)}$ la $i^{ème}$ colonne de la matrice racine carrée d'autre part λ est un paramètre de décalage [76].

Ensuite on calcule la transformation de ces sigma-points par les fonctions non linéaires du système (3.8). On obtient les points d'intérêt suivants :

$$\forall n \in \{0,...,2n_x\} \qquad \tilde{z}_x^{(n)} = f(z_x^{(n)}) \quad \& \quad \tilde{z}_y^{(n)} = f(\tilde{z}_x^{(n)})$$

L'étape de pondération (équation prédiction, paramètres d'innovation) consiste à poser :

$$\tilde{m}_{t+1} = \sum_{n=0}^{2n_x} \varpi_n^{(m)} \tilde{z}_x^{(n)} \qquad \& \qquad \tilde{P}_{t+1} = \sum_{n=0}^{2n_x} \varpi_n^{(c)} (\tilde{z}_x^{(n)} - \tilde{m}_{t+1})(\tilde{z}_x^{(n)} - \tilde{m}_{t+1})' + Q^W$$
$$\tilde{\mu}_{t+1} = \sum_{n=0}^{n_x} \varpi_n^{(m)} \tilde{z}_y^{(n)} \qquad \& \qquad \tilde{\Gamma}_{t+1} = \sum_{n=0}^{2n_x} \varpi_n^{(c)} (\tilde{z}_y^{(n)} - \tilde{\mu}_{t+1})(\tilde{z}_y^{(n)} - \tilde{\mu}_{t+1})' + Q^V,$$

où les paramètres de pondérations sont donnés par :

$$\varpi_0^{(m)} = \frac{\lambda}{(n_x + \lambda)}, \ \varpi_0^{(c)} = \frac{\lambda}{(n_x + \lambda)} + c_0, \ \varpi_n^{(m)} = \varpi_n^{(c)} = \frac{1}{2(n_x + \lambda)}, \qquad \forall n \in \{1,...,2n_x\}$$

Tout l'ingrédient étant présent on peut passer directement à l'étape de filtrage.

3.3.3 Filtre Particulaire

Le filtre particulaire, que nous allons décrire dans le cadre du modèle (3.8), est une méthode de Monte-Carlo séquentielle qui vise à propager une approximation de la densité conditionnelle $p(x_t \mid y_1^t)$, (les références [16] [11] [15] décrivent de nombreuses applications de ce type de filtrage en particulier celui de poursuite de cible).
Quel que soit le critère bayésien retenu, l'estimateur $\hat{m}_t$ est calculable à partir de la densité conditionnelle $p(x_t \mid y_1^t)$.

De manière générale nous avons ;

$$p(x_1^t \mid y_1^t) = \frac{p(x_1^t, y_1^t)}{p(y_1^t)} = \frac{p(x_t, y_t \mid x_1^{t-1}, y_1^{t-1})}{p(y_t \mid y_1^{t-1})} p(x_1^{t-1} \mid y_1^{t-1})$$
$$= \frac{p(y_t \mid x_1^t, y_1^{t-1}) p(x_t \mid x_1^{t-1}, y_1^{t-1})}{p(y_t \mid y_1^{t-1})} p(x_1^{t-1} \mid y_1^{t-1})$$

Alors par marginalisation on peut écrire :

$$p(x_t \mid y_1^t) = \int \frac{p(y_t \mid x_1^t, y_1^{t-1}) p(x_t \mid x_1^{t-1}, y_1^{t-1})}{p(y_t \mid y_1^{t-1})} p(x_1^{t-1} \mid y_1^{t-1}) \, dx_1^{t-1}$$

Dans le cas d'un système à dynamique markovienne, comme c'est le cas du modèle (3.8), cette marginale se simplifie de la manière suivante :

$$p(x_t \mid y_1^t) = \int \frac{p(y_t \mid x_t)\, p(x_t \mid x_{t-1})}{p(y_t \mid y_1^{t-1})}\, p(x_{t-1} \mid y_1^{t-1})\, dx_{t-1}$$

$$= \frac{p(y_t \mid x_t)}{p(y_t \mid y_1^{t-1})} \int p(x_t \mid x_{t-1})\, p(x_{t-1} \mid y_1^{t-1})\, dx_{t-1}$$

L'intégrale ci-dessus n'étant pas calculable dans le cas général, on n'a pas de formule récursive explicite sur $p(x_t \mid y_1^t)$.

3.3.3.1 Principe du filtre

Le filtrage particulaire, appelé aussi « Boostrap filter », est une méthode de Monte Carlo séquentielle permettant d'approcher la distribution de probabilité conditionnelle de l'état sachant les observations, au moyen de la distribution empirique d'un système de particules. Les particules se déplacent selon des réalisations indépendantes à partir de l'équation d'état, et elles sont corrigées (pondérées) en fonction de leur cohérence avec les observations.

Les conditions de base nécessaire à l'implémentation d'un algorithme particulaire basique sont :

- Savoir générer suivant la loi de transition de l'état $p(x_t \mid x_{t-1})$;
- Savoir évaluer la fonction de vraisemblance $p(y_t \mid x_t)$ en tout point de l'espace d'état.

Les algorithmes de filtrage particulaire consistent à remplacer la propagation de la loi exacte $p(x_t \mid y_1^t)$ par la propagation d'une approximation discrète de cette dernière, qu'on notera $\hat{p}(x_t \mid y_1^t)$, qui est obtenue par tirages aléatoires.

De manière générale, soit $p(x)$ une densité de probabilité associée à une variable aléatoire X, soit $\left\{x^{(1)},...,x^{(N)}\right\}$ un tirage (i.i.d) de taille N selon cette distribution. On définit la densité empirique correspondante à ce tirage par :

$$p(x) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\{x^{(n)}\}}(x),$$

Où δ désigne la mesure de Dirac donnée par

$$\delta_{\{y\}}(x) = \begin{cases} 1 & si \quad x = y \\ 0 & si \quad x \neq y \end{cases}.$$

On suppose qu'on a une densité empirique correspondant à un tirage (i.i.d) de la densité de probabilité $p(x_{t-1} \mid y_1^{t-1})$; le but alors est d'approcher la densité $p(x_t \mid y_1^t)$.
L'algorithme de filtrage comportera deux étapes : celle de « propagation » et celle d' « actualisation ».

3.3.3.2 Equation de propagation

L'équation de propagation correspond à la partie « prédiction » dans le filtre de Kalman, la différence est qu'ici on travaille directement sur la formulation et la propagation de la densité au lieu des espérances et matrices de var-covariance.
Posons

$$\hat{p}(x_{t-1} \mid y_1^{t-1}) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\{x_{t-1}^{(n)}\}}(x_{t-1}) \tag{3.12}$$

On en déduit une approximation de $p(x_t \mid y_1^{t-1})$.
Nous avons

$$\begin{aligned} p(x_t \mid y_1^{t-1}) &= \int p(x_t \mid x_{t-1})\, p(x_{t-1} \mid y_1^{t-1})\, dx_{t-1} \approx \int p(x_t \mid x_{t-1})\, \hat{p}(x_{t-1} \mid y_1^{t-1})\, dx_{t-1} \\ &= \hat{p}(x_t \mid y_1^{t-1}) \end{aligned}$$

Alors

$$\begin{aligned} \hat{p}(x_t \mid y_1^{t-1}) &= \int p(x_t \mid x_{t-1})\, \hat{p}(x_{t-1} \mid y_1^{t-1})\, dx_{t-1} \approx \frac{1}{N}\sum_{n=1}^{N} \int p(x_t \mid x_{t-1})\, \delta_{\{x_{t-1}^{(n)}\}}(x_{t-1})\, dx_{t-1} \\ &= \frac{1}{N}\sum_{n=1}^{N} p(x_t \mid x_{t-1}^{(n)}) \end{aligned}$$

3.3.3.3 Equation d'actualisation

Cette étape est similaire à celle de filtrage dans le filtre de Kalman. En utilisant la formule de Bayes nous avons :

$$p(x_t, y_t \mid y_1^{t-1}) = p(x_t \mid y_1^t)\, p(y_t \mid y_1^{t-1}),$$

$$p(x_t, y_t \mid y_1^{t-1}) = p(y_t \mid x_t, y_1^{t-1})\, p(x_t \mid y_1^{t-1}) = p(y_t \mid x_t)\, p(x_t \mid y_1^{t-1}).$$

De ces deux formules on déduit l'expression de $p(x_t \mid y_1^t)$ en fonction de $p(x_t \mid y_1^{t-1})$:

$$p(x_t \mid y_1^t) = \underbrace{\frac{p(y_t \mid x_t)}{p(y_t \mid y_1^{t-1})}}_{\text{terme d'actualisation}}\, p(x_t \mid y_1^{t-1})$$

Posons

$$\tilde{p}(x_t \mid y_1^t) = \frac{p(y_t \mid x_t)}{p(y_t \mid y_1^{t-1})}\, \hat{p}(x_t \mid y_1^{t-1}) = \frac{1}{N}\frac{p(y_t \mid x_t)}{p(y_t \mid y_1^{t-1})}\sum_{n=1}^{N} p(x_t \mid x_{t-1}^{(n)}),$$

alors :

$$\hat{p}(x_t \mid y_1^t) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\{x_t^{(n)}\}}(x_t),$$

où $\left\{x_t^{(1)},...,x_t^{(N)}\right\}$ est le résultat de tirage aléatoire (i.i.d) selon la loi $\tilde{p}(x_t \mid y_1^t)$.

On résume ces équations de propagation et d'actualisation dans l'algorithme suivant :

Algorithme 3.2 : Filtre Particulaire

Initialisation :

$$\omega_0^{(n)} = \frac{1}{N} \qquad \forall n \in \{1,...,N\} \quad .$$

Soit $\left\{x_0^{(1)},...,x_0^{(N)}\right\}$ un tirage (i.i.d) selon la loi $p(x_0)$.

$(n \geq 1)$

Propagation :

On tire $\left\{x_t^{(1)},...,x_t^{(N)}\right\}$, N particules selon la densité d'importance $q(x_t \mid (x_1^{t-1})^{(n)}, y_1^t)$

On pose : $(x_1^t)^{(n)} = ((x_1^{t-1})^{(n)}, x_t^{(n)})$

Correction :

Calcul des poids d'importance :

$$\tilde{\omega}_t^{(n)} = \frac{p(x_t^{(n)} \mid x_{t-1}^{(n)})\, p(y_t \mid x_t^{(n)})}{q(x_t^{(n)} \mid (x_1^{t-1})^{(n)}, y_1^t)}\, \omega_{t-1}^{(n)}$$

Normalisation des poids :

$$\omega_t^{(n)} = \frac{\tilde{\omega}_t^{(n)}}{\sum_{i=1}^{N} \tilde{\omega}_t^{(i)}}$$

On pose :

$$\hat{p}(x_t \mid y_1^t) = \sum_{n=1}^{N} \omega_t^{(n)} \delta_{\{x_t^{(n)}\}}(x_t) \quad .$$

Fin

3.4 Exemples de simulation

3.4.1 Système linéaire

On considère le système linéaire suivant :

$$\begin{cases} X_{t+1} = m_{t+1} + F\,X_t + W_{n+1} \\ Y_t = \mu_t + G\,X_t + V_t \end{cases}$$

On conserve les mêmes hypothèses du paragraphe (3.2). Pour un jeu de paramètre on va donner la restauration par filtre de Kalman $\hat{x}_0^t$ u signal caché x_0^t en observant y_0^t. On va également définir deux types d'erreurs de restauration :

$$Err_1 = \frac{1}{T}\sum_{T=1}^{T} \left| x_t - \hat{x}_t \right|, \quad Err_2 = \frac{1}{T}\sqrt{\sum_{T=1}^{T} (x_t - \hat{x}_t)^2}$$

3.4.1.1 Exemple 1 (cas de bruit gaussien)

Soit $X_0 \sim N(0,1)$, $W_t \sim N(0,\tau^2)$, et $V_t \sim N(0,\sigma^2)$.

On considère que $m_t = a_x \cos(\omega_x \frac{t\pi}{T})$ et $\mu_t = a_y \sin(\omega_y \frac{t\pi}{T})$, on simule les deux trajectoires selon le système et on restaure par filtre de Kalman (filtre optimal).

Cas $T = 500,\ \tau^2 = 0.25,\ \omega_x = 10,\ \omega_y = 5,\ a_x = a_y = 1,\ F = 0.5$ et $G = 2$

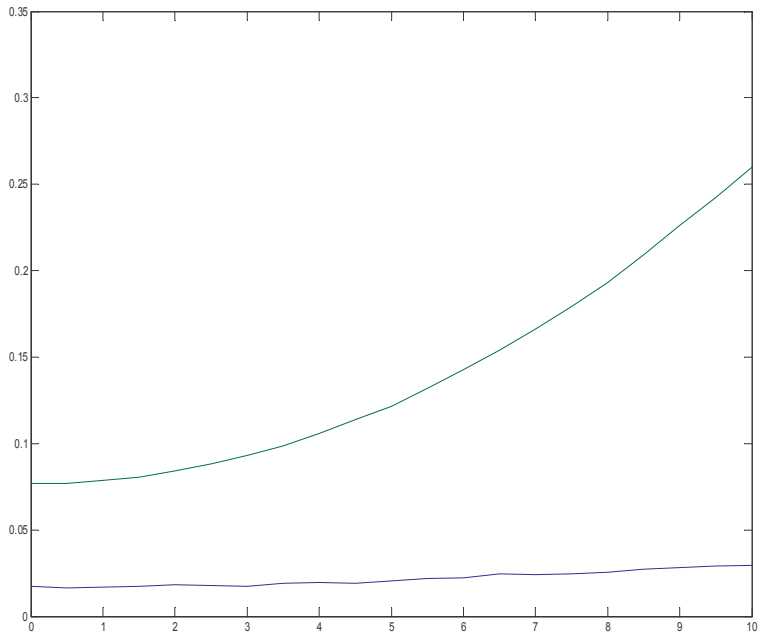


Figure 3.1 :Les courbe au-dessus représente l'évolution de l'erreur de restauration du type Err_1 en bleu et du type Err_2 en vert les deux en fonction de σ.

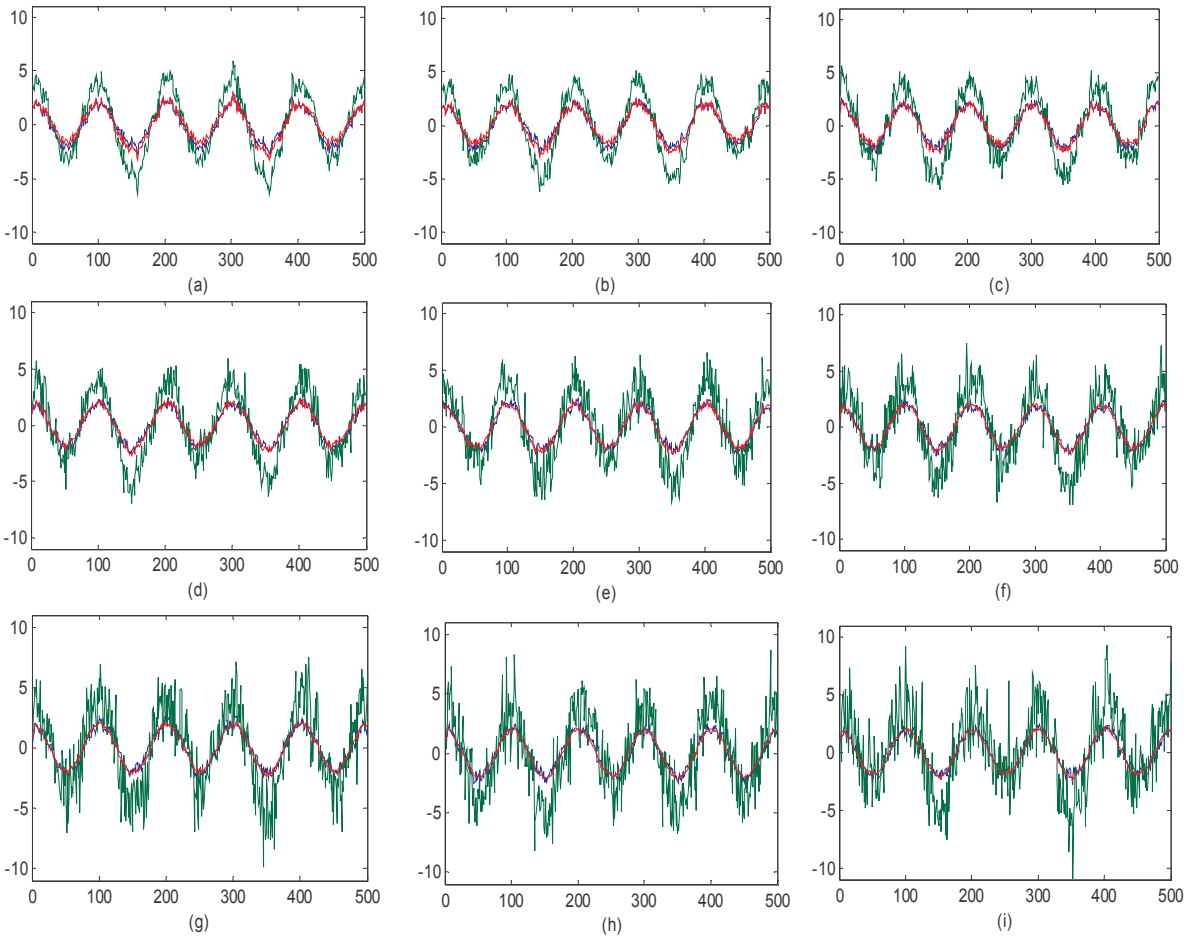


Figure 3.2 : Une simulation de trajectoire (a) pour $\sigma = 0$, (b) pour $\sigma = 0.25$, (c) pour $\sigma = 0.5$, (d) pour $\sigma = 0.75$, (e) pour $\sigma = 1$, (f) pour $\sigma = 1.25$, (g) pour $\sigma = 1.5$, (h) pour $\sigma = 1.75$ et (k) pour $\sigma = 2$. En vert le signal observé, en bleu le signal latent et en rouge la restauration par le filtre de Kalman-Bucy.

3.4.1.2 Exemple 2 (bruit non gaussien)

Soit $X_0 \sim N(0,1), W_t \sim Exp(\lambda_1), V_t \sim Exp(\lambda_2), m_t = \mu_t = 0$.

On donne un exemple de restauration par le filtre de Kalman (filtre sous-optimal)

Cas $T = 100,\ \lambda_1 = 0.5,\ \lambda_2 = 1,\ F = 0.5$ et $G = 2$

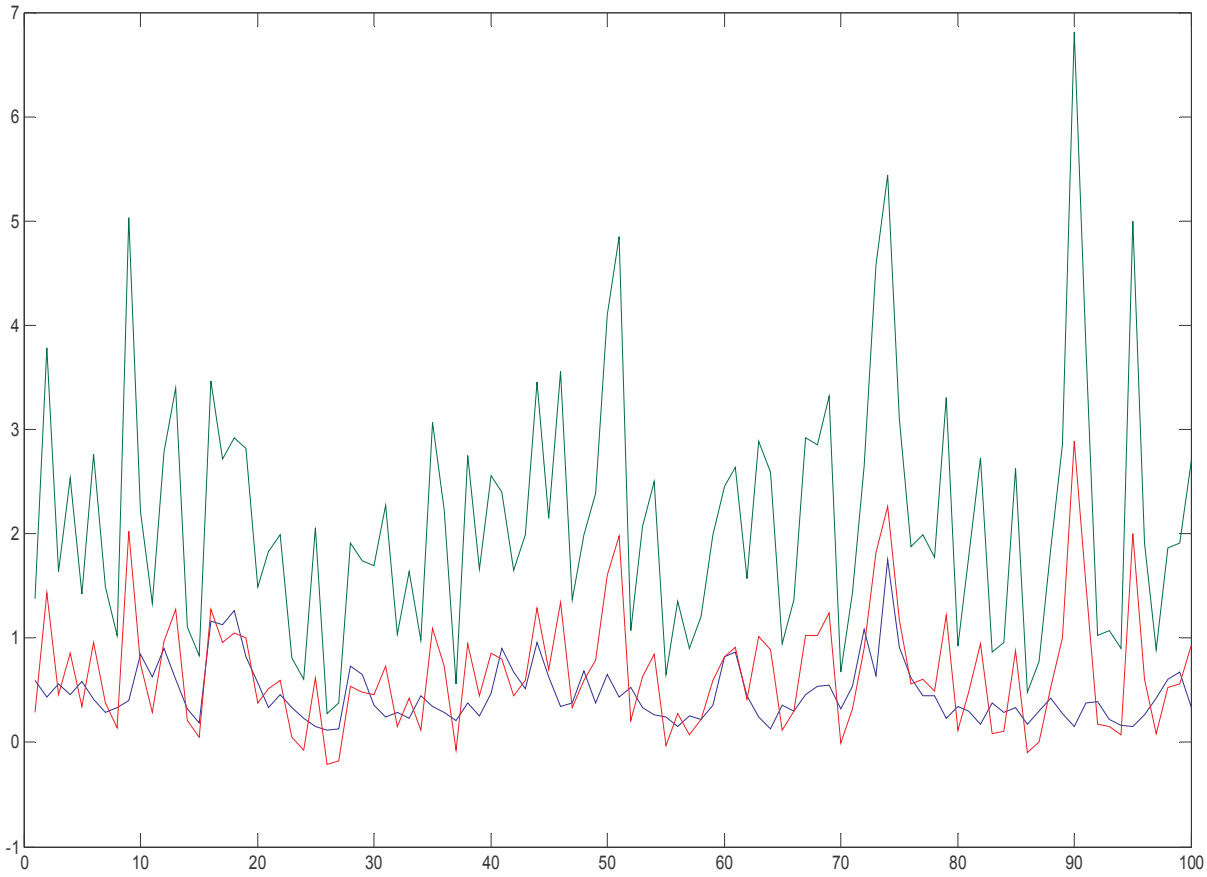


Figure 3.3 : les courbes au-dessus correspondent en vert au signal observé een bleu le vrai signal caché et en rouge la restauration donnée par le filtre de Kalman.

Conclusion

Le filtre de Kalman donne d'excellents résultats dans le cas linéaire gaussien ; cependant, dès qu'on s'éloigne de ce cas on n'a pas toujours de réponse satisfaisante avec les méthodes classiques du type Kalman linéarisé. D'autres méthodes du type « filtre particulaire » proposent une alternative intéressante, mais elles restent sensibles au bruit associé aux observations. Par ailleurs, l'approximation de la densité d'intérêt par une densité empirique discrète avec des sommes de Dirac ne semble pas toujours suffisante. Les méthodes dites « à noyaux » donnent des résultats qui semblent meilleurs. Dans le cadre du filtrage particulaire la dégénérescence des poids constituent un autre problème important.

Chapitre 4

Filtrage dans les chaînes triplets mixtes à sauts markoviens

Les modélisations avec des Chaînes de Markov Cachées (CMC), Chaînes de Markov Couple (CM-Couples), et Chaînes de Markov Triplets (CMT), décrites dans les chapitres précédents, permettent d'estimer l'état ou les caractéristiques d'un processus, à partir d'une réalisation disponible d'un autre processus dit « observé ». Dans ce chapitre on va se restreindre à l'étude d'un modèle triplet particulier où le processus d'intérêt et le processus observé sont à temps discrets mais à espace d'état continu, alors que le processus auxiliaire (processus des sauts) est une chaîne de Markov à espace d'états fini. Cette chaîne auxiliaire modélise les changements de régime (les stationnarités) interne aux deux processus étudiés.

4.1 Modèle dynamique

Dans tout ce chapitre on considère le modèle triplet $Z = (X, R, Y) = (X_t, R_t, Y_t)_{t\in\aleph}$, où les X_t prennent leurs valeurs dans $\Re^{d_X}$, les Y_t prennent leurs valeurs dans $\Re^{d_Y}$, et les R_t prennent leurs valeurs dans espace d'état Ω fini . La loi de Z est définie de la façon suivante :

$$\begin{aligned} &R = \{R_t\}_{t\in N} \text{ est une chaîne de Markov ;} \\ &X_{t+1} = F_{t+1}(R_{t+1}, X_t) + H(R_{t+1})W_{t+1} \ ; \\ &Y_t = G_t(R_t, X_t) + J(R_t)V_t, \end{aligned} \qquad (4.1)$$

où $W = (W_t)_{t\in\aleph}$, chaque W_t prenant ses valeurs dans $\Re^{d_W}$, $V = (V_t)_{t\in\aleph}$, chaque V_t prenant ses valeurs dans $\Re^{d_V}$. Les composantes des processus $W = (W_t)_{t\in\aleph}$, et $V = (V_t)_{t\in\aleph}$ sont centrés, indépendantes, mutuellement indépendantes et indépendantes de $X_0 \in \Re^{d_X}$; $F_{t+1}(R_{t+1}, X_t)$, $G_t(R_t, X_t)$, $H(R_t)$ et $J(R_t)$ sont des vecteurs et matrices variables dans le temps t en fonction des réalisations du processus d'intérêt et du processus auxiliaire.
Le graphe orienté représentant les dépendances entre les trois processus est donné à la Figure 4.1..

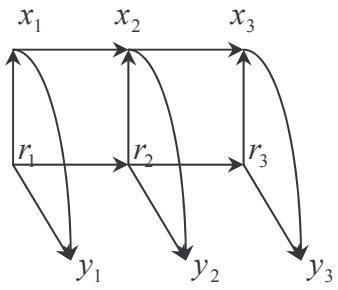


Figure 4.1 Graphe de dépendance du modèle (4.1)

Posons $S = (X, R)$, qui est donc le processus « couple » caché. De manière immédiate, vu les hypothèses sur les processus $W = (W_t)_{t\in\aleph}$ et $V = (V_t)_{t\in\aleph}$, on constate que le triplet $Z = (X, R, Y)$ est un processus de Markov.

Proposition 4.1:

Soit $Z = (X, R, Y) = (X_t, R_t, Y_t)_{t\in\aleph}$ processus triplet vérifiant les hypothèses ci-dessus. Alors

(a) Les processus S et Z sont de Markov ;

(b) $(X|R)$ de Markov;

(c) $\forall\ T \in N^*$, on a

(i) $p(x_0^T | r_0^T) = p(x_0|r_0)\prod_{t=0}^{T-1} p(x_{t+1}|x_t, r_{t+1})$

(ii) $p(y_0^T | s_0^T) = \prod_{t=0}^{T} p(y_t|s_t)$

(d) $\forall\ t \in N$, on a

(i) $p(s_{t+1}|s_t) = p(x_{t+1}|x_t, r_{t+1})p(r_{t+1}|r_t)$

(ii) $p(z_{t+1}|z_t) = p(y_{t+1}|s_{t+1})p(s_{t+1}|s_t)$

4.2 Filtrage

Le problème du filtrage dans le cadre qui nous intéresse ici consiste à estimer à chaque instant "t" la réalisation des deux variables non observées R_t et X_t, à partir des données observées y_0^t. Dans ce but, dans un premier temps on se propose de calculer la densité a posteriori dite

« de filtrage » $p(s_0^t \big| y_0^t)$. Ainsi que nous allons le préciser, il y a en effet un intérêt algorithmique à calculer de manière récursive dans le temps cette quantité.
L'utilisation classique de la règle de Bayes nous permet de montrer la récurrence suivante :

$$\forall\ t \in N\ ,\ p(s_0^{t+1} \big| y_0^{t+1}) = \frac{p(z_{t+1} \big| z_t)}{p(y_{t+1} \big| y_0^t)}\, p(s_0^t \big| y_0^t) \tag{4.2}$$

Posons $\Omega = \{\lambda_1, ..., \lambda_K\}$. Soient p_0 et $\Theta = \{\theta_{i,j}\}_{(i,j)\in\{1,...,K\}}$ respectivement la probabilité initiale et la matrice des transitions de la chaîne $(R_t)_{t\in N}$, qui sera supposée homogène. On s'intéresse au calcul récursif des trois densités marginales suivantes : $p(x_{t+1} \big|\ r_0^{t+1}, y_0^{t+1})$, $p(r_{t+1} \big|\ y_0^{t+1})$ et $p(x_{t+1} \big|\ y_0^{t+1})$, dont l'intérêt dans la résolution du problème de filtrage sera montré par la suite.

- *Calcul de* $p(x_{t+1} \big|\ r_0^{t+1}, y_0^{t+1})$

On a $\forall\ t \in N$:

$$p(x_{t+1} \big|\ r_0^{t+1}, y_0^t) = \int p(x_{t+1} \big|\ x_t, r_{t+1})\, p(x_t \big|\ r_0^t, y_0^t)\ dx_t \tag{4.3}$$

$$p(y_{t+1} \big|\ r_0^{t+1}, y_0^t) = \int p(y_{t+1} \big|\ x_{t+1}, r_{t+1})\, p(x_{t+1} \big|\ r_0^{t+1}, y_0^t)\ dx_{t+1} \tag{4.4}$$

Alors nous pouvons écrire:

$$\begin{aligned} p(x_{t+1} \big|\ r_0^{t+1}, y_0^{t+1}) &= \frac{p(x_{t+1}, y_{t+1} \big|\ r_0^{t+1}, y_0^t)}{p(y_{t+1} \big|\ r_0^{t+1}, y_0^t)} = \frac{p(y_{t+1} \big|\ x_{t+1}, r_{t+1})\, p(x_{t+1} \big|\ r_0^{t+1}, y_0^t)}{p(y_{t+1} \big|\ r_0^{t+1}, y_0^t)} \\ &= p(y_{t+1} \big| x_{t+1}, r_{t+1})\, \frac{\int p(x_{t+1} \big| x_t, r_{t+1})\, p(x_t \big| r_0^t, y_0^t)\ dx_t}{\int p(y_{t+1} \big| x_{t+1}, r_{t+1})\, p(x_{t+1} \big| r_0^{t+1}, y_0^t)\ dx_{t+1}} \end{aligned} \tag{4.5}$$

- *Calcul de* $p(r_{t+1} \big|\ y_0^{t+1})$

Nous avons $\forall\ t \in N$:

$$p(r_0^{t+1} \big|\ y_0^{t+1}) = \frac{p(r_{t+1}, y_{t+1} \big|\ r_0^t, y_0^t)}{\sum_{r_0^{t+1} \in \Omega^{t+2}} p(r_{t+1}, y_{t+1} \big|\ r_0^t, y_0^t)\, p(r_0^t \big|\ y_0^t)}\, p(r_0^t \big|\ y_0^t)$$

$$= \frac{p(y_{t+1}|r_0^{t+1}, y_0^t)p(r_{t+1}|r_t)}{\sum_{r_0^{t+1}\in\Omega^{t+2}} p(y_{t+1}|r_0^{t+1}, y_0^t)p(r_{t+1}|r_t)p(r_0^t|y_0^t)} p(r_0^t|y_0^t) \qquad (4.6)$$

Alors nous pouvons écrire:

$$p(r_{t+1}|\ y_0^{t+1}) = \sum_{r_0^t\in\Omega^{t+1}} p(r_0^{t+1}|\ y_0^{t+1}) \qquad (4.7)$$

- *Calcul de* $p(x_{t+1}|\ y_0^{t+1})$

On a $\forall\ t \in N$:

$$p(x_{t+1}|\ y_0^{t+1}) = \sum_{r_0^{t+1}\in\Omega^{t+2}} p(r_0^{t+1}|\ y_0^{t+1})p(x_{t+1}|\ r_0^{t+1}, y_0^{t+1}) \qquad (4.8)$$

Remarques

- Les égalités (4.3) et (4.4) ne sont pas toujours possibles à développer, le calcul analytique des intégrales qui s'y trouvent n'étant pas toujours possible. Cependant, on peut faire appel à des méthodes d'approximations numériques. Dans le cas particulier des systèmes linéaires gaussiens (conditionnellement aux sauts), comme on va le voir par la suite, ces quantités, qui correspondent aux équations de la partie « prédiction » dans le célèbre algorithme " filtre de Kalman", peuvent être calculées explicitement ;

- Le calcul dans (4.5) et (4.6) peut se simplifier selon la nature des deux densités définies dans (4.3) et (4.4). Dans le cas des systèmes linéaires gaussiens (conditionnellement aux sauts), cette expression permet d'extraire les équations de la partie « filtrage » de l'algorithme "filtre de Kalman".

On constate à partir de (4.7) que le calcul de la quantité d'intérêt à l'instant "$t+1$", nécessite d'effectuer une série d'opérations dont le nombre est une fonction en "K^{t+1}". Au niveau algorithmique un tel calcul représente un obstacle qui devient rapidement insurmontable lorsque t croit. Ainsi que l'on va voir par la suite, pour y remédier les méthodes de Montes Carlo séquentielles - qu'on appelle communément les méthodes de « filtrage particulaire » - peuvent permettre d'approcher convenablement ces différentes lois de probabilité tout en gardant l'aspect récursif du calcul en fonction du temps.

Finalement, le schéma du filtre st le suivant

$$\begin{bmatrix} p(r_0^t|\ y_0^t) \\ \\ p(x_t|\ r_0^t, y_0^t) \end{bmatrix} \xrightarrow{\text{prédiction}} \begin{bmatrix} p(x_{t+1}|\ r_0^{t+1}, y_0^t) \\ \\ p(y_{t+1}|\ r_0^{t+1}, y_0^t) \end{bmatrix} \xrightarrow{\text{filtrage}} \begin{bmatrix} p(r_0^{t+1}|\ y_0^{t+1}) \\ \\ p(x_{t+1}|\ r_0^{t+1}, y_0^{t+1}) \end{bmatrix}$$

4.3 Modèle gaussien

Ainsi que nous allons le voir, dans le système linéaire gaussien, qui est l'un des modèles classiques les plus simples, les différentes équations du filtre se simplifient de manière remarquable et les différentes quantités sont calculables explicitement.
On reprend ici le modèle (4.1) avec des bruits gaussiens. Ces distributions offrent un avantage d'être caractérisées uniquement par deux paramètres: la moyenne et la matrice des covariance. Les seules propriétés des vecteurs aléatoires gaussiens que l'on utilisera sont classiques et leurs preuves complètes peuvent être trouvées, par exemple, dans [38], [39].
On notera dans la suite $x' = (x^1, ..., x^{d_X})$ le transposé d'un vecteur $x \in \Re^{d_X}$.

Les processus X_0, W, et V considérés maintenant sont mutuellement indépendantes, leurs caractéristiques sont les suivantes:

- Les processus $V = \{V_t,\ T > t \geq 0\}$ et $W = \{W_t,\ T > t \geq 1\}$ sont gaussiens centrés et indépendantes de matrice de covariance respectivement:

$$C_{t+1}^W = \mathrm{E}[W_{t+1} W_{t+1}^{'}] \ , \qquad C_t^V = \mathrm{E}[V_t V_t^{'}], \text{ pour} \qquad t \geq 0$$

- X_0 est un vecteur gaussien de moyenne et de matrice de covariance :

$$m_0 \ , \qquad C_0^X = \mathrm{E}[(X_0 - m_0)(X_0 - m_0)']$$

En notant $\mathcal{N}(m, \Gamma)$ la loi gaussienne de moyenne m et de matrice de covariance Γ, on a alors:

$$p(x_{t+1} | x_t = x, r_{t+1} = \lambda] \sim \mathcal{N}(F_{t+1}(\lambda, x), H(\lambda)\, C_{t+1}^W\, H'(\lambda))$$

$$p(y_t | x_t = x, r_t = \xi) \sim \mathcal{N}(G_t(\xi, x), J(\xi)\ C_t^V\ J'(\xi))$$

On suppose que pour tout $x \in \Re^{d_X}$ on a :

$$F_{t+1}(\lambda_j, x) = A_{t+1}(\lambda_j)x + b_{t+1}(\lambda_j) \qquad \forall j \in \{1, ..., K\}$$

$$G_t(\lambda_i, x) = D_t(\lambda_i)x + e_t(\lambda_i) \qquad \forall i \in \{1, ..., K\}$$

Posons :

$$\forall t \in \mathrm{N} \qquad m_t = \mathrm{E}[X_t] \quad et \quad P_t = \mathrm{E}[(X_t - m_t)(X_t - m_t)']$$

On considère les notations suivantes :

$$\overline{m}_t(r_0^t) = \mathrm{E}[X_t \mid r_0^t, y_0^t], \text{ et } \overline{P}_t(r_0^t) = \mathrm{E}[(X_t - \overline{m}_t(r_0^t))(X_t - \overline{m}_t(r_0^t))' \mid r_0^t, y_0^t]$$

$$\widetilde{m}_{t+1}(r_0^{t+1}) = \mathrm{E}[X_{t+1} \mid r_0^{t+1}, y_0^t], et\, \widetilde{P}_{t+1}(r_0^{t+1}) = \mathrm{E}[(X_{t+1} - \widetilde{m}_{t+1}(r_0^{t+1}))(X_{t+1} - \widetilde{m}_{t+1}(r_0^{t+1}))' \mid r_0^{t+1}, y_0^t]$$
$$\widetilde{\mu}_{t+1}(r_0^{t+1}) = \mathrm{E}[Y_{t+1} \mid r_0^{t+1}, y_0^t], et\, \widetilde{\Gamma}_{t+1}(r_0^{t+1}) = \mathrm{E}[(Y_{t+1} - \widetilde{\mu}_{t+1}(r_0^{t+1}))(Y_{t+1} - \widetilde{\mu}_{t+1}(r_0^{t+1}))' \mid r_0^{t+1}, y_0^t]$$

Les équations du filtre de Kalman se calculent alors de la façon suivante.

A partir des deux formules intégrales (4.3) et (4.4) on déduit les équations de prédictions suivantes:

$$\begin{cases} \widetilde{m}_{t+1}(r_0^{t+1}) = A_{t+1}(r_{t+1})\, \overline{m}_t(r_0^t) + b_{t+1}(r_{t+1}) \\ \\ \widetilde{P}_{t+1}(r_0^{t+1}) = A_{t+1}(r_{t+1})\, \overline{P}_t(r_0^t)\, A_{t+1}^{'}(r_{t+1}) + H(r_{t+1})\, C_{t+1}^{W}\, H^{'}(r_{t+1}), \end{cases} \tag{4.10}$$

et

$$\begin{cases} \widetilde{\mu}_{t+1}(r_0^{t+1}) = D_{t+1}(r_{t+1})\, \widetilde{m}_{t+1}(r_0^{t+1}) + e_{t+1}(r_{t+1}) \\ \\ \widetilde{\Gamma}_{t+1}(r_0^{t+1}) = D_{t+1}(r_{t+1})\, \widetilde{P}_{t+1}(r_0^{t+1})\, D_{t+1}^{'}(r_{t+1}) + J(r_{t+1})\, C_{t+1}^{V}\, J^{'}(r_{t+1}) \end{cases} \tag{4.11}$$

Comme le vecteur couple $[(X_{t+1}, Y_{t+1}) \mid r_0^{t+1}, y_0^t]$ est gaussien, le vecteur aléatoire $[X_{t+1} \mid r_0^{t+1}, y_0^{t+1}]$ est également gaussien et ses paramètres sont donnés par :

$$\overline{m}_{t+1}(r_0^{t+1}) = \widetilde{m}_{t+1}(r_0^{t+1}) + C_{t+1}^{XY}\, [C_{t+1}^{YY}]^{-1}\, (y_{t+1} - \widetilde{\mu}_{t+1}(r_0^{t+1}));$$

$$\overline{P}_{t+1}(r_0^{t+1}) = \widetilde{P}_{t+1}(r_0^{t+1}) + C_{t+1}^{XY}\, [C_{t+1}^{YY}]^{-1}\, C_{t+1}^{YX},$$

avec

$$\begin{aligned} C_{t+1}^{XY} = [C_{t+1}^{YX}]' &= \mathrm{E}[\,(X_{t+1} - \widetilde{m}_{t+1}(r_0^{t+1}))\,\,(Y_{t+1} - \widetilde{\mu}_{t+1}(r_0^{t+1}))' \mid r_0^{t+1}, y_0^t] \\ &= \mathrm{E}[\,(X_{t+1} - \widetilde{m}_{t+1}(r_0^{t+1}))\,\,(X_{t+1} - \widetilde{m}_{t+1}(r_0^{t+1}))'\, D_{t+1}^{'}(r_{t+1}) \mid r_0^{t+1}, y_0^t] \\ &= \widetilde{P}_{t+1}(r_0^{t+1}) D_{t+1}^{'}(r_{t+1}), \end{aligned}$$

et

$$\begin{aligned} C_{t+1}^{YY} &= \mathrm{E}[(Y_{t+1} - \widetilde{\mu}_{t+1}(r_0^{t+1}))\,\,(Y_{t+1} - \widetilde{\mu}_{t+1}(r_0^{t+1}))' \mid r_0^{t+1}, y_0^t] \\ &= \widetilde{\Gamma}_{t+1}(r_0^{t+1}). \end{aligned}$$

Alors nous avons:

$$\begin{cases} \overline{m}_{t+1}(r_0^{t+1}) = \widetilde{m}_{t+1}(r_0^{t+1}) + \widetilde{P}_{t+1}(r_0^{t+1})\, D'_{t+1}(r_{t+1})\ [\,\widetilde{\Gamma}_{t+1}(r_0^{t+1})\,]^{-1}\ (y_{t+1} - \widetilde{\mu}_{t+1}(r_0^{t+1})\,) \\ \overline{P}_{t+1}(r_0^{t+1}) = \widetilde{P}_{t+1}(r_0^{t+1}) \ - \ \widetilde{P}_{t+1}(r_0^{t+1})\, D'_{t+1}(r_{t+1})\ [\,\widetilde{\Gamma}_{t+1}(r_0^{t+1})\,]^{-1}\ D_{t+1}(r_{t+1})\ \left(\widetilde{P}_{t+1}\right)'(r_0^{t+1}) \end{cases} \tag{4.12}$$

Si le processus auxiliaire est une suite déterministe prédéfinie, ou aléatoire mais dont la réalisation est accessible (observée), alors (4.10), (4.11) et (4.12) correspondent respectivement aux étapes de prédiction et de filtrage dans l'algorithme du filtre de Kalman classique. Pour une séquence $\{r_t\}_{t=0}^{T}$ fixée, la solution donnée par le filtre est optimale (au sens de l'erreur quadratique moyenne).

Dans le cas où la réalisation du processus auxiliaire est latente (cachée), on a besoin de calculer à chaque instant les probabilités $p(r_0^t \big| y_0^t)$, ou du moins donner une approximation convenable. Utiliser directement la récursivité proposée dans (4.6) n'est pas possible, car cela consiste à faire tourner l'algorithme de Kalman pour toutes les séquences possibles de la suite de variables R_0^T, c'est-à-dire les K^{T+1} possibilités. Rapidement, lorsque T croit, cela rend cette expression inexploitable (le nombre d'opérations et l'espace de stockage nécessaire au bon fonctionnement de l'algorithme croissent exponentiellement), et ceci même dans un cas simple, celui des systèmes linéaires gaussiens.

Nous proposons d'approcher cette densité en utilisant deux méthodes de type différent. La première approximation, classique, sera donnée par l'algorithme de filtrage particulaire. La seconde, qui est nouvelle et qui constitue un apport original de la présente thèse, est fondée sur une exploration sous contrôle de l'arbre déterministe représentant toutes les réalisations possible de la chaîne R. Comme on va le détailler par la suite, ce dernier algorithme consiste en élimination de toutes les séquences qui contribuent faiblement à la construction de la densité $p(r_0^t \big|\ y_0^t)$. Cette élimination va utiliser un seuillage des probabilités à chaque niveau de l'arbre (à chaque instant "t").

On commencera par rappeler brièvement le principe du filtrage particulaire dans le cadre du présent modèle.

4.4 Filtrage particulaire

La méthode de filtrage particulaire consiste, dans le cadre du modèle considéré, à proposer un estimateur convenable pour la densité $p(r_0^t \big| y_0^t)$ sous la forme d'une somme de masses de Dirac pondérée [2], [6], [10]. On approche donc la loi discrète $p(.\big| y_0^t)$ par la loi empirique obtenue à partir de N réalisations d'une variable selon la loi discrète que l'on cherche à approcher. Nous noterons :

$$\hat{p}_N(r_0^t \big| y_0^t) = \tfrac{1}{N} \sum_{n=1}^{N} \delta_{\{r_{0:t}^{(n)}\}}(r_{0:t}) \tag{4.13}$$

Cette approximation se justifie par la loi des grands nombres : lorsque N tend vers l'infini, la suite $\hat{p}_N(.|y_0^t)$ tend (en loi) vers $p(.|y_0^t)$. Le support de $\hat{p}_N(.|y_0^t)$ sont les « particules » $\{r_{1:t}^n\}_{n=1}^N$; propager une telle approximation dans le temps de manière récursive nous impose de choisir le support de cette densité de façon cumulative au cours du temps, c'est-à-dire :

$$\{r_{0:t+1}^{(n)}\}_{n=1}^N = \{r_{0:t}^{(n)}\}_{n=1}^N \cup \{r_{t+1}^{(n)}\}_{n=1}^N \tag{4.14}$$

D'autre part on a :

$$p(r_0^{t+1}|\ y_0^{t+1}) = \frac{p(y_{t+1}|\ r_0^{t+1}, y_0^t)\, p(r_{t+1}|\ r_t)}{p(y_{t+1}|\ y_0^t)}\, p(r_0^t|\ y_0^t) \tag{4.15}$$

Cette équation est la formule récursive principale pour propager la densité d'intérêt.
Cependant, la simulation directe selon $p(r_0^t|y_0^t)$ est rapidement impossible lorsque t croît ; en effet, la condition de récursivité (4.14) n'est pas vérifiée. Le problème est contourné de la manière suivante. Il est possible de simuler des particules selon n'importe quelle loi q sur Ω^{t+1}, à condition de multiplier (4.13) par $\frac{p(r_0^t|y_0^t)}{q(r_0^t)}$. En effet, lorsque les particules sont simulées selon q et lorsque N tend vers l'infini, la suite $\frac{p(r_0^t|y_0^t)}{Nq(r_0^t)}\sum_{n=1}^N \delta_{\{r_{0:t}^{(n)}\}}(r_{0:t})$ tend également (en loi) vers $p(.|y_0^t)$. Le problème est alors de choisir q, qui sera appelée « densité d'importance », de manière, d'une part, à ce que q soit le plus « proche » possible de $p(.|y_0^t)$ d'une part, et qu'elle vérifie (4.14), d'autre part. Pour satisfaire la première condition on prend alors q qui dépend des y_0^t ; on pose $q(.) = q(.|y_0^t)$. La deuxième condition est remplie dès que $q(.|y_0^t)$ vérifie

$$q(r_0^t|y_0^{t+1}) = q(r_0^t|y_0^t), \tag{4.16}$$

ce qui est équivalent au fait que la densité $q(r_0^{t+1}|y_0^{t+1})$ admet $q(r_0^t|y_0^t)$ comme distribution marginale. Cela est vrai si nous restreignons notre choix de densité d'importance aux fonctions qui se factorisent sous la forme suivante:

$$q(r_0^{t+1}|\ y_0^{t+1}) = q(r_0|\ y_0)\prod_{s=1}^{t+1} q(r_s|\ r_0^{s-1}, y_0^s) = q(r_{t+1}|\ r_0^t, y_0^{t+1})\, q(r_0^t|\ y_0^t) \tag{4.17}$$

Comme noté ci-dessus, les particules simulées selon $q(.|y_0^t)$ doivent être pondérées par « les poids de pondérations » données par

$$\omega(r_0^{t+1}) = \frac{p(r_0^{t+1}\left| y_0^{t+1}\right.)}{q(r_0^{t+1}\left| y_0^{t+1}\right.)} \tag{4.18}$$

On déduit de (4.14) et (4.17) la récurrence suivante:

$$\omega(r_0^{t+1}) = \frac{p(y_{t+1}\left|\ r_0^{t+1}, y_0^t\right.)p(r_{t+1}\left|\ r_t\right.)}{p(y_{t+1}\left|\ y_0^t\right.)q(r_{t+1}\left|\ r_0^t, y_0^{t+1}\right.)} \times \omega(r_0^t) = \omega_{t+1} \times \omega(r_0^t) = \prod_{s=0}^{t+1} \omega_s \ , \tag{4.19}$$

avec :

$$\omega_{t+1} = \frac{p(y_{t+1}\left|\ r_0^{t+1}, y_0^t\right.)p(r_{t+1}\left|\ r_t\right.)}{p(y_{t+1}\left|\ y_0^t\right.)q(r_{t+1}\left|\ r_0^t, y_0^{t+1}\right.)}, \text{ et } \omega_0 = \frac{p(y_0\left|\ r_0\right.)p(r_0)}{p(y_0)q(r_0\left|\ y_0\right.)}.$$

Dans le but de faire disparaître le terme $p(y_{t+1}\left|\ y_0^t\right.)$, une normalisation des poids est suffisante.

Finalement, pour obtenir $\left\{\tilde{r}_{0:\,t+1}^{(n)}\right\}_{n=1}^N$ à partir de $\left\{r_{0:\,t}^{(n)}\right\}_{n=1}^N$ on tire N particules (i.i.d) selon la loi.

$$\forall\, n \in \{1, ..., N\} \qquad \tilde{r}_{t+1}^{(n)} \sim q(r_{t+1}\left|\ (r_0^t)^{(n)}, y_0^{t+1}\right.),$$

et on pose :

$$\left\{\tilde{r}_{0:\,t+1}^{(n)}\right\}_{n=1}^N = \left\{r_{0:\,t}^{(n)}\right\}_{n=1}^N \bigcup \left\{\tilde{r}_{t+1}^{(n)}\right\}_{n=1}^N.$$

D'autre part on a $\omega((r_0^t)^{(n)}) = \frac{1}{N}$ pour tout n dans $\{1, ..., N\}$. Les pondérations $\varpi_{t+1}^{(i)}$ sont alors obtenues, pour tout i dans $\{1, ..., N\}$, par:

$$\begin{aligned}\varpi_{t+1}^{(i)} = \omega((\tilde{r}_0^{t+1})^{(i)}) &= \frac{\omega_{t+1}^{(i)}}{\sum_{n=1}^N \omega_{t+1}^{(n)}} \\ &= \frac{p(y_{t+1}\left|\ \tilde{r}_{t+1}^{(i)}, (r_0^t)^{(i)}, y_0^t\right.)p(\tilde{r}_{t+1}^{(i)}\left|\ r_t^{(i)}\right.)}{q(\tilde{r}_{t+1}^{(i)}\left|\ (r_0^t)^{(i)}, y_0^{t+1}\right.)\sum_{n=1}^N \frac{p(y_{t+1}\left|\ \tilde{r}_{t+1}^{(n)}, (r_0^t)^{(n)}, y_0^t\right.)p(\tilde{r}_{t+1}^{(n)}\left|\ r_t^{(n)}\right.)}{q(\tilde{r}_{t+1}^{(n)}\left|\ (r_0^t)^{(n)}, y_0^{t+1}\right.)}}\end{aligned}$$

On pose :

$$\tilde{p}_N(r_{t+1}\left|\ y_0^{t+1}\right.) = \sum_{n=1}^N \varpi_{t+1}^{(n)}\ \delta_{\{(\tilde{r}_{t+1}^{(n)})\}}(dr_{t+1})$$

On tire les particules support $\left\{(r_{t+1})^{(n)}\right\}_{n=1}^N$ suivant la densité marginale $\tilde{p}_N(r_{t+1}\left|\ y_0^{t+1}\right.)$.

On en déduit l'approximation de la densité de filtrage à l'instant "$(t+1)$" :

$$\hat{p}_N(r_0^{t+1} \mid y_0^{t+1}) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\{(r_0^{t+1})^{(n)}\}}(dr_0^{t+1}) \ ,$$

avec :

$$\left\{ r_{0:\,t+1}^{(n)} \right\}_{n=1}^{N} = \left\{ r_{0:\,t}^{(n)} \right\}_{n=1}^{N} \cup \left\{ r_{t+1}^{(n)} \right\}_{n=1}^{N}$$

Remarque

Le choix de la densité d'importance doit se faire de telle manière que les poids d'importance non normalisés soient bornés [5], [6]. La densité la plus convenable doit tenir compte de l'information apportée par toutes les observations jusqu'à l'instant présent. Justement dans notre modèle un tel choix optimal implique que l'on fasse des tirages à l'instant "$(t+1)$", selon la distribution $p(r_{t+1} \mid r_t^{(n)}, y_{t+1})$. Un choix plus simple pourrait être le tirage selon $p(r_{t+1} \mid r_t^{(n)})$, choix qui serait donc sous optimal car il ne tient pas compte de l'observation à l'instant présent y_{t+1}.

Un autre problème lié à la dégénérescence (décroissance vers zéro) des poids d'importance peut être observé lorsque le temps augmente. Pour y remédier il y a d'abord le choix de la densité d'importance qui doit être le plus optimal possible. Sinon il y a des méthodes dites de rééchantillonnage, dont l'idée principale consiste à éliminer les particules de faibles poids et à augmenter celle d'un poids plus grand.

4.5 Approximations fondées sur l'exploration de l'arbre des réalisations.

Nous présentons ci-après une famille d'algorithmes originale de filtrage qui permet de construire des approximations de la densité $p(r_0^t \mid y_0^t)$ de manière récursive dans le temps.

4.5.1 Filtrage par l'exploration de l'arbres des réalisations

Le problème est de donner $\hat{p}(r_0^t \mid y_0^t)$ à partir de $\hat{p}(r_0^{t-1} \mid y_0^{t-1})$. Nous proposons trois méthodes de généralité croissante.

Soit $\hat{p}(r_0^{t-1} \mid y_0^{t-1})$.

Méthode 1

Considérons la suite (L_t) donnée par

$$L_0(r_0) = p(y_0|r_0)p(r_0)\text{, et } L_t(r_t) = \sum_{r_0^{t-1}\in\Omega^t} p(y_t|r_0^t, y_0^{t-1})p(r_t|r_{t-1})\hat{p}(r_0^{t-1}|y_0^{t-1}),$$

et la probabilité $(\hat{\alpha}_t)$ définie sur Ω par

$$\hat{\alpha}_t(\lambda) = \frac{L_t(\lambda)}{\sum_{\omega\in\Omega} L_t(\omega)}.$$

Considérons la suite d'ensembles suivante:

$$\forall t \in \aleph \qquad \Lambda_t^{(\max)} = \left\{ \tilde{\omega} \in \Omega \ / \ \tilde{\omega} = \arg\max_{\omega\in\Omega} \ [\hat{\alpha}_t(\omega)] \ \right\},$$

et

$$\overline{p}(r_t|\ y_0^t) = \frac{1}{Card(\Lambda_t^{(\max)})}\delta_{\Lambda_t^{(\max)}}(r_t)$$

On pose alors

$$\hat{p}(r_0^t|\ y_0^t) = \prod_{j=0}^{t} \overline{p}(r_j|\ y_0^j)\text{, et on a } \hat{p}(r_t|\ y_0^t) = \hat{\alpha}_t(r_t)$$

Méthode 2

On considère les mêmes suites (L_t), $(\hat{\alpha}_t)$ que dans la Méthode 1. Soit $q \in [0,1]$ fixé. On définit la suite d'ensembles suivante:

$$\forall t \in \aleph, \qquad \Lambda_t^{(q)} = \left\{ \omega \in \Omega \ / \ \hat{\alpha}_t(\omega) \geq q \ \right\}$$

et

$$\overline{p}(r_t|\ y_0^t) = \frac{\hat{\alpha}_t(r_t)}{\sum_{\omega\in\Lambda_t^{(q)}} \hat{\alpha}(\omega)}\ \delta_{\Lambda_t^{(q)}}(r_t)$$

On pose alors

$$\hat{p}(r_0^t|\ y_0^t) = \prod_{j=0}^{t} \overline{p}(r_j|\ y_0^j),\ \text{et on a } \hat{p}(r_t|\ y_0^t) = \hat{\alpha}_t(r_t)$$

Méthode 3

Considérons la suite (L_t) donnée par

$$L_0(r_0) = p(y_0|r_0)p(r_0),\ L_t(r_0^t) = p(y_t|r_0^t, y_0^{t-1})p(r_t|r_{t-1})\hat{p}(r_0^{t-1}|y_0^{t-1})$$

et la probabilité $(\hat{\alpha}_t)$ définie sur Ω^{t+1} par

$$\forall \lambda_0^t \in \Omega^{t+1}, \qquad \hat{\alpha}_t(\lambda_0^t) = \frac{L_t(\lambda_0^t)}{\sum_{\omega_0^t \in \Omega^{t+1}} L_t(\omega_0^t)}$$

Soit $q_t \in [0,1]$ une suite déterministe décroissante qui vérifie $(K \times q_t \leq q_{t-1})$. On définit la suite d'ensemble suivante:

$$\forall t \in \aleph, \qquad \Xi_t^{(q)} = \left\{ \omega_0^t \in \Omega^{t+1} / \hat{\alpha}_t(\omega_0^t) \geq q_t \right\}$$

et

$$\overline{p}(r_0^t | y_0^t) = \frac{\hat{\alpha}_t(r_0^t)}{\sum_{\omega_0^t \in \Xi_t^{(q)}} \hat{\alpha}(\omega_0^t)} \delta_{\Xi_t^{(q)}}(r_0^t),$$

On pose alors

$$\hat{p}(r_0^t | y_0^t) = \hat{\alpha}_t(r_0^t), \text{ et on a } \hat{p}(r_t | y_0^t) = \sum_{r_0^{t-1} \in \Omega^t} \overline{p}(r_0^t | y_0^t)$$

Ainsi dans les trois méthodes détaillées au-dessus on considère les approximations $p(r_t | y_0^t) \approx \hat{p}(r_t | y_0^t)$ et $p(r_0^t | y_0^t) \approx \hat{p}(r_0^t | y_0^t)$, fondées sur une exploration contrôlée de l'arbre de toutes les réalisations possibles de la chaîne. En d'autres termes, elles sont fondées sur l'élimination des trajectoires les moins probables en progressant dans le temps selon la contrainte choisie. A chaque instant la taille de l'espace d'état dépend du critère de sélection et du temps.

Le filtrage particulaire est également fondé sur l'exploration de l'arbre des réalisations possibles de la chaîne [20], [21], mais cette exploration se fait à travers des tirages aléatoires. Généralement dans ce type de technique la taille du support de la densité (nombre de tirages effectués à chaque instant) est fixée au début de l'algorithme, contrairement aux méthodes alternatives que nous proposons ci-dessus, où la taille du support est variable est dépend de la contrainte choisie. Dans ce cas, comme on va le voir, le support peut être plus riche.

Des simulations comparant la Méthode 1 avec le filtrage particulaire sont présentées dans la sous-section suivante.

L'expression (4.8) nous permet de déduire les paramètres de la densité $p(x_t | y_0^t)$ de la loi a posteriori, à savoir la moyenne et la variance. En posant

$$m_{t|t} = \mathrm{E}[X_t | y_0^t], \text{ et } P_{t|t} = \mathrm{E}[(X_t - m_{t|t})(X_t - m_{t|t})' | y_0^t],$$

nous avons:

$$\begin{cases} m_{t|t} = \sum_{r_0^t \in \Omega^{t+1}} p(r_0^t \mid y_0^t)\, \overline{m}_t(r_0^t) \\ P_{t|t} = \sum_{r_0^t \in \Omega^{t+1}} p(r_0^t \mid y_0^t)\, (\overline{P}_t(r_0^t) + \overline{m}_t(r_0^t)\, \overline{m}_t(r_0^t)') - m_{t|t}\, m_{t|t}' \end{cases} \qquad (4.20)$$

On propose alors deux estimateurs de ces quantités ; le premier en utilisant l'approximation particulaire et le second utilisant le nouvel algorithme, correspondant à la Méthode 1.

Dans l'approximation particulaire on a

$$p(x_t \mid y_0^t) \approx \sum_{r_0^t \in \Omega^{t+1}} \hat{p}_N(r_0^t \mid y_0^t)\, p(x_t \mid r_0^t, y_0^t) = \frac{1}{N} \sum_{n=1}^{N} p(x_t \mid (r_0^t)^{(n)}, y_0^t),$$

ce qui donne

$$\begin{cases} \hat{m}_{t|t}^{(1)} = \dfrac{1}{N} \sum_{n=1}^{N} \overline{m}_t((r_0^t)^{(n)}) \\ \hat{P}_{t|t}^{(1)} = \dfrac{1}{N} \sum_{n=1}^{N} (\overline{P}_t((r_0^t)^{(n)}) + \overline{m}_t((r_0^t)^{(n)})\, \overline{m}_t((r_0^t)^{(n)})') - \hat{m}_{t|t}^{(1)}\, (\hat{m}_{t|t}^{(1)})' \end{cases}$$

Dans l'approximation fondée sur l'exploration déterministe sous contrainte on a donc la démarche suivante. Soit $q \in [0,\ 1]$ et, pour tout t dans $\aleph$, $\Delta_t^{(q)} = \bigotimes_{j=0}^{t} \Lambda_j^{(q)}$. Alors

$$p(x_t \mid y_0^t) \approx \sum_{r_0^t \in \Omega^{t+1}} \hat{p}(r_0^t \mid y_0^t)\, p(x_t \mid r_0^t, y_0^t) = \sum_{r_0^t \in \Delta_t^{(q)}} \hat{p}(r_0^t \mid y_0^t)\, p(x_t \mid r_0^t, y_0^t),$$

ce qui donne

$$\begin{cases} \hat{m}_{t|t}^{(2)} = \sum_{r_0^t \in \Delta_t^{(q)}} \hat{p}(r_0^t \mid y_0^t)\, \overline{m}_t(r_0^t) \\ \hat{P}_{t|t}^{(2)} = \sum_{r_0^t \in \Delta_t^{(q)}} \hat{p}(r_0^t \mid y_0^t)\, (\overline{P}_t(r_0^t) + \overline{m}_t(r_0^t)\, \overline{m}_t(r_0^t)') - \hat{m}_{t|t}^{(2)}\, (\hat{m}_{t|t}^{(2)})' \end{cases}$$

Rappelons le schéma du filtre utilisant le filtrage particulaire :

$$\begin{bmatrix} p(r_0^t \mid y_0^t) \\ p(x_t \mid r_0^t, y_0^t) \end{bmatrix} \xrightarrow{\text{prédiction}} \begin{bmatrix} p(x_{t+1} \mid r_0^{t+1}, y_0^t) \\ \downarrow \\ p(y_{t+1} \mid r_0^{t+1}, y_0^t) \end{bmatrix} \xrightarrow{\text{filtrage}} \begin{bmatrix} p(r_0^{t+1} \mid y_0^{t+1}) \\ p(x_{t+1} \mid r_0^{t+1}, y_0^{t+1}) \end{bmatrix}$$

L'estimation des sauts cachés est alors effectuée par:

$$\hat{r}_t = \arg\max_{r_t \in \Omega}([p(r_t \mid y_0^t)]),$$

et la restauration du signal caché à l'instant t sera effectuée par deux estimateurs différents suivants:

$$\hat{x}_t^{(1)} = E[X_t \mid \hat{r}_0^t, y_0^t] = \overline{m}_t(\hat{r}_0^t),$$

ou

$$\hat{x}_t^{(2)} = E[X_t \mid y_0^t] = \sum_{r_0^t \in \Omega^{t+1}} p(r_0^t \mid y_0^t)\, E[X_t \mid r_0^t, y_0^t] = \sum_{r_0^t \in \Omega^{t+1}} p(r_0^t \mid y_0^t)\, \overline{m}_t(r_0^t)$$

Pour $X_0 \sim \mathcal{N}(m_0, P_0)$ et $p(r_0) = p_0$ donnés on va décrire les différents algorithmes.

Algorithme 1 (filtre particulaire)

$$(t = 0) \quad \textbf{(Initialisation)}$$

Etape (0)

➔ On tire N particules selon la distribution initiale $q(r_0) = p_0$.

➔ On pose : $\forall\, i \in \{1,...,N\} \quad \varpi_0^{(i)} = \dfrac{1}{N}$.

$$(t > 0) \quad \textbf{(Récurrence)}$$

Etape (1) (*Filtrage Particulaire SIR*)

➔ Pour $i \in \{1,...,N\}$, on fait un tirage ***(i.i.d)*** (*actualisation)*

$$\tilde{r}_t^{(i)} \sim q(r_t \mid r_{1:t-1}^{(i)}, y_{1:t-1}) \quad \text{On pose alors :} \quad \tilde{r}_{1:t}^{(i)} = \left\{ r_{1:t-1}^{(i)}, \tilde{r}_t^{(i)} \right\}$$

➔ Pour $i \in \{1,...,N\}$, on calcule les poids nécessaire à l'actualisation à l'étape $'t'$:

$$\tilde{\varpi}_t^{(i)} \propto \frac{p(y_t \mid \tilde{r}_{0:t}^{(i)}, y_0^{t-1})\, p(\tilde{r}_t^{(i)} \mid r_{t-1}^{(i)})}{q(r_t \mid r_{0:t-1}^{(i)}, y_0^t)} \quad \text{on normalise les poids } \varpi_t^{(i)} = \frac{\tilde{\varpi}_t^{(i)}}{\sum_{j=1}^{N} \tilde{\varpi}_t^{(j)}}$$

Les paramètres de la densité gaussienne $p(y_t \mid \tilde{r}_{1:t}^{(i)}, y_{1:t-1})$ sont donnés par les équations de prédictions de l'algorithme du filtre de Kalman).

➔ Pour $i \in \{1,...,N\}$, on fait un tirage ***(i.i.d)*** (*ré échantillonnage*)

$$r_t^{(i)} \sim \widetilde{p}_N(r_t \mid y_{1:t}) = \sum_{n=1}^{N} \varpi_t^{(i)} \delta_{r_t^{(i)}}$$

alors

$$\begin{cases} r_{1:t}^{(i)} = \left\{ r_{1:t-1}^{(i)}, r_t^{(i)} \right\} \\ p(r_t \mid y_{1:t}) \approx \hat{p}_N(r_t \mid y_{0:t}) = \frac{1}{N} \sum_{n=1}^{N} \delta_{\{(r_t^{(n)})\}}(dr_t) \end{cases}$$

Etape (2) (*Filtrage de Kalman*)

➔ Pour $i \in \{1,...,N\}$ on calcul les paramètres des densités $p(x_t \mid r_{0:t}^{(i)}, y_{0:t-1})$, $p(y_t \mid r_{0:t}^{(i)}, y_{0:t-1})$ et $p(x_t \mid r_{0:t}^{(i)}, y_{0:t})$ respectivement par filtre de Kalman

Etape (3) (*Estimation d'état caché*)

➔ Pour le processus des sauts on a $p(r_t \mid y_{0:t}) \approx \hat{\alpha}_t^{(1)}(r_t) = \hat{p}_N(r_t \mid y_{0:t})$ et alors:

$$\hat{r}_t = \arg\max_{r_t} \left\{ \hat{\alpha}_t^{(1)}(r_t) \right\}$$

➔ Pour le processus d'intérêt on peut proposer deux estimateurs possibles:

$$\hat{x}_t^{(2)} = \frac{1}{N} \sum_{n=1}^{N} E[X_t \mid r_{0:t}^{(n)}, y_{0:t}] = \frac{1}{N} \sum_{n=1}^{N} \hat{m}_t^{(1)}(r_{0:t}^{(n)}) \quad et \quad \hat{x}_t^{(1)} = E[X_t \mid \hat{r}_{0:t}^{(n)}, y_{0:t}] = \hat{m}_t^{(1)}(\hat{r}_{0:t}^{(n)})$$

$t = t + 1$; Retour à l'étape 1.

Fin.

Les équations du filtre fondé sur la Méthode 1

Le schéma de ce filtre est le suivant:

$$\begin{bmatrix} \hat{r}_t = \arg\max_{r_t} (\hat{\alpha}_t^{(2)}(r_t)) \\ \downarrow \\ p(x_t \mid \hat{r}_0^t, y_0^t) \end{bmatrix} \xrightarrow{prédiction} \begin{bmatrix} p(x_{t+1} \mid r_{t+1}, \hat{r}_0^t, y_0^t) \\ \downarrow \\ p(y_{t+1} \mid r_{t+1}, \hat{r}_0^t, y_0^t) \\ \downarrow \\ p(r_{t+1} \mid y_0^{t+1}) \approx \hat{\alpha}_{t+1}^{(2)}(r_{t+1}) \end{bmatrix} \xrightarrow{filtrage} \begin{bmatrix} \hat{r}_{t+1} = \arg\max_{r_{t+1}} (\hat{\alpha}_{t+1}^{(2)}(r_{t+1})) \\ \downarrow \\ p(x_{t+1} \mid \hat{r}_0^{t+1}, y_0^{t+1}) \end{bmatrix}$$

On considère les notations suivantes:

$$\widetilde{m}_{t+1}^{(2)}(r_{t+1}) = \mathrm{E}[X_{t+1} \mid r_{t+1}, \hat{r}_0^t, y_0^t] \quad et \quad \widetilde{P}_{t+1}^{(2)}(r_{t+1}) = \mathrm{E}[(X_{t+1} - \widetilde{m}_{t+1}^{(2)}(r_{t+1}))(X_{t+1} - \widetilde{m}_{t+1}^{(2)}(r_{t+1}))' \mid r_{t+1}, \hat{r}_0^t, y_0^t]$$

$$\widetilde{\mu}_{t+1}^{(2)}(r_{t+1}) = \mathrm{E}[Y_{t+1} \mid r_{t+1}, \hat{r}_0^t, y_0^t] \quad et \quad \widetilde{\Gamma}_{t+1}^{(2)}(r_{t+1}) = \mathrm{E}[(Y_{t+1} - \widetilde{\mu}_{t+1}^{(2)}(r_{t+1}))(Y_{t+1} - \widetilde{\mu}_{t+1}^{(2)}(r_{t+1}))' \mid r_{t+1}, \hat{r}_0^t, y_0^t]$$

$$\hat{m}_{t+1}^{(2)} = \mathrm{E}[X_{t+1} \mid \hat{r}_0^{t+1}, y_0^{t+1}] \quad et \quad \hat{P}_{t+1}^{(2)} = \mathrm{E}[(X_{t+1} - \hat{m}_{t+1}^{(2)})(X_{t+1} - \hat{m}_{t+1}^{(2)})' \mid \hat{r}_0^{t+1}, y_0^{t+1}]$$

Les équations de prédiction et de filtrage sont les suivantes:

Prédiction:

$$\begin{cases} \widetilde{m}_{t+1}^{(2)}(r_{t+1}) = A(r_{t+1})\, \hat{m}_t^{(2)} + b_{t+1}(r_{t+1}) \\ \\ \widetilde{P}_{t+1}^{(2)}(r_{t+1}) = A(r_{t+1})\, \hat{P}_t^{(2)}\, A'(r_{t+1}) + H(r_{t+1})\, C_{t+1}^{W}\, H'(r_{t+1}), \end{cases} \tag{4.21}$$

et

$$\begin{cases} \widetilde{\mu}_{t+1}^{(2)}(r_{t+1}) = D(r_{t+1})\, \widetilde{m}_{t+1}^{(2)}(r_{t+1}) + e_{t+1}(r_{t+1}) \\ \\ \widetilde{\Gamma}_{t+1}^{(2)}(r_{t+1}) = D(r_{t+1})\, \widetilde{P}_{t+1}^{(2)}(r_{t+1})\, D'(r_{t+1}) + J(r_{t+1})\, C_{t+1}^{V}\, J'(r_{t+1}) \end{cases} \tag{4.22}$$

Filtrage:

$$p(r_{t+1} \mid y_0^{t+1}) \approx \hat{\alpha}_{t+1}^{(2)}(r_{t+1}) = \frac{p(r_{t+1} \mid \hat{r}_t)\, p(y_{t+1} \mid r_{t+1}, \hat{r}_0^t, y_0^t)}{\sum_{r_{t+1} \in \Omega} p(r_{t+1} \mid \hat{r}_t)\, p(y_{t+1} \mid r_{t+1}, \hat{r}_0^t, y_0^t)} \tag{4.23}$$

$$\begin{cases} \hat{m}_{t+1}^{(2)} = \widetilde{m}_{t+1}^{(2)}(\hat{r}_{t+1}) + \widetilde{P}_{t+1}^{(2)}(\hat{r}_{t+1})\, D'(\hat{r}_{t+1})\, [\widetilde{\Gamma}_{t+1}^{(2)}(\hat{r}_{t+1})]^{-1}\, (y_{t+1} - \widetilde{\mu}_{t+1}^{(2)}(\hat{r}_{t+1})) \\ \\ \hat{P}_{t+1}^{(2)} = \widetilde{P}_{t+1}^{(2)}(\hat{r}_{t+1}) - \widetilde{P}_{t+1}^{(2)}(\hat{r}_{t+1})\, D'(\hat{r}_{t+1})\, [\widetilde{\Gamma}_{t+1}^{(2)}(\hat{r}_{t+1})]^{-1}\, D(\hat{r}_{t+1})\, \left(\widetilde{P}_{t+1}^{(2)}\right)'(\hat{r}_{t+1}) \end{cases} \tag{4.24}$$

Algorithme 2 (Méthode 1)

$(t = 0)$ **(Initialisation)**

Etape (0)

➔ $\forall\, k \in \{1, ..., K\}$ on calcul les paramètres des densités $\psi_0^k(y_0) = p(y_0 \mid r_0 = \lambda_k)$.

$$\begin{cases} \widetilde{\mu}_0^{(2)}(\lambda_k) = D(\lambda_k)\, m_0 + e_0(\lambda_k) \\ \\ \widetilde{\Gamma}_0^{(2)}(\lambda_k) = D(\lambda_k)\, P_0\, D'(\lambda_k) + J(\lambda_k)\, C_0^{V}\, J'(\lambda_k) \end{cases}$$

→ $\forall\, k \in \{1, ..., K\}$ on calcul les coefficients Forward :

$$\hat{\alpha}_0^{(2)}(\lambda_k) = p(r_0 = \lambda_k \mid y_0) = \frac{p_0(\lambda_k)\ \psi_0^k(y_0)}{\sum_{l=1}^{K} p_0(\lambda_l)\ \psi_0^l(y_0)}$$

→ Estimation de l'état du processus d'intérêt et celui des sauts :

$$\hat{r}_0 = \arg\max_{r_0}\{\hat{\alpha}_0^{(2)}(r_0)\} = \arg\max_{\lambda\in\Omega}\{\hat{\alpha}_0^{(2)}(\lambda)\}$$

$$\hat{x}_0^{(2)} = \hat{m}_0^{(2)} = m_0 + P_0\ D'(\hat{r}_0)\ [\,\tilde{\Gamma}_0^{(2)}(\hat{r}_0)\,]^{-1}\ (y_0 - \tilde{\mu}_0^{(2)}(\hat{r}_0))$$

$$\hat{P}_0^{(2)} = P_0 - P_0\ D'(\hat{r}_0)\ [\,\tilde{\Gamma}_0^{(2)}(\hat{r}_0)\,]^{-1}\ D(\hat{r}_0)\ P_0^{'}$$

$(t > 0)$ **(Récurrence)**

Etape (1)

→ $\forall\, k \in \{1, ..., K\}$ On calcul les paramètres des densités :

$$\tilde{\varphi}_t^k(x_t) = p(x_t \mid r_t = \lambda_k, \hat{r}_0^{t-1}, y_0^{t-1}) \qquad \tilde{\psi}_t^k(y_t) = p(y_t \mid r_t = \lambda_k, \hat{r}_0^{t-1}, y_0^{t-1})$$

Données par les équations **(5.14)** et **(5.15),** respectivement.

→ $\forall\, k \in \{1, ..., K\}$ On calcul les coefficients Forward $\hat{\alpha}_t^{(2)}(\lambda_k)$ données par l'équation **(5.16)**.

Etape (2) (*Estimation*)

→ On donne l'estimateur de l'état du processus des sauts:

→

$$\hat{r}_t = \arg\max_{r_t}\{\hat{\alpha}_t^{(2)}(r_t)\} = \arg\max_{\lambda\in\Omega}\{\hat{\alpha}_t^{(2)}(\lambda)\}$$

→ On calcul les paramètres de la densité $\hat{\phi}_t(x_t) = p(x_t \mid \hat{r}_0^t, y_0^t)$ données par l'équation **(5.17)**.

→ Pour le processus d'intérêt on propose alors l'estimateur :

$$\hat{x}_t^{(2)} = \hat{m}_t^{(2)} = E[X_t \mid \hat{r}_0^t, y_0^t]$$

$t = t + 1$; Retour à l'étape 1.

Fin.

4.5.2 Expérimentations

On considère le système linéaire et gaussien avec ($d_X = d_Y = 1$) suivant:

$$\begin{cases} X_{t+1} = b_{t+1}(R_{t+1}) + A(R_{t+1})\, X_t + H(R_{t+1})\, W_{t+1} \\ \\ Y_t = e_t(R_t) + D(R_t)\, X_t + J(R_t)\, V_t \end{cases} \quad (4.25)$$

Avec, pour tout naturel t, $E[X_0] = E[W_{t+1}] = E[V_t] = 0$ et $E[X_0^2] = E[W_{t+1}^2] = E[V_t^2] = 1$.
R une chaîne de Markov homogène, stationnaire, et prenant ses valeurs dans l'espace à deux états $\Omega = \{\lambda_1, \lambda_2\}$. Sa loi est donnée par la loi initiale $p_0 = (\pi_0, 1-\pi_0)$ et les probabilité de transition:

$$p(r_2 |\ r_1 = r_2) = 1 - p(r_2 |\ r_1 \neq r_2) = 1 - \theta \quad (4.26)$$

On considère les notations suivantes: $\forall t \in \aleph^*, \forall \lambda \in \Omega$, $e_t(\lambda) = e(t, \lambda)$ et $b_t(\lambda) = b(t, \lambda)$, et on supposera dans toutes les expérimentations que

$$\begin{aligned} A(\lambda_2) &= -A(\lambda_1) = a\,, \\ D(\lambda_2) &= -D(\lambda_1) = d\,, \\ H(\lambda_2) &= \tau_2\,, \quad H(\lambda_1) = \tau_1\,, \\ J(\lambda_2) &= \sigma_2\,, \quad J(\lambda_1) = \sigma_1\,. \end{aligned} \quad (4.27)$$

Finalement, dans ce qui suit, on supposera que $e = b = 0$. Cette hypothèse rend le modèle assez bruité car les modèles conditionnels aux sauts correspondant aux deux valeurs de $\Omega = \{\lambda_1, \lambda_2\}$ ne se différentient que par des variances et des corrélations.
Finalement, nous considérons le modèle suivant

$$\begin{cases} X_{t+1} = A(R_{t+1})\, X_t + H(R_{t+1})\, W_{t+1} \\ \\ Y_t = D(R_t)\, X_t + J(R_t)\, V_t \end{cases}, \quad (4.28)$$

avec A, D, H, J définies par (4.27). L'objet de notre étude est de simuler, pour différentes valeurs des paramètres, les trois trajectoires du modèle et de restaurer les processus cachés en utilisant des algorithmes de filtrages vus précédemment.
Partant d'une réalisation y du processus Y on cherche donc une estimation des deux trajectoires r et x réalisations des deux processus R et X. Les trois processus étudiés sont corrélés suivant le système linéaire et gaussien (4.25). On va étudier l'évolution de l'erreur de restauration en fonction de la variation des paramètres:
On considère $T = 100$ et on étudie deux algorithmes de filtrage suivants :

(i) Le filtre particulaire, qui sera noté FKP ;

(ii) Le filtre fondé sur l'exploration de l'arbre des réalisations, noté FKA (A pour « alternatif), qui est « Algorithme 2 » fondée sur la Méthode 1 de la sous-section précédente.

Les erreurs seront mesurées par

$$Err_0 = \sum_{t=1}^{100} 1_{\{r_t = \hat{r}_t\}} \text{ (en \%)}, \; Err_1 = \frac{1}{100}\left[\sum_{t=1}^{100}(x_t - \hat{x}_t)^2\right]^{1/2}$$

Notons que le temps d'exécution de FKP dépend uniquement du nombre de particules choisi, le choix des autres paramètres ne joue pas de rôle. En moyennant sur plusieurs trajectoires pour différents paramètres on obtient la courbe présentée à la Figure 4.2.

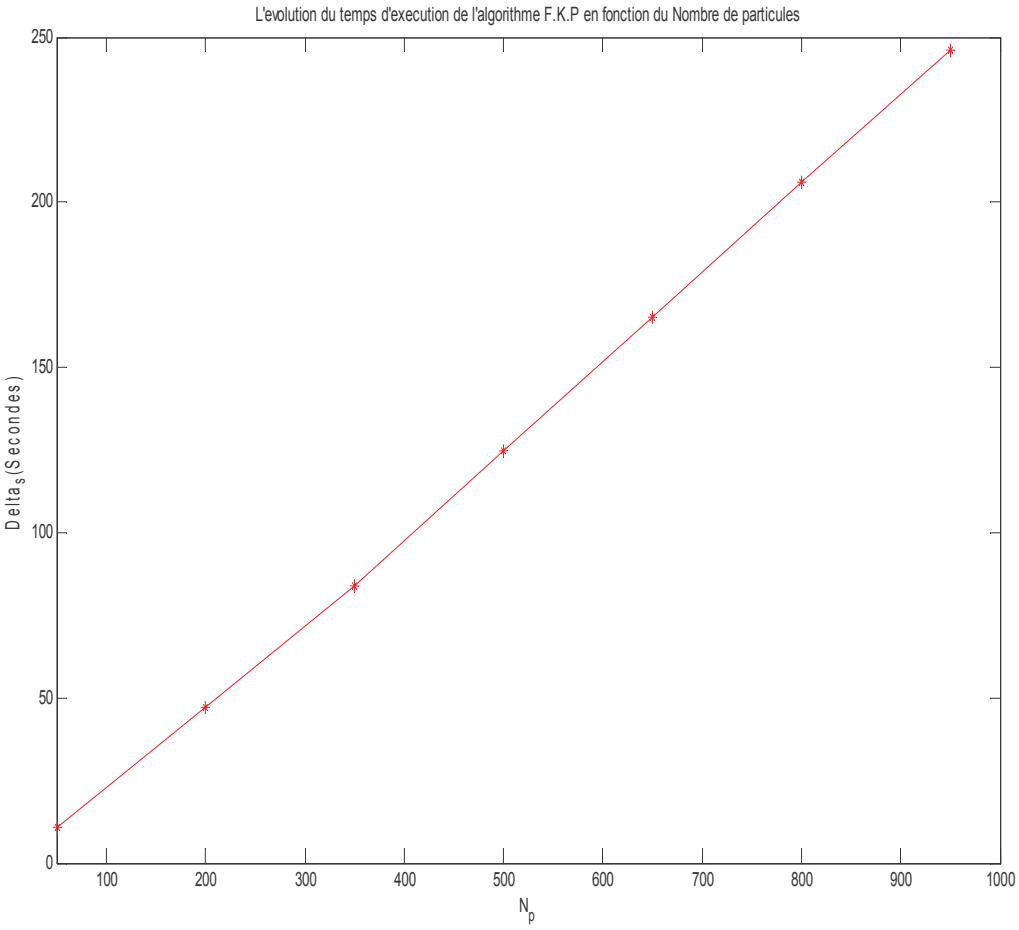


Figure 4.2 L'évolution du temps $\Delta_s(F.K.P)$ d'exécution de l'algorithme FKP en fonction du nombre des particules. Le temps $\Delta_s(F.K.A)$ d'exécution de l'algorithme FKA est $\Delta_s(F.K.A)$= 0.15 secondes.

On considère le cas $\sigma_1 = \sigma_2 = 0$, $a = 0.5$, $d = -2$, $\theta = p(r_2 | r_1 \neq r_2) = 0.1$, et on compare l'erreur Err_1 dans les cas des deux algorithmes en fonction de τ_1 et τ_2. Les différents résultats sont présentés sur les Figures 4.3 et 4.4.

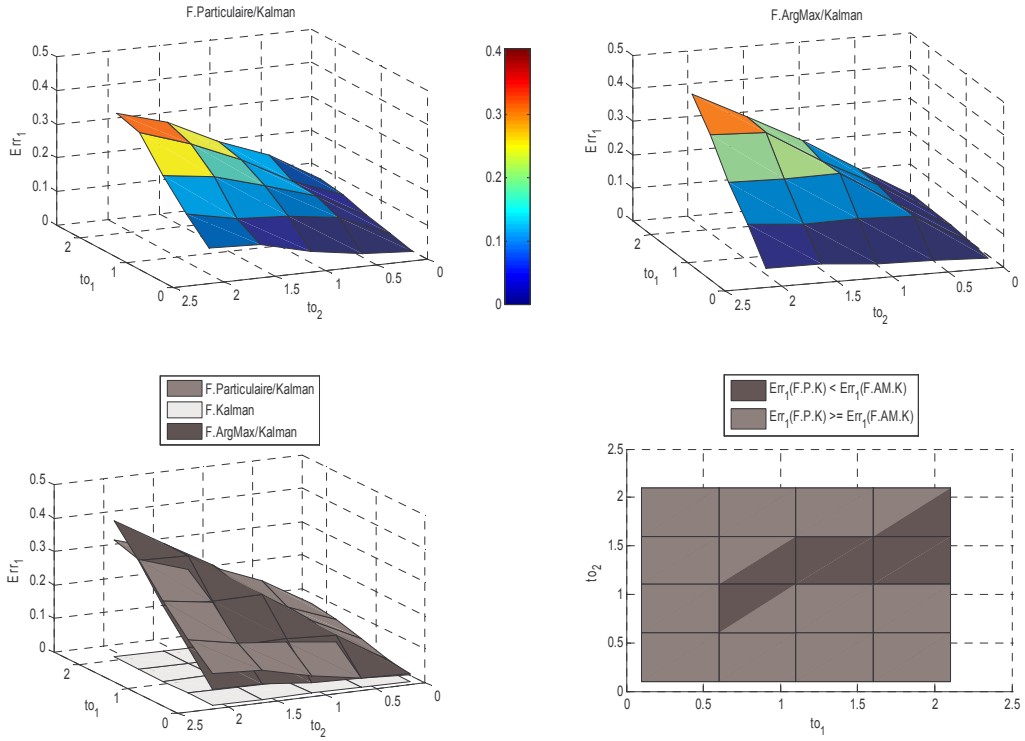


Figure 4.3 L'évolution de l'erreur Err_1 dans les cas des deux algorithmes en fonction de τ_1 et τ_2.

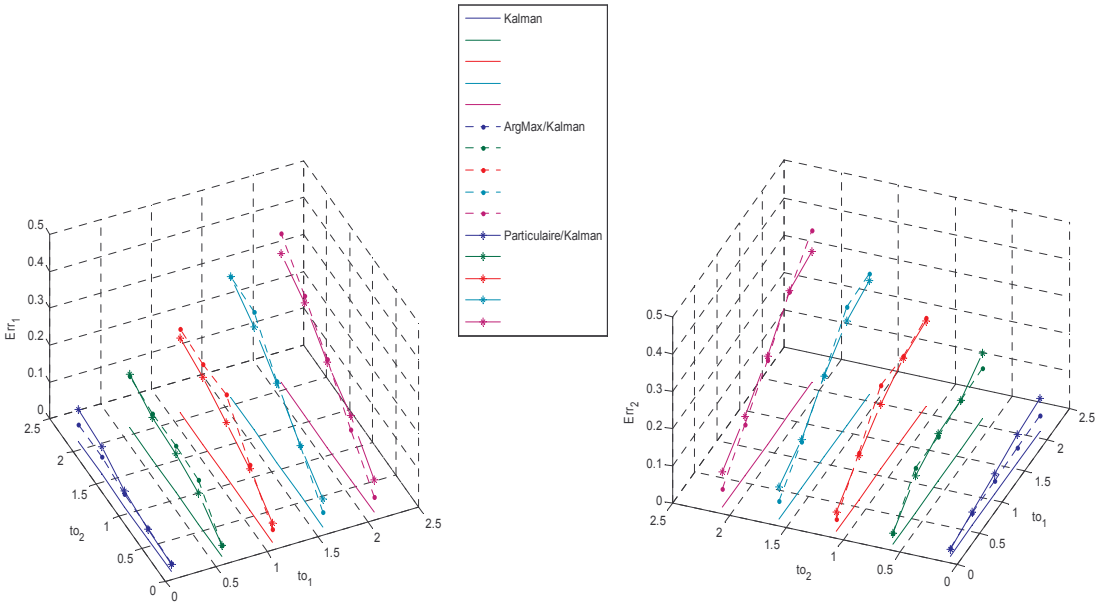


Figure 4.4 L'évolution de l'erreur Err_1 dans les cas des deux algorithmes en fonction de τ_1 et τ_2.

Nous donnons ci-dessous l'évolution de Err_1 en fonction du nombre des particules ; on constate que dans le cadre de notre expérience il n'est pas nécessaire d'aller au-delà de 350.

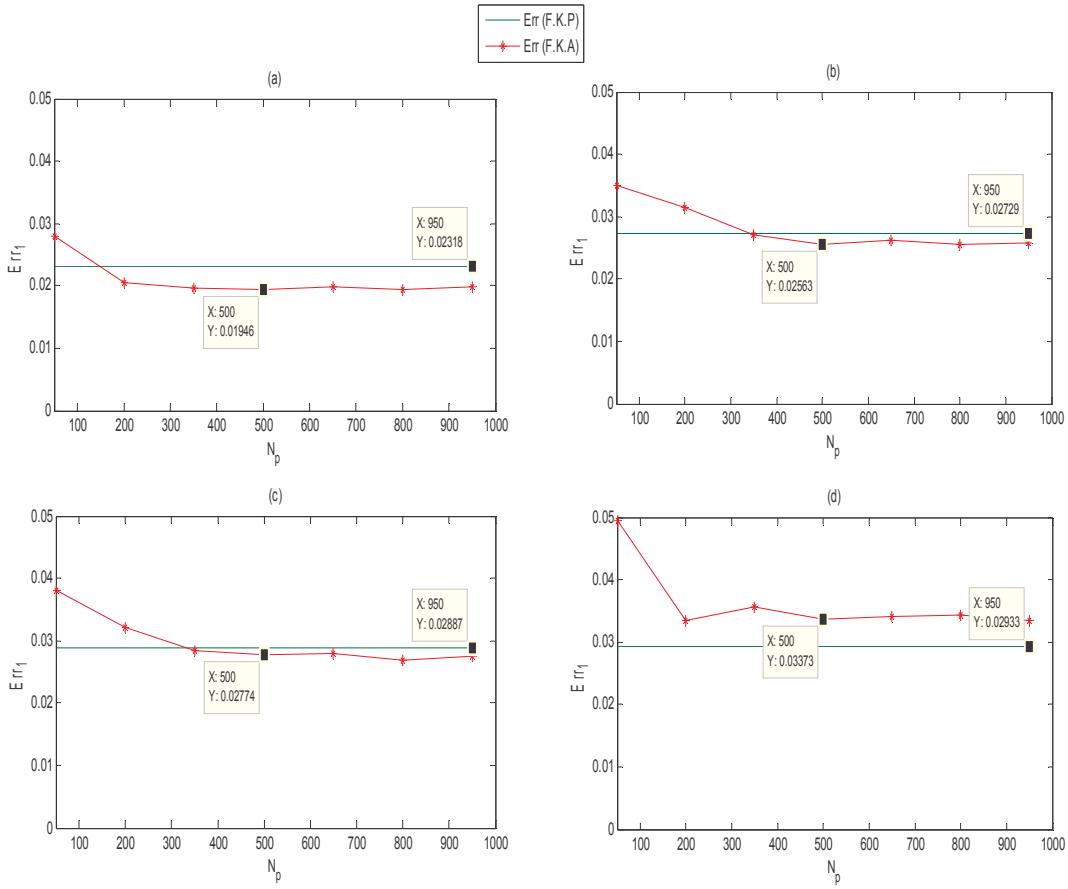


Figure 4.5 L'évolution de l'erreur Err_1 dans les cas des deux algorithmes en fonction du nombre de particules

θ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Err_0 (FKP)	7 %	7 %	15 %	16 %	19 %	18 %	14 %	10 %	9 %
Err_0 (FKA)	10 %	11 %	13 %	16 %	18 %	18 %	14 %	11 %	9%

Tableau 4.1 L'évolution de l'erreur Err_0 dans les cas des deux algorithmes en fonction de $\theta = p(r_2|r_1 \neq r_2)$, avec $a = 0.5,\ d = -2,\ \tau_1 = 2$, et $\tau_2 = 0.2$.

Nous notons que les différents résultats sont comparables, avec parfois un léger avantage à FKA. L'intérêt de ce dernier par rapport à FKP réside donc surtout dans sa vitesse d'exécution.

4.6 Approximation fondée sur la mémoire longue

Dans ce paragraphe on va proposer deux approximations du modèle initial (4.1) et on comparera les résultats obtenus avec les résultats donnés par le classique filtre particulaire.
L'idée générale est de remplacer la loi du couple $p(r_0^t, y_0^t)$, qui n'est pas connue dans le cas du modèle (4.1), par des lois manipulables. En effet, le calcul de la quantité $p(r_t \mid y_0^t)$, en théorie peut se faire de manière récursive dans le temps en utilisant la formule (4.7), mais en pratique elle est difficilement exploitable et on fait appelle à des algorithmes du type filtre particulaire, qui permet la propagation d'une approximation de cette dernière. Notre démarche consiste à se restreindre à des lois $p(r_0^t, y_0^t)$ de forme particulière où le calcul des quantités $p(r_t \mid y_0^t)$ reste possible, même pour t grand. Plus précisément, nous allons considérer pour les lois $p(r_0^t, y_0^t)$ la loi d'une CMC-BI, et la loi d'une CMC-BML étudiées dans le chapitre 2.
Rappelons que dans le cas d'une CMC-BI $p(r_0^t, y_0^t)$ s'écrit :

$$p(r_0^t, y_0^t) = p(r_0^t)p(y_0^t \mid r_0^t) = p(r_0^t)\prod_{j=0}^{t} p(y_j \mid r_j),$$

On a alors:

$$p(r_{t+1} \mid y_0^{t+1}) = \frac{p(y_{t+1} \mid r_{t+1})\sum_{r_t} p(r_{t+1} \mid r_t)p(r_t \mid y_0^t)}{\sum_{r_{t+1}} p(y_{t+1} \mid r_{t+1})\sum_{r_t} p(r_{t+1} \mid r_t)p(r_t \mid y_0^t)} \qquad (4.29)$$

Dans une CMC-BML $p(r_0^t, y_0^t)$ s'écrit

$$p(r_0^t, y_0^t) = p(r_0^t)p(y_0^t \mid r_0^t) = p(r_0^t)p(y_0 \mid r_0)\prod_{j=1}^{t} p(y_j \mid r_j, y_0^{j-1}),$$

et on a:

$$p(r_{t+1} \mid y_0^{t+1}) = \frac{p(y_{t+1} \mid r_{t+1}, y_0^t)\sum_{r_t} p(r_{t+1} \mid r_t)p(r_t \mid y_0^t)}{\sum_{r_{t+1}} p(y_{t+1} \mid r_{t+1}, y_0^t)\sum_{r_t} p(r_{t+1} \mid r_t)p(r_t \mid y_0^t)} \qquad (4.30)$$

Notons que les deux approximations proposées sont notamment utiles dans le cas où les paramètres de la chaîne des sauts ne sont pas connus. En effet, ainsi qu'exposé dans le chapitre 2 on peut alors faire appel à des méthodes d'estimations itératives du type ECI qui permettent d'estimer, dans le cas gaussien, tous les paramètres de la loi $p(r_0^t, y_0^t)$.

Nous avons

$$\forall\ t \in N \qquad p(x_{t+1}, r_{t+1} \mid y_0^{t+1}) = p(r_{t+1} \mid y_0^{t+1})p(x_{t+1} \mid r_{t+1}, y_0^{t+1})$$

Alors:

$$p(x_{t+1}|\ y_0^{t+1}) = \sum_{k=1}^{K} p(r_{t+1} = \lambda_k|\ y_0^{t+1}) p(x_{t+1}|\ r_{t+1} = \lambda_k, y_0^{t+1}) \tag{4.31}$$

On a:

$$p(x_{t+1}|\ r_{t+1}, y_0^t) = \int p(x_{t+1}|\ x_t, r_{t+1}) p(x_t|\ r_{t+1}, y_0^t)\ dx_t\ ; \tag{4.32}$$

$$p(x_{t+1}|\ r_{t+1}) = \int p(x_{t+1}|\ x_t, r_{t+1}) p(x_t|\ r_{t+1})\ dx_t\ ; \tag{4.33}$$

$$; p(y_{t+1}|\ r_{t+1}, y_0^t) = \int p(y_{t+1}|\ x_{t+1}, r_{t+1}) p(x_{t+1}|\ r_{t+1}, y_0^t)\ dx_{t+1} \tag{4.34}$$

$$p(y_{t+1}|\ r_{t+1}) = \int p(y_{t+1}|\ x_{t+1}, r_{t+1}) p(x_{t+1}|\ r_{t+1})\ dx_{t+1}\ . \tag{4.35}$$

D'autre part:

$$p(r_{t+1}|\ y_0^{t+1}) = \frac{p(y_{t+1}|\ r_{t+1}, y_0^t) \sum_{i=1}^{K} p(r_{t+1}|\ r_t = \lambda_i) p(r_t = \lambda_i|\ y_0^t)}{\sum_{l=1}^{K} p(y_{t+1}|\ r_{t+1} = \lambda_l, y_0^t) \sum_{k=1}^{K} p(r_{t+1} = \lambda_l|\ r_t = \lambda_k) p(r_t = \lambda_k|\ y_0^t)} \tag{4.36}$$

On définit alors les probabilités suivantes:

$$\overline{p}(r_{t+1}|\ y_0^{t+1}) = \frac{p(y_{t+1}|\ r_{t+1}) \sum_{i=1}^{K} p(r_{t+1}|\ r_t = \lambda_i) \overline{p}(r_t = \lambda_i|\ y_0^t)}{\sum_{l=1}^{K} p(y_{t+1}|\ r_{t+1} = \lambda_l) \sum_{k=1}^{K} p(r_{t+1} = \lambda_l|\ r_t = \lambda_k) \overline{p}(r_t = \lambda_k|\ y_0^t)}, \tag{4.37}$$

et

$$p(x_{t+1}|\ r_{t+1}, y_0^{t+1}) = \frac{p(x_{t+1}, r_{t+1}, y_{t+1}|\ y_0^t)}{p(r_{t+1}, y_{t+1}|\ y_0^t)} = \frac{p(x_{t+1}, y_{t+1}|\ r_{t+1}, y_0^t)}{p(y_{t+1}|\ r_{t+1}, y_0^t)} \tag{4.38}$$

$$= \frac{p(y_{t+1}|\ x_{t+1}, r_{t+1}) p(x_{t+1}|\ r_{t+1}, y_0^t)}{p(y_{t+1}|\ r_{t+1}, y_0^t)} \tag{4.39}$$

Remarque

Dans le calcul du filtre dans le vrai modèle les deux égalités (4.3) et (4.4) résultent directement de la règle de Bayes et de la simplification $p(x_t|r_0^{t+1}, y_0^t) = p(x_t|r_0^t, y_0^t)$, alors que dans la dernière approche qu'on propose c'est différent car généralement $p(x_t|\ r_{t+1}) \neq p(x_t|\ r_t)$ ou même $p(x_t|\ r_{t+1}, y_0^t) \neq p(x_t|\ r_t, y_0^t)$.

Posons pour tous i, j dans $\{1, ..., K\}$:

$$\pi_{t+1}^{(i,j)} = \frac{p(r_{t+1} = \lambda_j \mid r_t = \lambda_i) p(r_t = \lambda_i \mid y_0^t)}{\sum_{k=1}^{K} p(r_{t+1} = \lambda_j \mid r_t = \lambda_k) p(r_t = \lambda_k \mid y_0^t)} \tag{4.40}$$

$$\overline{\pi}_{t+1}^{(i,j)} = \frac{p(r_{t+1} = \lambda_j \mid r_t = \lambda_i) \overline{p}(r_t = \lambda_i \mid y_0^t)}{\sum_{k=1}^{K} p(r_{t+1} = \lambda_j \mid r_t = \lambda_k) \overline{p}(r_t = \lambda_k \mid y_0^t)} \tag{4.41}$$

Alors nous avons

$$p(x_t \mid r_{t+1} = \lambda_j, y_0^t) = \sum_{i=1}^{K} \pi_{t+1}^{(i,j)} \; p(x_t \mid r_t = \lambda_i, y_0^t) \tag{4.42}$$

et

$$p(x_t \mid r_{t+1} = \lambda_j) = \sum_{i=1}^{K} \overline{\pi}_{t+1}^{(i,j)} \; p(x_t \mid r_t = \lambda_i) \tag{4.43}$$

Les deux filtres fonctionnent selon les schémas suivants :

$$\begin{bmatrix} p(r_t \mid y_0^t) \\ p(x_t \mid r_t, y_0^t) \end{bmatrix} \rightarrow \left[p(x_t \mid r_{t+1}, y_0^t) \right] \xrightarrow{\text{prédiction}} \begin{bmatrix} p(x_{t+1} \mid r_{t+1}, y_0^t) \\ p(y_{t+1} \mid r_{t+1}, y_0^t) \end{bmatrix} \xrightarrow{\text{filtrage}} \begin{bmatrix} p(r_{t+1} \mid y_0^{t+1}) \\ p(x_{t+1} \mid r_{t+1}, y_0^{t+1}) \end{bmatrix}$$

$$\begin{bmatrix} \overline{p}(r_t \mid y_0^t) \\ p(x_t \mid r_t) \end{bmatrix} \rightarrow \left[p(x_t \mid r_{t+1}) \right] \xrightarrow{\text{prédiction}} \begin{bmatrix} p(x_{t+1} \mid r_{t+1}) \\ p(y_{t+1} \mid r_{t+1}) \end{bmatrix} \xrightarrow{\text{filtrage}} \begin{bmatrix} \overline{p}(r_{t+1} \mid y_0^{t+1}) \\ p(x_{t+1} \mid r_{t+1}) \end{bmatrix}$$

Les expressions (4.42) et (4.43) montrent que ces densités sont des mélanges, de lois gaussiennes dans le cas considéré. De même, à chaque instant t les densités $p(y_{t+1} \mid r_{t+1}, y_0^t)$ et $p(y_{t+1} \mid r_{t+1})$, sont des mélanges de lois. Si l'on fait l'hypothèse que ces lois sont des lois gaussiennes (et non des mélanges) on peut en calculer les paramètres. Cependant, la qualité d'une telle approximation dépend des coefficients de pondération ; elle dépend donc fortement de l'information apportée par les observations sur l'état de la chaîne à chaque instant t.

On peut donc raisonner différemment et supposer que (R, Y) est une CMC-BI (ou CMC-BML). Pour chaque t on pose $\hat{r}_t = \arg\max_{r_t \in \Omega} [p(r_t \mid y_0^t)]$ et on utilise l'algorithme de Kalman pour calculer la moyenne $E[X_t \mid \hat{r}_0^t, y_0^t] = \hat{x}_t$. Cette démarche peut être partiellement non supervisée dans la mesure où les paramètres de la loi $p(r_0^t, y_0^t)$ peuvent être estimés par la méthode ECI décrite dans le chapitre 2.

Exemple de simulation

On considère le modèle (4.28) avec $\Omega = \{\lambda_1, \lambda_2\}$, $A(\lambda_1) = -0.25$, $A(\lambda_2)$, $D(\lambda_1) = -2$, $D(\lambda_2) = 2$, $H(\lambda_1) = 0.1$, $H(\lambda_2) = 0.5$, $J(\lambda_1) = 0.5$, $J(\lambda_2) = 1$, $e_t = b_t = 0$, et $\theta = p(r_2 | r_1 \neq r_2) = 0.1$.

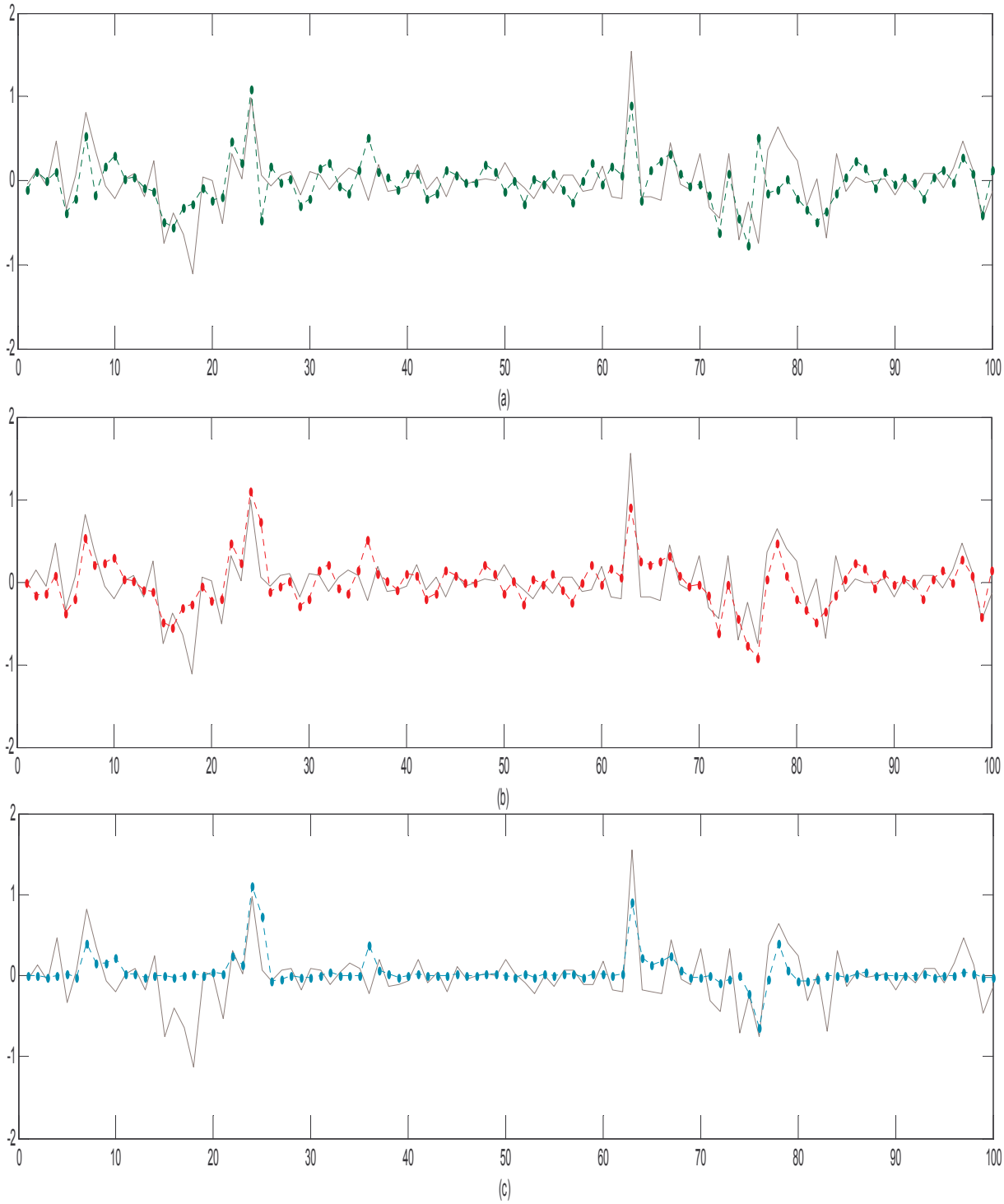


Figure 4.6 Données simulées suivant le modèle (4.28). (a) : la restauration par le filtre particulaire (500 particules) ; (b) : la restauration donnée par le nouveau filtre avec l'hypothèse CMC-BML ; (c) : la restauration donnée par le filtre avec l'hypothèse CMC-BI.

Conclusion

Les résultats présentés à la Figure 4.6 et d'autres résultats des simulations similaires montrent que le filtrage fondé sur les CMC-BML est plus efficaces que les le filtrage fondé sur les CMC-BI. Par ailleurs, il est souvent d'efficacité comparable à celle du filtrage particulaire. Il peut être très rapide si les paramètres de la CMC-BML sont connus, ou si leur estimation avec ECI est facile et nécessite peu d'itérations.

Chapitre 5

Chaines de Markov couples continues et modèles à sauts

Les modèles de chaînes de Markov couples et triplet sont strictement plus générales que les chaînes de Markov cachées. Dans le modèle couple la chaîne cachée n'est plus nécessairement de Markov, par contre la loi sa posteriori est toujours markovienne. Dans le modèle triplet la situation est encore plus générale car ni la chaîne cachée ni la chaîne couple ne sont nécessairement de Markov. Dans ce chapitre on s'intéresse, comme dans le chapitre précédent, au problème de restauration bayésienne par filtrage. Nous allons présenter des extensions des algorithmes présentés précédemment et nous allons procéder à leurs comparaisons. Ce chapitre présente un modèle de Markov triplet à sauts original, dans lequel le filtrage optimal exact est possible avec une complexité linéaire en temps. Les premières simulations tendent à montrer qu'il peut être perçu comme une alternative intéressante, au plan du temps de calcul, aux méthodes classiques fondées sur le filtrage particulaire. Les simulations et les exemples, comme tout au long de ce mémoire, concernent exclusivement les modèles avec bruits gaussiens.

5.1 Chaînes de Markov couples

Un processus $Z = (X, Y)$ sera appelé « chaîne de Markov couple », en abrégé CM-Co, si Z est une chaîne de Markov d'ordre 1. Comme précédemment, on supposera que X est caché et Y est observé. Une CM-Co peut donc être vue comme étant une chaîne de Markov vectorielle partiellement observée. On montre aisément qu'une chaîne de Markov cachée est une chaîne couple mais la réciproque est fausse [47]. On trouvera dans [46], [47] deux conditions nécessaires et suffisantes pour que, dans une CM-Co $Z = (X, Y)$ la chaîne X soit de Markov. Comme on va le voir par la suite la propriété de Markovianité de la loi a posteriori permet dans le cas de la chaîne couple de faire de la restauration bayésienne aussi simplement que pour les chaînes de Markov cachées.

Sachant que $p(z_t \mid z_1^{t-1}) = p(x_t \mid z_1^{t-1})\, p(y_t \mid x_t, z_1^{t-1})$, on a dans une CM-Co :

$$p(x_t \mid z_1^{t-1}) = p(x_t \mid z_{t-1}) = p(x_t \mid x_{t-1}, y_{t-1}) \tag{5.1}$$

$$p(y_t \mid x_t, z_1^{t-1}) = p(y_t \mid x_t, z_{t-1}) = p(y_t \mid x_t, x_{t-1}, y_{t-1}) \quad (5.2)$$

Par ailleurs, les lois de X_1^t et X_t conditionnelles à Y_1^t sont données par les densités $p(x_1^t \mid y_1^t)$, $p(x_t \mid y_1^t)$, qui peuvent s'écrire

$$p(x_1^t \mid y_1^t) = \frac{p(x_t \mid x_{t-1}, y_{t-1})\, p(y_t \mid x_t, x_{t-1}, y_{t-1})}{\int (x_t \mid x_{t-1}, y_{t-1})\, p(y_t \mid x_t, x_{t-1}, y_{t-1}) p(x_1^{t-1} \mid y_1^{t-1})\, dx_{1:t}}\, p(x_1^{t-1} \mid y_1^{t-1}), \qquad (5.3)$$

$$p(x_t \mid y_1^t) = \frac{\int p(y_t \mid x_t, x_{t-1}, y_{t-1})\ (x_t \mid x_{t-1}, y_{t-1})\ p(x_{t-1} \mid y_1^{t-1})\, dx_{t-1}}{\iint p(y_t \mid x_t, x_{t-1}, y_{t-1})\ (x_t \mid x_{t-1}, y_{t-1})\ p(x_{t-1} \mid y_1^{t-1}) dx_{t-1:t}}. \qquad (5.4)$$

Système linéaire gaussien

Dans le cas du modèle linéaire et gaussien que nous allons considérer maintenant la récurrence dans l'équation (5.4) nous permet de déduire une généralisation de l'algorithme de Kalman au cas couple, comme proposé dans [8]. Dans les cas plus généraux, une solution par filtrage particulaire à également était proposé dans [27].
Nous allons utiliser la propriété classique suivante :

Propriété 5.1:

Soit Z un vecteur aléatoire gaussien de moyenne $m_z \in \Re^{n_z}$ et de matrice variances-covariances $\Gamma^z \in M_{n_z \times n_z}(\Re)$. Alors le vecteur aléatoire défini par $\overline{Z} = \mu_z + H\, Z$ est gaussien de moyenne $\mu_z + H\, m_z$ et de matrice variances-covariances $H\, \Gamma^z\, H'$.

Nous adoptons les notations suivantes pour les vecteurs gaussiens. $Z = (X, Y)' \sim \mathcal{N}(m_z,\ \Gamma^z)$ avec $X \sim \mathcal{N}(m_x, \Gamma^x)$ et $Y \sim \mathcal{N}(m_y, \Gamma^y)$, on pose alors: $m_z = (m_x, m_y)'$ et $\Gamma^z = \begin{pmatrix} \Gamma^x & \Gamma^{xy} \\ \Gamma^{yx} & \Gamma^y \end{pmatrix}$, où $\Gamma^{xy} = (\Gamma^{yx})' = E[(X - m_x)(Y - m_y)']$. Par ailleurs, les lois conditionnelles seront parfois notées $(X \mid Y = y)$.

On a alors deux résultats importants qu'on va exposer ici sous forme de propositions.

Proposition 5.1:

Soit $Z = (X, Y)' \sim \mathcal{N}(m_z,\ \Gamma^z)$, avec $m_z = (m_x, m_y)'$ et $\Gamma^z = \begin{pmatrix} \Gamma^x & \Gamma^{xy} \\ \Gamma^{yx} & \Gamma^y \end{pmatrix}$.
Alors:

$$(X \mid Y = y) \sim \mathcal{N}(m_x + \Gamma^{xy}\left[\Gamma^y\right]^{-1}(y - m_y),\ \Gamma^x - \Gamma^{xy}\left[\Gamma^y\right]^{-1}\Gamma^{yx}),$$

$$(Y \mid X = x) \sim \mathcal{N}(m_y + \Gamma^{yx}\left[\Gamma^x\right]^{-1}(x - m_x),\ \Gamma^y - \Gamma^{yx}\left[\Gamma^x\right]^{-1}\Gamma^{xy}).$$

Proposition 5.2:

Soit $(X \mid Y = y) \sim \mathcal{N}(\mu_x + P\,y,\ \Gamma^{x|y})$ et $(Y \mid X = x) \sim \mathcal{N}(\mu_y + Q\,x,\ \Gamma^{y|x})$ alors:

$Z = (X, Y)' \sim \mathcal{N}(m_z,\ \Gamma^z)$, avec:

$$m_z = (P\,m_y + \mu_y,\ m_y)', \text{ et } \Gamma^z = \begin{pmatrix} \Gamma^{x|y} + P\,\Gamma^y P' & P\,\Gamma^y \\ \Gamma^y\,P' & \Gamma^y \end{pmatrix},$$

où

$$m_z = (m_x,\ Q\,m_x + \mu_x)', \text{ et } \Gamma^z = \begin{pmatrix} \Gamma^x & \Gamma^x\,Q' \\ Q\,\Gamma^x & \Gamma^{y|x} + Q\,\Gamma^x\,Q' \end{pmatrix}.$$

Loi de transition de la chaîne couple

Soit $(Z_t)_{t \in \aleph^*}$ une chaîne de Markov gaussienne et stationnaire (elle est alors également homogène), avec $Z_t \in \Re^{n_z}$.

Pour $Z_t \sim \mathcal{N}(m_z, \Gamma^z)$, on notera $Cov(Z_t, Z_{t+1}) = Cov(Z_1, Z_2) = G' = \begin{pmatrix} A & D \\ E & C \end{pmatrix}$. En utilisant la Proposition ***(5.1)*** on a:

$$\forall t \in \aleph^* \qquad (Z_{t+1} \mid Z_t = z) \sim \mathcal{N}(m_z + G\left[\Gamma^z\right]^{-1}(z - m_z),\ \Gamma^z - G\ \left[\Gamma^z\right]^{-1} G')$$

On pose:

$$F = G\left[\Gamma^z\right]^{-1},\ \ \mu_z = (\mu_x,\ \mu_y)' = (I_z - F)\,m_z, \text{ et } \ \Sigma = \Gamma^z - F\,G'$$

La matrice Σ étant symétrique et définie positive, il existe au moins une matrice réelle triangulaire inférieure S telle que $\Sigma = S\,S'$ (décomposition de Cholesky). De plus, si on impose que les éléments diagonaux de la matrice S soient tous positifs alors cette décomposition est unique.

On considère pour la suite les notations suivantes:

I_z : La matrice identité de taille $n_z \times n_z$;

$o_z = \underbrace{(0,...,0)}_{n_z}'$: Le vecteur nul de taille n_z ;

$[O]_{xy}$: La matrice nulle de taille $n_x \times n_y$.

On note les paramètres de la loi gaussienne du couple comme il suit:

$$\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m_z \\ m_z \end{pmatrix}, \begin{pmatrix} \Gamma^z & G' \\ G & \Gamma^z \end{pmatrix} \right).$$

On déduit alors la formulation du système linéaire correspondant suivante:

$$\forall t \in \aleph^*, \quad Z_{t+1} = \mu_z + F\, Z_t + S\, W_{t+1}, \tag{5.5}$$

où $W_{t+1} = (U_{t+1}, V_{t+1})' \sim \mathcal{N}(o_z, I_z)$ est une suite i.i.d. telle que pour tout $t \in \aleph^*$
$W_{t+1} = (U_{t+1}, V_{t+1})'$ est indépendante de Z_1^t.

Prédiction et filtrage

Dans ce paragraphe on expose une méthode qui permet le calcul des paramètres des deux densités gaussiennes $p(x_t \mid y_1^{t-1})$ et $p(x_t \mid y_1^t)$, à partir de $p(x_{t-1} \mid y_1^{t-1})$ et y_t, ce qui généralise l'algorithme de Kalman au modèle couple. Comme dans le cas du filtrage de Kalman, l'intérêt de cet algorithme de filtrage est le calcul des paramètres d'intérêts de manière récursive dans le temps. On adopte pour la suite les notations suivantes:

$$(X_t \mid Y_1^t = y_1^t) \sim \mathcal{N}(m_t, \Delta_t)$$

$$(Z_t,\ Z_{t+1} \mid Y_1^t = y_1^t) \sim \mathcal{N}\left(\begin{pmatrix} \eta_t \\ M_{t+1} \end{pmatrix}, \begin{pmatrix} L_t & \Lambda'_{t+1} \\ \Lambda_{t+1} & \Xi_{t+1} \end{pmatrix} \right)$$

$$(Z_t \mid Y_1^t = y_1^t) \sim \mathcal{N}\left(\eta_t = \begin{pmatrix} \eta_t^x \\ \eta_t^y \end{pmatrix},\ L_t = \begin{pmatrix} L_t^x & L_t^{xy} \\ L_t^{yx} & L_t^y \end{pmatrix} \right)$$

$$(Z_{t+1} \mid Y_1^t = y_1^t) \sim \mathcal{N}\left(M_{t+1} = \begin{pmatrix} M_{t+1}^x \\ M_{t+1}^y \end{pmatrix},\ \Xi_{t+1} = \begin{pmatrix} \Xi_{t+1}^x & \Xi_{t+1}^{xy} \\ \Xi_{t+1}^{yx} & \Xi_{t+1}^y \end{pmatrix} \right)$$

Montrons:

$$\eta_t = (m_t, y_t)^{'}, M_{t+1} = \mu_z + F\,\eta_t, L_t = \begin{pmatrix} \Delta_t & [O]_{xy} \\ [O]_{yx} & [O]_{yy} \end{pmatrix}, \Lambda_{t+1} = F\,L_t, \tag{5.6}$$

et $\Xi_{t+1} = F\,L_t\,F^{'} + \Sigma$

Nous avons pour tout t dans $\aleph^*$:

$$\eta_t = E[Z_t \mid Y_1^t = y_1^t] = \begin{pmatrix} E[X_t \mid Y_1^t = y_1^t] \\ E[Y_t \mid Y_1^t = y_1^t] \end{pmatrix} = (m_t, y_t)^{'}$$

$$\begin{aligned} M_{t+1} = E[Z_{t+1} \mid Y_1^t = y_1^t] &= E[\mu_z + F\,Z_t + S\,W_{t+1} \mid Y_1^t = y_1^t] \\ &= \mu_z + F\,E[Z_t \mid Y_1^t = y_1^t] + S\,E[W_{t+1} \mid Y_1^t = y_1^t] \\ &= \mu_z + F\,\eta_t \end{aligned}$$

$$\begin{aligned} L_t &= E[(Z_t - \eta_t)(Z_t - \eta_t)^{'} \mid Y_1^t = y_1^t] \\ &= E\left[\begin{pmatrix} (X_t - m_t) \\ (Y_t - y_t) \end{pmatrix}\left((X_t - m_t)^{'}, (Y_t - y_t)^{'}\right) \mid Y_1^t = y_1^t\right] \\ &= \begin{pmatrix} E[(X_t - m_t)(X_t - m_t)^{'} \mid Y_1^t = y_1^t] & E[(X_t - m_t)(Y_t - y_t)^{'} \mid Y_1^t = y_1^t] \\ E[(Y_t - y_t)(X_t - m_t)^{'} \mid Y_1^t = y_1^t] & E[(Y_t - y_t)(Y_t - y_t)^{'} \mid Y_1^t = y_1^t] \end{pmatrix} \\ &= \begin{pmatrix} \Delta_t & [O]_{xy} \\ [O]_{yx} & [O]_{yy} \end{pmatrix} \end{aligned}$$

$$\begin{aligned} \Lambda_{t+1} &= E[(Z_{t+1} - M_{t+1})(Z_t - \eta_t)^{'} \mid Y_1^t = y_1^t] \\ &= E[F\,(Z_t - \eta_t)(Z_t - \eta_t)^{'} \mid Y_1^t = y_1^t] \\ &= F\,E[F\,(Z_t - \eta_t)(Z_t - \eta_t)^{'} \mid Y_1^t = y_1^t] \\ &= F\,L_t \end{aligned}$$

$$\begin{aligned} \Xi_{t+1} &= E[(Z_{t+1} - M_{t+1})(Z_{t+1} - M_{t+1})^{'} \mid Y_1^t = y_1^t] \\ &= E[(F\,(Z_t - \eta_t) + S\,W_{t+1})\ (F\,(Z_t - \eta_t) + S\,W_{t+1})^{'} \mid Y_1^t = y_1^t] \\ &= F\,E[(Z_t - \eta_t)(Z_t - \eta_t)^{'} \mid Y_1^t = y_1^t]\,F^{'} + S\,E[W_{t+1}\,W_{t+1}^{'} \mid Y_1^t = y_1^t]\,S^{'} \\ &= F\,L_t\,F^{'} + S\,E[W_{t+1}\,W_{t+1}^{'}]\,S^{'} \\ &= F\,L_t\,F^{'} + S\,S^{'} \\ &= F\,L_t\,F^{'} + \Sigma \end{aligned}$$

Partant des paramètres de l'équation itérative définie dans ***(5.5)*** on peut ainsi calculer ceux de la loi de la chaîne cachée conditionnellement aux observations.

$$\begin{pmatrix} \mu_z \\ F \\ \Sigma \\ m_t \\ \Delta_t \end{pmatrix} \quad ----------> \quad \begin{pmatrix} \eta_t \\ M_{t+1} \\ L_t \\ \Lambda_{t+1} \\ \Xi_{t+1} \end{pmatrix}$$

Notons que les équations de filtrage et de prédiction peuvent être données selon deux schémas : soit en commençant par l'étape de prédiction suivie de celle du filtrage, ou bien inversement commencer par l'étape de filtrage suivie de celle de la prédiction.
Le schéma le plus commode dépendra de l'application traitée et des conditions initiales. Nous avons :

$$p(x_t \mid y_1^t) \xrightarrow[prédiction]{} p(x_{t+1} \mid y_1^t) \xrightarrow[filtrage]{} p(x_{t+1} \mid y_1^{t+1}) \tag{5.7}$$

$$p(x_t \mid y_1^{t-1}) \xrightarrow[filtrage]{} p(x_t \mid y_1^t) \xrightarrow[prédiction]{} p(x_{t+1} \mid y_1^t) \tag{5.8}$$

On fait le choix de décrire le premier parcours, sachant que le second est analogue et diffère du premier uniquement l'ordre des équations dans l'algorithme.

$$\begin{pmatrix} \mu_z \\ F \\ \Sigma = S\,S' \\ m_t \\ \Delta_t \end{pmatrix} \xrightarrow[prédiction]{} \begin{pmatrix} M_{t+1}^x \\ M_{t+1}^y \\ \Xi_{t+1}^x \\ \Xi_{t+1}^y \\ \Xi_{t+1}^{xy} \end{pmatrix} \xrightarrow[filtrage]{} \begin{pmatrix} m_{t+1} \\ \Delta_{t+1} \end{pmatrix}$$

On garde les notations $G' = \begin{pmatrix} A & D \\ E & C \end{pmatrix}$, $F = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$ on peut expliciter la matrice de variance-covariance.

On pose $\widetilde{\Gamma}^z = (\Gamma^z)^{-1} = \begin{pmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{pmatrix}$ on a :

$$F = G\,\widetilde{\Gamma}^{z} \quad \Rightarrow \quad \begin{cases} \alpha = A^{'}\,\widetilde{\Gamma}_{11} + E^{'}\,\widetilde{\Gamma}_{21} \\ \beta = A^{'}\,\widetilde{\Gamma}_{12} + E^{'}\,\widetilde{\Gamma}_{22} \\ \gamma = D^{'}\,\widetilde{\Gamma}_{11} + C^{'}\,\widetilde{\Gamma}_{21} \\ \delta = D^{'}\,\widetilde{\Gamma}_{12} + C^{'}\,\widetilde{\Gamma}_{22} \end{cases}$$

$$G^{'} = \Gamma^{z}\,F^{'} \quad \Rightarrow \quad \begin{cases} A = \Gamma^{x}\alpha^{'} + \Gamma^{xy}\beta^{'} \\ D = \Gamma^{x}\gamma^{'} + \Gamma^{xy}\delta^{'} \\ E = \Gamma^{yx}\alpha^{'} + \Gamma^{y}\beta^{'} \\ C = \Gamma^{yx}\gamma^{'} + \Gamma^{y}\delta^{'} \end{cases}$$

$$\begin{aligned} \Sigma &= \Gamma^{z} - F\,\Gamma^{z}\,F^{'} \\ &= \begin{pmatrix} \Gamma^{x} - (\alpha\Gamma^{x}\alpha^{'} + \beta\Gamma^{y}\beta^{'} + \alpha\Gamma^{xy}\beta^{'} + \beta\Gamma^{yx}\alpha^{'}) & \Gamma^{xy} - (\alpha\Gamma^{x}\gamma^{'} + \beta\Gamma^{yx}\gamma^{'} + \alpha\Gamma^{xy}\delta^{'} + \beta\Gamma^{y}\delta^{'}) \\ \Gamma^{yx} - (\gamma\Gamma^{x}\alpha^{'} + \delta\Gamma^{yx}\alpha^{'} + \gamma\Gamma^{xy}\beta^{'} + \delta\Gamma^{y}\beta^{'}) & \Gamma^{y} - (\gamma\Gamma^{x}\gamma^{'} + \delta\Gamma^{y}\delta^{'} + \gamma\Gamma^{xy}\delta^{'} + \delta\Gamma^{yx}\gamma^{'}) \end{pmatrix} \end{aligned} \tag{5.10}$$

On peut alors réécrire les équations de prédiction et filtrage données dans ***(5.6)***.en fonction des paramètres connus à l'instant courant.
Nous avons :

Prédiction:

$$M_{t+1} = \mu_z + F\,\eta_t \quad \Rightarrow \quad \begin{cases} M^{x}_{t+1} = \mu_x + \alpha\,m_t + \beta\,y_t = m_x + \alpha\,(m_t - m_x) + \beta\,(y_t - m_y) \\ M^{y}_{t+1} = \mu_y + \gamma\,m_t + \delta\,y_t = m_y + \gamma\,(m_t - m_x) + \delta\,(y_t - m_y) \end{cases},$$

$$\Xi_{t+1} = F\,L_t\,F^{'} + \Sigma \quad \Rightarrow \quad \begin{cases} \Xi^{x}_{t+1} = \Sigma_{11} + \alpha\ \Delta_t\,\alpha^{'} \\ \Xi^{y}_{t+1} = \Sigma_{22} + \gamma\ \Delta_t\,\gamma^{'} \\ \Xi^{xy}_{t+1} = \Sigma_{12} + \alpha\ \Delta_t\,\gamma^{'} \\ \Xi^{yx}_{t+1} = \Sigma_{21} + \gamma\ \Delta_t\,\alpha^{'} \end{cases}$$

Filtrage

Etant donné que les paramètres Ξ_{t+1} et M_{t+1} de la densité gaussienne $p(z_{t+1}\,|\,y_1^t)$ sont calculés, la proposition 5.1 nous donne les paramètres de la densité gaussienne conditionnelle $p(x_{t+1}\,|\,y_1^{t+1})$.

$$m_{t+1} = M^{x}_{t+1} + (\Sigma_{12} + \alpha\,\Delta_t\,\gamma^{'})\,(\Sigma_{22} + \gamma\,\Delta_t\,\gamma^{'})^{-1}\,(y_{t+1} - M^{y}_{t+1}), \tag{5.11}$$

$$\Delta_{t+1} = (\Sigma_{12} + \alpha\,\Delta_t\,\alpha^{'}) - (\Sigma_{12} + \alpha\,\Delta_t\,\gamma^{'})\,(\Sigma_{22} + \gamma\,\Delta_t\,\gamma^{'})^{-1}\,(\Sigma_{21} + \gamma\,\Delta_t\,\alpha^{'}) \quad (5.12)$$

Finalement l'algorithme de l'estimation de l'état caché par la moyenne a posteriori dans le cas linéaire général des chaînes de Markov couples gaussiennes et stationnaires se déroule de la manière suivante.

Algorithme 2.3: L'estimation de l'état caché

- Initialisation : pour $t = 1$, on a $Z_1 = (X_1, Y_1)^{'} \sim \mathcal{N}(m_z,\ \Gamma^z)$ alors :

$$(X_1 \,|\, Y_1 = y_1) \sim \mathcal{N}(m_1,\ \Delta_1) \text{ , avec } \begin{cases} m_1 = m_x + \Gamma^{xy}\left[\Gamma^y\right]^{-1}(y_1 - m_y) \\ \Delta_1 = \Gamma^x - \Gamma^{xy}\left[\Gamma^y\right]^{-1}\Gamma^{yx} \end{cases}$$

$$\hat{x}_1 = m_1$$

- Etape Intermédiaire (1) : pour $t \geq 1$, on a $Cov(Z_t, Z_{t+1}) = G^{'}$ on pose $F = G\left[\Gamma^z\right]^{-1}$ et $\Sigma = \Gamma^z - F\,G^{'}$ Alors :

$$\left(\begin{pmatrix} Z_t \\ Z_{t+1} \end{pmatrix} \,|\, Y_1^t = y_1^t\right) \sim \mathcal{N}\left(\begin{pmatrix} \eta_t \\ M_{t+1} \end{pmatrix}, \begin{pmatrix} L_t & \Lambda_{t+1}^{'} \\ \Lambda_{t+1} & \Xi_{t+1} \end{pmatrix}\right),$$

avec :

$$\eta_t = (m_t, y_t)^{'},\ \ M_{t+1} = m_z + F\,(\eta_t - m_z),\ \ L_t = \begin{pmatrix} \Delta_t & [O]_{xy} \\ [O]_{yx} & [O]_{yy} \end{pmatrix},\ \ \Lambda_{t+1} = F\,L_t,$$

$$\Xi_{t+1} = F\,L_t\,F^{'} + \Sigma$$

- Etape prédiction (2) : pour $t \geq 1$ on a $(Z_{t+1} \,|\, Y_1^t = y_1^t) \sim \mathcal{N}\left(M_{t+1},\ \Xi_{t+1}\right)$ alors :

$$(X_{t+1} \,|\, Y_1^t = y_1^t) \sim \mathcal{N}\left(M_{t+1}^x,\ \Xi_{t+1}^x\right)$$

Etape filtrage (3)

- Pour $t \geq 1$ on a $(Z_{t+1} \,|\, Y_1^t = y_1^t) \sim \mathcal{N}\left(M_{t+1},\ \Xi_{t+1}\right)$ alors :

$$(X_{t+1} \,|\, Y_1^{t+1} = y_1^{t+1}) \sim \mathcal{N}\left(m_{t+1},\ \Delta_{t+1}\right) \quad \text{(Proposition 5.1)}$$

Recommencer à l'instant suivant les étapes (1)-(2)-(3) pour $t = t + 1$.

Fin

Modèle de chaîne de Markov cachées avec bruit indépendant

On montre dans ce paragraphe que le modèle de chaîne de Markov couple englobe le cas classique des chaînes de Markov cachées à bruit indépendants. On se place toujours dans le cas stationnaire homogène et on considère les signaux caché et observé centrés.

On s'intéresse donc au système dynamique linéaire et gaussien classique, où les bruits sont mutuellement indépendants, qui s'écrit de la manière suivante.

$$\begin{cases} X_{t+1} = \phi\, X_t + \Phi\, U_{t+1} \\ \\ Y_t = \psi\, X_t + \Psi\, V \end{cases}, \qquad (5.13)$$

avec :

$$\Sigma^x = \Phi\,\Phi^{'} = \Gamma^x - \phi\;\Gamma^x\,\phi^{'} \quad \& \quad \Gamma^y = \psi\,\Gamma^x\,\psi^{'} + \Sigma^y = \psi\,\Gamma^x\,\psi^{'} + \Psi\,\Psi^{'}$$

On a :

$$\forall t \in \aleph^* \qquad Y_{t+1} = \psi\, X_{t+1} + \Psi\, V_{t+1}$$

$$= \psi\,(\phi\, X_t + \Phi\, U_{t+1}) + \Psi\, V_{t+1}$$

$$= \psi\,\phi\, X_t + \psi\,\Phi\, U_{t+1} + \Psi\, V_{t+1}$$

Posons $\forall t \in \aleph^* \quad Z_{t+1} = (X_{t+1}, Y_{t+1})^{'}$ le système **(5.13)** est équivalent au modèle couple donné par l'équation vectorielle suivante :

$$\forall t \in \aleph^*,\ Z_{t+1} = F\, Z_t + S\, W_{t+1}, \qquad (5.14)$$

avec $W_{t+1} = (U_{t+1}, V_{t+1})^{'} \sim \mathcal{N}(o_z, I_z)$, $F = \begin{pmatrix} \phi & 0 \\ \psi\,\phi & 0 \end{pmatrix}$ et $S = \begin{pmatrix} \Phi & 0 \\ \psi\,\Phi & \Psi \end{pmatrix}$

Nous constatons en quoi le modèle couple est plus général que le modèle caché classique. La matrice F est quelconque dans le cas général, alors qu'elle contient deux zéros dans le cas caché classique. De même, la matrice S est générale dans le cas couple alors qu'elle contient un zéro dans la cas caché classique.

Reformulons (5.5) sous la forme suivante

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \end{bmatrix} + \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} U_{t+1} \\ V_{t+1} \end{bmatrix} \qquad (5.15)$$

Ainsi que nous l'avons mentionné plus haut, le couple $Z=(X,Y)$ est de Markov, mais une des chaînes X, Y n'et nécessairement de Markov. On peut voir rapidement des conditions suffisantes pour que X ou Y soit de Markov. En effet, si $\beta=0$ la chaîne X est de Markov, et si $\gamma=0$ la chaîne Y est de Markov. On a donc deux cas particuliers « symétriques ». Le premier est une extension du cas classique (5.13), dans lequel on a en plus $\delta=0$, et le deuxième est un cas introduit très récemment, qui aura une grande importance dans l'étude des modèles à sauts dans la sous-section suivante.

Exemples de simulations

Dans ce paragraphe on considère le problème de restauration dans le cas monodimensionnel $n_x=n_y=1$. On se fixe également un horizon temps $T=200$ pour les différentes courbes ainsi que pour le calcul de l'erreur de restauration.

On va adopter des notations en minuscules pour les paramètres :

$$\Gamma^z=\begin{pmatrix}\upsilon^2 & b\\ b & \omega^2\end{pmatrix},\ G^{'}=\begin{pmatrix}a & d\\ e & c\end{pmatrix},\ F=\begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix},$$

On va s'intéresser dans notre exemple aux chaînes couples telle que :

$$S=\begin{pmatrix}\tau & 0\\ \sigma\,\theta & \sigma\sqrt{(1-\theta^2)}\end{pmatrix},\ \text{et}\ \Sigma=SS^{'}=\begin{pmatrix}\tau^2 & \tau\,\sigma\,\theta\\ \tau\,\sigma\,\theta & \sigma^2\end{pmatrix},\ \text{avec}\ \ \theta\in[0,1].$$

On suppose que $\varpi=\upsilon^2\omega^2-b^2\neq 0$, on a alors: $\left[\Gamma^z\right]^{-1}=\frac{1}{\varpi}\begin{pmatrix}\omega^2 & -b\\ -b & \upsilon^2\end{pmatrix}$ et l'ensemble des paramètres vérifie les contraintes suivantes

$$\tau^2=(1-\alpha^2)\ \upsilon^2-\beta^2\omega^2-2\,\alpha\,\beta\,b=\frac{1}{\varpi}\left[(\varpi-e^2)\,\upsilon^2-a^2\omega^2+2\,a\,e\,b\right]$$

$$\sigma^2=(1-\delta^2)\ \omega^2-\gamma^2\upsilon^2-2\,\gamma\,\delta\,b=\frac{1}{\varpi}\left[(\varpi-d^2)\,\omega^2-c^2\upsilon^2+2\,c\,d\,b\right]$$

$$\tau\,\sigma\,\theta=(1-\alpha\,\delta-\gamma\,\beta)\,b-(\alpha\,\gamma\,\upsilon^2+\beta\,\delta\,\omega^2)=\frac{1}{\varpi}\left[(\varpi+a\,c+e\,d)\,b-a\,d\,\omega^2-c\,e\upsilon^2\right]$$

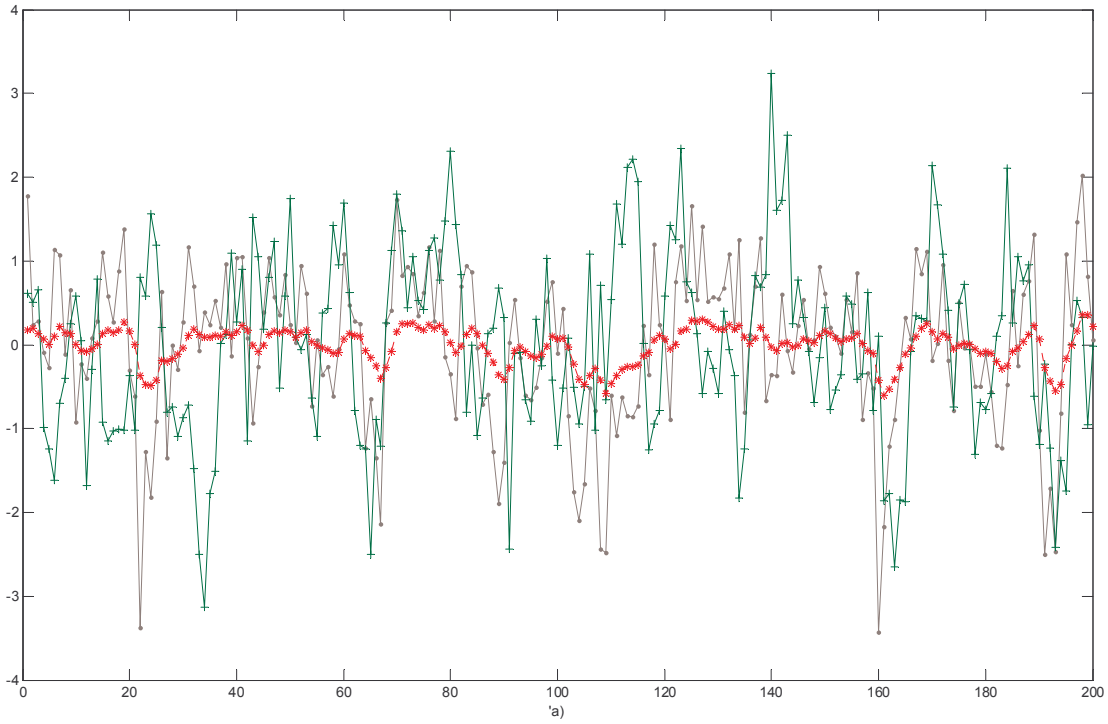
On a:

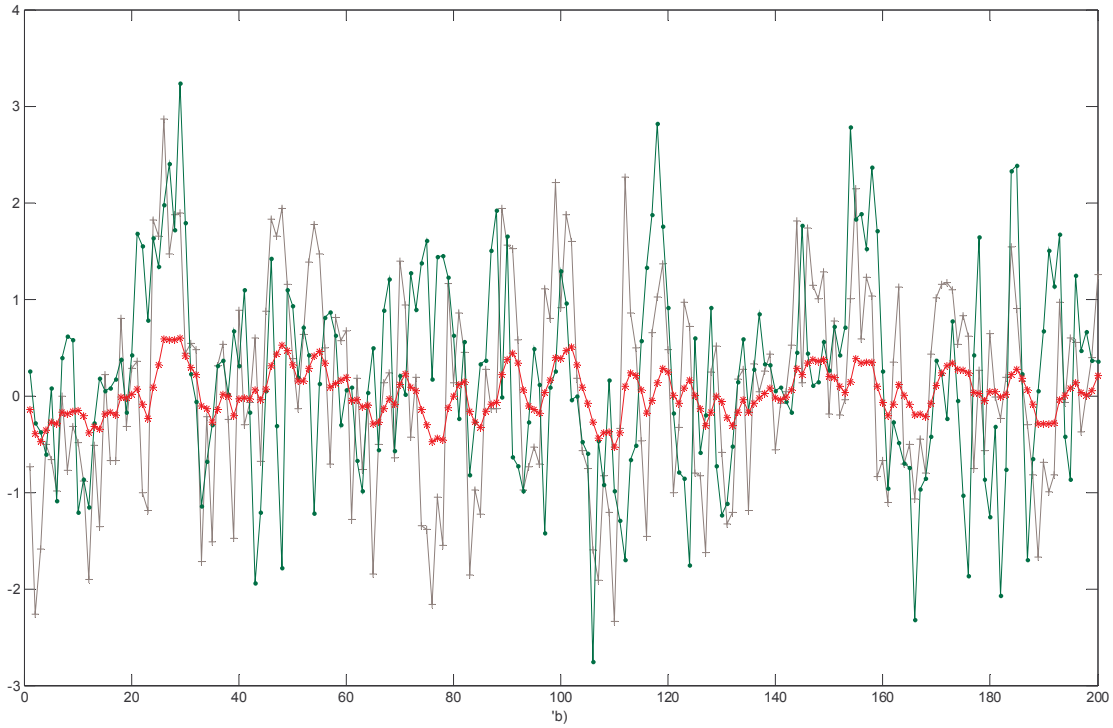
$$F = G\ (\Gamma^z)^{-1} \Rightarrow \begin{cases} \alpha = \dfrac{1}{\varpi}(a\,\omega^2 - e\,b) \\ \beta = \dfrac{1}{\varpi}(e\,\upsilon^2 - a\,b) \\ \gamma = \dfrac{1}{\varpi}(d\,\omega^2 - c\,b) \\ \delta = \dfrac{1}{\varpi}(c\,\upsilon^2 - d\,b) \end{cases}, \text{ et } G' = \Gamma^z\ F' \Rightarrow \begin{cases} a = \alpha\,\upsilon^2 + \beta\,b \\ d = \gamma\,\upsilon^2 + \delta\,b \\ e = \alpha\,b + \beta\,\omega^2 \\ c = \gamma\,b + \delta\,\omega^2 \end{cases}$$

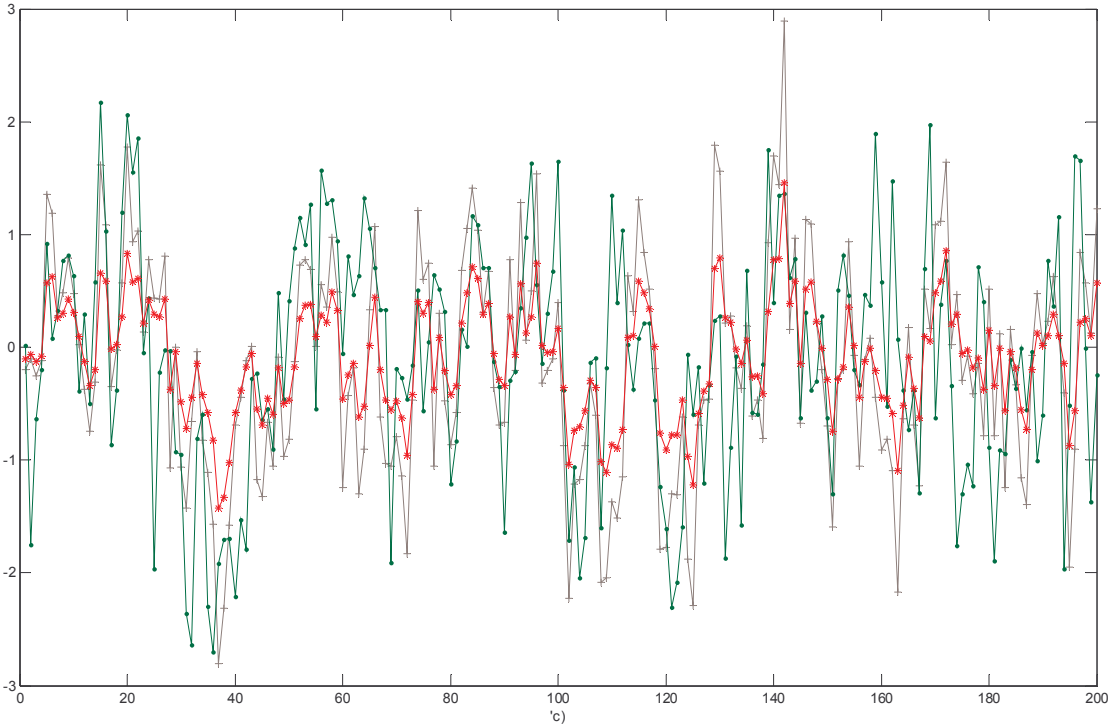
Dans les exemples ci-après on considère exclusivement les signaux centrés $m_x = m_y = 0$.

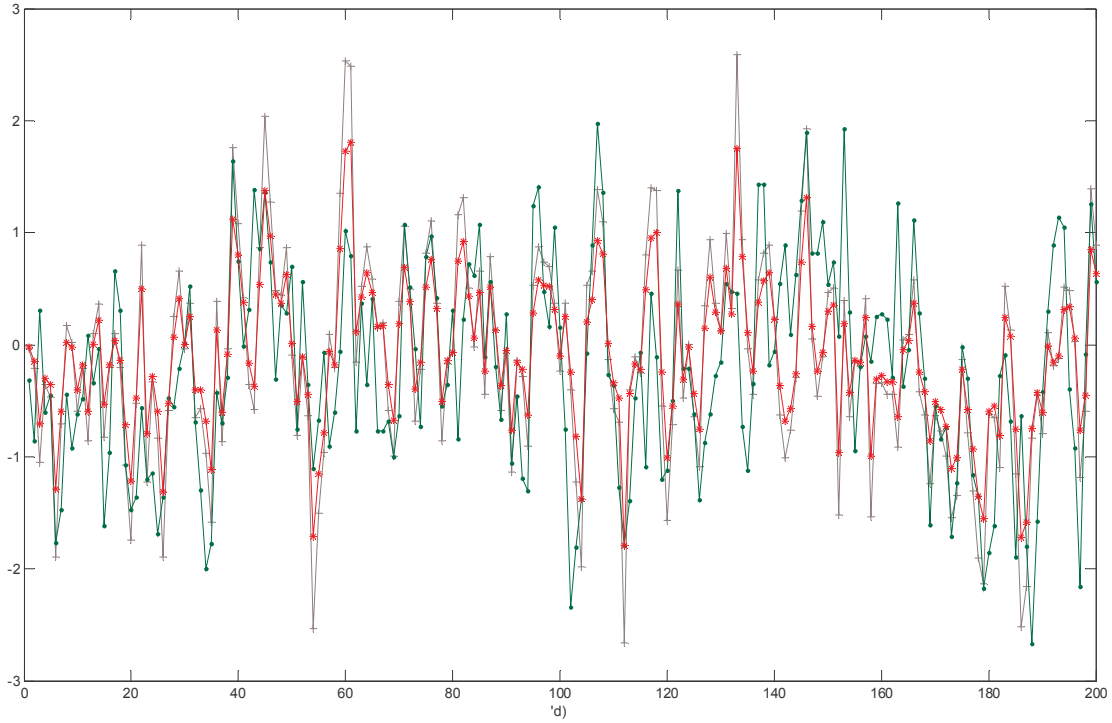
Exemple 1 (cas symétrique)

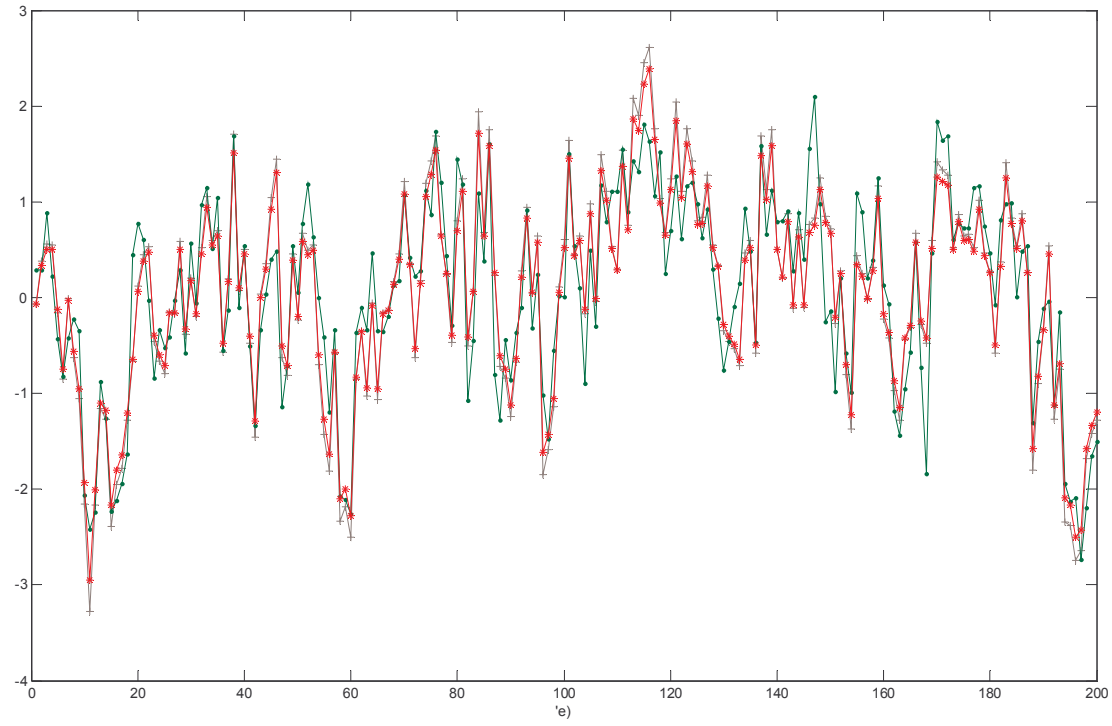
$\beta = \gamma = 0.1$, $\alpha = \delta = 0.5$, $\upsilon = \omega = 1$, et $b \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$




4
3
2
1
0
-1
-2
-3
0
20
40
60
80
100
120
140
160
180
200
'b)


3
2
1
0
-1
-2
-3
0
20
40
60
80
100
120
140
160
180
200
'c)





Les cinq figures au-dessus représentent le signal vrai en vert le signal bruité en gris et enfin la restauration par filtrage en rouge respectivement pour les valeurs de $b \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ dans l'ordre croissant et l'erreur de restauration correspondante dans l'ordre $\{0.963, 0.901, 0.604, 0.51, 0.307\}$.

→ **Cas** $\beta = 0 \quad (e\,\upsilon^2 = a\,b)$

On a:

$$\begin{cases} a = \alpha\,\upsilon^2 \\ d = \gamma\,\upsilon^2 + \delta\,b \\ e = \alpha\,b \\ c = \gamma\,b + \delta\,\omega^2 \end{cases} \quad \text{et} \quad \begin{cases} \tau^2 = (1-\alpha^2)\;\upsilon^2 = \upsilon^2 + \dfrac{a}{\varpi}(e\,b - a\omega^2) \\ \sigma^2 = (1-\delta^2)\,\omega^2 - \gamma^2\,\upsilon^2 - 2\,\gamma\,\delta\,b = \omega^2 - \dfrac{d}{\varpi}(c\,b + d\,\omega^2) \\ \tau\,\sigma\,\theta = b - \alpha\,\gamma\,\upsilon^2 = b + \dfrac{d}{\varpi}(e\,b - a\,\omega^2) \end{cases}$$

Le dernier système est bien définit si $\begin{cases} 1 \geq |\alpha| \\ \omega^2 \geq \delta^2\omega^2 + \gamma^2\upsilon^2 + 2\,\gamma\,\delta\,b \end{cases}$

Exemple 2

$$\beta = 0,\ \gamma = 0.1,\ \ \alpha = \delta = 0.5,\ \ \upsilon = \omega = 1 \quad \& \quad b \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$$

Si $\beta = \delta = 0 \quad (e\,\upsilon^2 = a\,b \ \&\ c\,\upsilon^2 = d\,b)$ on a:

$$\begin{cases} a = \alpha\,\upsilon^2 \\ d = \gamma\,\upsilon^2 \\ e = \alpha\,b \\ c = \gamma\,b \end{cases} \quad \text{et} \quad \begin{cases} \tau^2 = (1-\alpha^2)\;\upsilon^2 = \upsilon^2 + \dfrac{a}{\varpi}(e\,b - a\omega^2) \\ \sigma^2 = \omega^2 - \gamma^2\upsilon^2 = \omega^2 - \dfrac{d}{\varpi}(c\,b + d\,\omega^2) \\ \tau\,\sigma\,\theta = b - \alpha\,\gamma\,\upsilon^2 = b + \dfrac{1}{\varpi}(c\,e\,\upsilon^2 - a\,d\,\omega^2) \end{cases}$$

Ce dernier système est bien définit si $\begin{cases} |\alpha| \leq 1 \\ \upsilon\,|\gamma| \leq \omega \end{cases}$.

Exemple 3

$$\beta = \delta = 0,\ \gamma = 0.1,\ \ \alpha = 0.5,\ \ \upsilon = \omega = 2 \quad \& \quad b \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$$

Si $\beta = \delta = \alpha = 0 \quad (e\,\upsilon^2 = a\,b,\ c\,\upsilon^2 = d\,b \ \&\ a\,\omega^2 = e\,b)$ on a:

$$\begin{cases} a = e = 0 \\ d = \gamma\,\upsilon^2 \\ c = \gamma\,b \end{cases} \quad \text{et} \quad \begin{cases} \tau^2 = \upsilon^2 \\ \sigma^2 = \omega^2 - \gamma^2\upsilon^2 = \omega^2 - \dfrac{1}{\varpi}(c^2\,\upsilon^2 + d^2\,\omega^2) \\ \tau\,\sigma\,\theta = b \end{cases}$$

Si $\alpha \neq 0$ tel que $\tau\,\gamma = \sigma\,\theta\,\alpha$, on retourne au cas des chaînes de Markov cachées.

On a :

$$Z_{t+1} = F_0\,Z_t + S_0\,W_{t+1} \tag{5.14}$$

Avec:

$$F_0 = \begin{pmatrix} \alpha & 0 \\ \gamma_0 & 0 \end{pmatrix}, \quad S_0 = \begin{pmatrix} \tau & 0 \\ \tau\,\gamma_0 & \sigma\sqrt{(1-\theta^2)} \end{pmatrix} \quad \text{où} \quad \gamma_0 = \frac{\gamma}{\alpha} \quad \text{et} \quad \tau\,\gamma_0 = \sigma\,\theta$$

Le système **(5.13)** est alors équivalent au système :

$$\begin{cases} X_{t+1} = \alpha\, X_t + \tau\, U_{t+1} \\ \\ Y_t = \gamma_0\, X_t + \sigma\sqrt{(1-\theta^2)}\, V_t \end{cases} \qquad \text{Où} \quad \tau\,\gamma_0 = \sigma\,\theta \qquad (5.15)$$

→ **Cas** $\gamma = 0 \quad (d\,\omega^2 = c\,b)$

On a :.

$$\begin{cases} a = \alpha\,\upsilon^2 + \beta\, b \\ d = \delta\, b \\ e = \alpha\, b + \beta\,\omega^2 \\ c = \delta\,\omega^2 \end{cases} \qquad \text{et} \qquad \begin{cases} \tau^2 = (1-\alpha^2)\,\upsilon^2 - \beta^2\omega^2 - 2\,\alpha\,\beta\, b = \upsilon^2 - \dfrac{1}{\varpi}(e^2\,\upsilon^2 + a^2\omega^2 - 2\,a\,e\,b) \\ \sigma^2 = (1-\delta^2)\,\omega^2 = \omega^2 - \dfrac{c}{\varpi}(c\,\upsilon^2 + d\,b) \\ \tau\,\sigma\,\theta = b - \delta(\alpha\, b + \beta\,\omega^2) = b - \dfrac{e}{\varpi}(c\,\upsilon^2 - d\,b) \end{cases}$$

Le dernier système est bien définit si $\begin{cases} 1 \geq |\delta| \\ \upsilon^2 \geq \alpha^2\upsilon^2 + \beta^2\omega^2 + 2\,\alpha\,\beta\, b \end{cases}$

Conclusion

L'étude a été menée pour plusieurs jeux de paramètres, et il apparaît que le filtrage de Kalman demeure très efficace lorsque le système linéaire gaussien classique est étendu au système linéaire gaussien couple.
Nous allons voir dans la seconde partie de ce chapitre que sous l'hypothèse $\gamma = 0$ le filtrage dans les modèles couples à sauts s'effectue de manière directe et il apparaitra que l'approximation du modèle général consistant à poser $\gamma = 0$ ne semble pas grossière.

5.2 Chaînes de Markov couples gaussiennes à sauts

Dans la seconde partie de ce chapitre nous traitons du problème du filtrage optimal dans les systèmes linéaires, markoviens couples et gaussiens, à sauts. Nous présentons un modèle général, qui est un processus de Markov gaussien couple à sauts, et nous étudions deux approximations du filtre optimal :

(i) filtre particulaire ;

(ii) filtre optimal exact, possible dans un modèle particulier.

Le premier peut ainsi être vu comme un filtrage approché dans un modèle exact, et le deuxième peut être vu comme un filtrage exact dans un modèle approché. Ces deux approches sont des contributions originales de la thèse.

5.2.1 Modèle général

On considère donc trois processus stochastiques $X_1^T = (X_1,...,X_T)$, $R_1^T = (R_1,...,R_T)$ et $Y_1^T = (Y_1,...,Y_T)$. Comme précédemment, les processus X_1^T et Y_1^T sont à valeurs respectivement dans $\Re^{n_x}$ et $\Re^{n_y}$, alors que le processus R_1^T est à valeurs dans un espace fini dit « espace des sauts » $\Omega = \{\lambda_1,...,\lambda_K\}$. Les deux processus X_1^T et R_1^T sont cachés et Y_1^T est observé. Le processus couple $Z_1^T = (X_1^T, Y_1^T)$ est gaussien conditionnellement à la chaîne des sauts R_1^T. Le problème consiste à estimer séquentiellement X_1^T et R_1^T à partir des réalisations observées du processus Y_1^T. Plus précisément, on souhaite calculer l'espérance conditionnelle $E[X_{t+1}\left|r_1^{t+1}, y_1^{t+1}\right.]$ et la probabilité $p(r_1^{t+1}\left|y_1^{t+1}\right.)$ à partir de l'espérance $E[X_t\left|r_1^t, y_1^t\right.]$, de la probabilité $p(r_1^{t+1}\left|y_1^{t+1}\right.)$, et l'observation y_{t+1}. Dans cette démarche le filtrage est optimal au sens de l'erreur quadratique moyenne et donc on souhaitera retenir comme estimateur pour x_t l'espérance conditionnelle $\hat{x}_t = E[X_t\left| y_1^t\right.]$. Par ailleurs, les sauts sont estimés par la méthode MPM. On a donc à l'instant $(t+1)$:

$$\hat{r}_{t+1} = \arg\underset{r_{t+1}\in\Omega}{Max}[p(r_{t+1}\left| y_1^{t+1}\right.)] \tag{5.16}$$

$$\hat{x}_{t+1} = E[X_{t+1}\left| y_1^{t+1}\right.] = \sum_{r_1^{t+1}\in\Omega^{t+1}} p(r_1^{t+1}\left|y_1^{t+1}\right.) E[X_{t+1}\left|r_1^{t+1}, y_1^{t+1}\right.] \tag{5.17}$$

Pour les mêmes raisons que dans le cas des chaînes triplets gaussiens à sauts markoviens vues au chapitre précédent, dans le cas des chaînes couples avec sauts le calcul de ces deux derniers estimateurs n'est pas possible en général car le nombre d'opération est exponentiel en fonction du temps. Notre contribution ici est double. La première est de proposer une extension du filtrage particulaire valable dans les modèles à sauts vu dans le chapitre précédent. La deuxième originalité consiste en filtrage exact, mais dans un modèle particulier, lorsque les données suivent le modèle couple à sauts général. On compare alors, via les simulations, l'efficacité des deux approches pour résoudre le problème de la restauration.

Dans l'approche classique on fait l'hypothèse que le processus des sauts R_1^T est une chaîne de Markov et la loi du couple (X_1^T, Y_1^T) conditionnellement aux sauts est celle d'un système linéaire gaussien suivant :

$$\begin{cases} X_{t+1} = \phi(R_{t+1})\, X_t + \Phi(R_{t+1})\, U_{t+1} \\ \\ Y_t = \psi(R_t)\, X_t + \Psi(R_t)\, V_t \end{cases} \tag{5.18}$$

avec $\phi(R_{t+1})$, $\psi(R_t)$, $\Phi(R_{t+1})$ et $\Psi(R_t)$ des matrices de taille adéquates et $(U_t)_{t\in\{2,...,T\}}$, $(V_t)_{t\in\{2,...,T\}}$ deux suites de vecteurs aléatoires gaussiens mutuellement indépendant centrés et

de variance-covariance la matrice identité. On considère également que pour tout $t \in \{1,\ldots,T-1\}^{'}$, $W_{t+1} = (U_{t+1}, V_{t+1})^{'}$ est indépendant de $Z_1^t = (X_1^t, Y_1^t)$ et les variables R_{t+1} et $Z_t = (X_t, Y_t)$ sont indépendantes conditionnellement à R_t.
On a ainsi, conditionnellement à R_1^T, le système classique (5.11) de la sous-section précédente.

Rappelons que le filtrage optimal avec une complexité raisonnable dans le système à saut (5.18) n'a pas été proposé jusqu'à maintenant et il est fait appel à diverses méthodes d'approximation, dont le filtrage particulaire.

Comme dans la sous-section précédente, il est possible de mettre le système (5.18) sous forme matricielle. En effet, on peut remplacer la seconde équation par une autre équivalente. On a :

$$Y_{t+1} = \psi(R_{t+1})\,\phi(R_{t+1})\,X_t + \psi(R_{t+1})\,\Phi(R_{t+1})\,U_{t+1} + \Psi(R_{t+1})\,V_{t+1}$$

On peut alors réécrire (5.18) sous la forme matricielle suivante :

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \end{bmatrix} = \begin{bmatrix} \phi(R_{t+1}) & 0 \\ \phi(R_{t+1})\,\psi(R_{t+1}) & 0 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \end{bmatrix} + \begin{bmatrix} \Phi(R_{t+1}) & 0 \\ \psi(R_{t+1})\,\Phi(R_{t+1}) & \Psi(R_{t+1}) \end{bmatrix} \begin{bmatrix} U_{t+1} \\ V_{t+1} \end{bmatrix} \qquad (5.19)$$

Comme dans la sous-section précédente (sans sauts), on généralise ce système en considérant l'équation matricielle suivante :

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \end{bmatrix} = \begin{bmatrix} \alpha(R_{t+1}) & \beta(R_{t+1}) \\ \gamma(R_{t+1}) & \delta(R_{t+1}) \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \end{bmatrix} + \begin{bmatrix} S_{11}(R_{t+1}) & S_{12}(R_{t+1}) \\ S_{21}(R_{t+1}) & S_{22}(R_{t+1}) \end{bmatrix} \begin{bmatrix} U_{t+1} \\ V_{t+1} \end{bmatrix} \qquad (5.20)$$

Le modèle (5.20) sera appelé « Modèle conditionnellement gaussien à sauts markoviens (MCGSM).

On pose pour la suite :

$$F(R_{t+1}) = \begin{bmatrix} \alpha(R_{t+1}) & \beta(R_{t+1}) \\ \gamma(R_{t+1}) & \delta(R_{t+1}) \end{bmatrix}, \text{ et } S(R_{t+1}) = \begin{bmatrix} S_{11}(R_{t+1}) & S_{12}(R_{t+1}) \\ S_{21}(R_{t+1}) & S_{22}(R_{t+1}) \end{bmatrix}.$$

On peut alors écrire le modèle sous la forme linéaire compacte suivante :

$$Z_{t+1} = F(R_{t+1})Z_t + S(R_{t+1})W_{t+1}$$

La loi de $Z_1^T = (X_1^T, Y_1^T)$ conditionnellement à R_1^T est alors celle d'un modèle de Markov couple gaussien. Cependant comme montré dans [3] et rappelé dans la première partie de ce paragraphe, le filtrage exact « l'algorithme de Kalman » peut être étendu en conservant les mêmes avantages en terme de complexité algorithmique, aux modèles de Markov couples gaussiens.

On s'intéresse donc au problème du filtrage optimal dans le système à sauts général (5.20). Dans un premier temps, nous proposons une extension au système (5.20) du filtrage particulaire utilisé dans le système classique (5.18). Dans un deuxième temps nous proposerons un cas particulier du système (5.20) dans lequel le calcul exact du filtre optimal avec une complexité linéaire en temps est possible.
Notons dès à présent que ce cas particulier est le suivant :

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \end{bmatrix} = \begin{bmatrix} \alpha(R_{t+1}) & \beta(R_{t+1}) \\ 0 & \delta(R_{t+1}) \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \end{bmatrix} + \begin{bmatrix} S_{11}(R_{t+1}) & S_{12}(R_{t+1}) \\ S_{21}(R_{t+1}) & S_{22}(R_{t+1}) \end{bmatrix} \begin{bmatrix} U_{t+1} \\ V_{t+1} \end{bmatrix} \tag{5.21}$$

Notons également le cas particulier symétrique suivant :

$$\begin{bmatrix} X_{t+1} \\ Y_{t+1} \end{bmatrix} = \begin{bmatrix} \alpha(R_{t+1}) & 0 \\ \gamma(R_{t+1}) & \delta(R_{t+1}) \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \end{bmatrix} + \begin{bmatrix} S_{11}(R_{t+1}) & S_{12}(R_{t+1}) \\ S_{21}(R_{t+1}) & S_{22}(R_{t+1}) \end{bmatrix} \begin{bmatrix} U_{t+1} \\ V_{t+1} \end{bmatrix} \tag{5.22}$$

Le modèle (5.22) reste une généralisation du modèle classique (5.19) ce n'est pas le cas du modèle (5.21).
Notons que dans le modèle classique (5.19) le couple (X_1^T, R_1^T) est markovien et le couple (R_1^T, Y_1^T) ne l'est pas ; dans le modèle (5.21) la situation est inverse : le couple (R_1^T, Y_1^T) est markovien et le couple (X_1^T, R_1^T) ne l'est pas. Dans le général (5.20) aucun des deux couples (X_1^T, R_1^T), (R_1^T, Y_1^T) n'est nécessairement markovien.

A $R_1^T = r_1^T$ donné, le filtrage exact optimal au sens de l'erreur moyenne quadratique est faisable dans les modèles (5.20), (5.21) et (5.22) ; il est alors intéressant de se poser la question suivante. Supposons que les données sont simulées selon le modèle général (5.20), et ensuite restaurées, à sauts connus, avec les modèles (5.21) et (5.22). La différence entre ces restaurations et celle obtenue avec le modèle (5.20) nous renseignera sur l'accroissement des erreurs lorsque l'on utilise (5.21)ou (5.22) au lieu du vrai modèle (5.20).

Comme mentionné précédemment, le filtrage optimal exact n'est pas possible dans le modèle (5.19) et des approximations, comme celles utilisant le filtrage particulaire, sont indispensables. Bien entendu, le filtrage exact n'est pas non plus possible dans le modèle (2.20), qui généralise le modèle (5.19).

Nous proposons dans la suite de cette sous-section trois contributions suivantes :

(i) Nous étendons le filtrage particulaire, devenu une méthode de filtrage classique dans le modèle (5.19), au modèle général (5.20) et étudions son comportement dans ce nouveau contexte ;

(ii) Nous montrons que le filtrage exact avec complexité linéaire en temps est possible dans le modèle (5.21) et nous donnons les formules correspondantes. En effet, nous montrons que ce modèle est un cas particulier du modèle récemment présenté dans [9] ;

(iii) Nous proposons une étude numérique dont l'objectif est de comparer, en traitant des données simulées par le modèle (5.20), l'efficacité du nouveau filtrage particulaire (voir point (i) ci-dessus) avec le filtrage exact utilisant le modèle (2.21) obtenu à partir du modèle (5.20) en remplaçant $\gamma(R_{t+1})$ par 0. Ainsi l'efficacité d'une méthode approximative (filtrage particulaire) appliquée à un modèle exact est comparée à l'efficacité d'une méthode exacte appliquée à un modèle approché.

5.2.2 Filtrage particulaire dans le modèle couple gaussien général

Nous présentons brièvement dans cette section une extension du filtrage particulaire utilisé dans les modèles classique (5.19) au modèle MCGSM (5.20). De manière générale, le problème vient du fait que dans le modèle (5.19) – et donc également dans le modèle (5.20) – la loi $p(r_{t+1}\big|\, y_1^{t+1})$ ne peut pas être calculée récursivement et doit être approchée. La méthode brièvement décrite ci-dessous est une extension relativement directe de celles proposées dans les modèles classiques dans les références [6], [10].
Considérons le modèle (5.20) et supposons que la réalisation $R_1^T = r_1^T$ est donnée. La loi de $Z_1^T = (X_1^T, Y_1^T)$ conditionnelle à $R_1^T = r_1^T$ est alors celle d'une chaîne de Markov gaussienne couple ; il en résulte que $p(x_{t+1}\big|\, r_1^{t+1}, y_1^{t+1})$ est calculable à partir de $p(x_t\big|\, r_1^{t+1}, y_1^t)$ et y_{t+1} en utilisant le filtre exact « algorithme de Kalman généralisé au modèle couple », comme rappelé ci-dessus. Par ailleurs, nous avons la formule suivante, qui donne $p(r_1^{t+1}\big|y_1^{t+1})$ à partir de $p(r_1^t\big|y_1^t)$ et y_{t+1} :

$$
\begin{aligned}
p(r_1^{t+1}\big|\, y_1^{t+1}) &= \frac{p(y_{t+1}\big|\, r_1^{t+1}, y_1^t)\, p(r_{t+1}\big|\, r_t)}{p(y_{t+1}\big|\, y_1^t)}\, p(r_1^t\big|\, y_1^t) \\
&= \frac{p(y_{t+1}\big|\, r_1^{t+1}, y_1^t)\, p(r_{t+1}\big|\, r_t)}{\sum_{r_1^{t+1}} p(y_{t+1}\big|\, r_1^{t+1}, y_1^t)\, p(r_{t+1}\big|\, r_t)\, p(r_1^t\big|\, y_1^t)}\, p(r_1^t\big|\, y_1^t)
\end{aligned}
\tag{5.23}
$$

Cette dernière égalité est alors utilisée séquentiellement pour approcher $p(r_{t+1}\big|\, y_1^{t+1})$ par un ensemble de masses de Dirac pondérées, selon la technique classique du filtrage particulaire. Pour commencer, $p(r_1^t\big|y_1^t)$ est approché avec

$$
p(r_1^t\big|y_1^t) \approx \hat{p}_{N_p}(r_1^t\big|y_1^t) = \frac{1}{N_p}\sum_{i=1}^{N_p} \delta\{r_1^{t,i}\}, \tag{5.24}
$$

où N_p est le nombre de trajectoires simulées. Connaissant $p(r_1^t\big|\, y_1^t)$, $p(x_t\big|\, y_1^t)$ est donné par

$$
p(x_t\big|y_1^t) = \sum_{r_1^t} p(r_1^t\big|y_1^t)\, p(x_t\big|r_1^t, y_1^t) \tag{5.25}
$$

Reportant (5.24) dans (5.25) on obtient

$$p(x_t|y_1^t) \approx \hat{p}_{N_p}(x_t|y_1^t) = \frac{1}{N_p}\sum_{i=1}^{N_p} p(x_t|r_1^{t,i}, y_1^t). \qquad (5.26)$$

Précisons que les paramètres des lois $p(y_{t+1}|r_1^{t+1}, y_1^t)$ dans (5.23) et ceux des lois $p(x_t|r_1^{t,i}, y_1^t)$, pour $i \in \{1,...,N_p\}$, dans (5.26) sont calculés séquentiellement par l'algorithme de filtrage donné dans la première partie de ce chapitre « extension de l'algorithme de Kalman pour le cas du modèle couple ».

5.2.3 Filtrage exact dans le modèle approché

Considérons le modèle (5.21). Le problème est donc de calculer $p(r_{t+1}|y_1^{t+1})$ et $E[X_{t+1}|r_{t+1}, y_1^{t+1}]$ à partir de $p(r_t|y_1^t)$, $E[X_t|r_t, y_1^t]$, et y_{t+1}. La loi $p(r_{t+1}|y_1^{t+1})$ est obtenue à partir de $p(r_t|y_1^t)$ et y_{t+1} classiquement, en utilisant le fait que (R_1^T, Y_1^T) est une chaîne de Markov cachée. Par ailleurs, l'égalité (5.21) implique la possibilité de l'écriture suivante:

$$X_{t+1} = A(R_{t+1})X_t + B(R_{t+1})Y_t + C(R_{t+1})Y_{t+1} + D(R_{t+1})W_{n+1}, \qquad (5.27)$$

où $A(R_{t+1}), B(R_{t+1}), C(R_{t+1})$, et $D(R_{t+1})$ sont des matrices de taille adéquate dépendant de r_{t+1} (calculées classiquement, en utilisant le conditionnement dans le cas gaussien, à partir de (5.20), et W_2, …, W_T sont des vecteurs gaussiens indépendants entre eux, centrés, de matrices de covariance unité et telle que pour tout $t = 1,...,T-1$, W_{t+1} est indépendant de $(X_1^t, R_1^{t+1}, Y_1^t)$. En prenant l'espérance de **(5.27)** conditionnelle à $(R_{t+1}, Y_1^{t+1}) = (r_{t+1}, y_1^{t+1})$ nous avons, en utilisant le fait que dans le modèle **(5.21)** on a $E[X_t|r_t, r_{t+1}, y_1^{t+1}] = E[X_t|r_t, y_1^t]$ ((R_{t+1}, Y_{t+1}) et X_t sont indépendantes conditionnellement à (R_t, Y_t)) :

$$\begin{aligned} E[X_{t+1}|r_{t+1}, y_1^{t+1}] &= \sum_{r_t} E[X_{t+1}|r_t,\ r_{t+1}, y_1^{t+1}]p(r_t|y_1^{t+1}) \\ &= A(r_t^{t+1})[\sum_{r_t} E[X_t|r_t,\ y_1^t]p(r_t|y_1^{t+1})] + B(r_t^{t+1})y_t + C(r_t^{t+1})y_{n+1} \end{aligned} \qquad (5.28)$$

$$= A(r_t^{t+1})\frac{\sum_{r_t} E[X_t|r_t,\ y_1^t]p(r_t|y_1^t)p(y_{t+1}|r_t, y_t)}{\sum_{r_t} p(r_t|y_1^t)p(y_{t+1}|r_t, y_t)} + B(r_t^{t+1})y_t + C(r_t^{t+1})y_{t+1}$$

Ce qui nous donne le filtre.

5.3 Expérimentations

On considère X_1^N et Y_1^N à valeurs dans R . Le triplet $T_1^N = (X_1^N, R_1^N, Y_1^N)$ est supposé stationnaire ; sa loi est donc définie par la loi de (T_1, T_2), que considérons sous la forme

$p(t_1,t_2) = p(r_1,r_2)\,p(x_1,y_1,x_2,y_2|r_1,r_2)$. Oublions temporairement la dépendance de $p(x_1,y_1,x_2,y_2|r_1,r_2)$ des sauts (r_1,r_2). Supposons que toutes les moyennes de la loi gaussienne $p(x_1,y_1,x_2,y_2)$ sont nulles et toutes les variances sont égales à 1. La loi $p(x_1,y_1,x_2,y_2)$ est alors donnée par la matrice de covariance

$$\Gamma^{Z_1^2} = \begin{array}{c} \\ X_1 \\ Y_1 \\ X_2 \\ Y_2 \end{array} \begin{array}{c} \begin{array}{cccc} X_1 & Y_1 & X_2 & Y_2 \end{array} \\ \begin{bmatrix} 1 & b & a & d \\ b & 1 & e & c \\ a & e & 1 & b \\ d & c & b & 1 \end{bmatrix} \end{array} = \begin{bmatrix} \Gamma & A^T \\ A & \Gamma \end{bmatrix}, \text{ avec } \Gamma = \begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix} \text{ et } A = \begin{bmatrix} a & e \\ d & c \end{bmatrix}$$

Donc la loi $p(x_1,y_1,x_2,y_2)$ est donnée par les cinq co-variances a,b,c,d,e, avec la condition que $\Gamma^{Z_1^2}$ est définie positive. Le graphe de dépendance est donné sur la Figure 5.1.

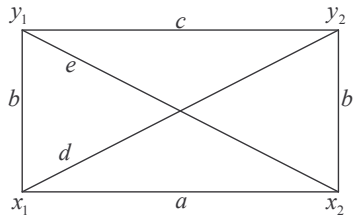


Figure 5.1 Graphe de dépendance de (X_1,Y_1,X_2,Y_2)

L'équation (5.20) prend la forme

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix},$$

avec

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} = \begin{bmatrix} a & e \\ d & c \end{bmatrix} \begin{bmatrix} 1 & b \\ b & 1 \end{bmatrix}^{-1} = \frac{1}{1-b^2} \begin{bmatrix} a-eb & -ab+e \\ d-cb & -db+c \end{bmatrix},$$

et

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}^T = \Gamma - A\Gamma^{-1}A^T$$

Donc le modèle (2.20) est un modèle (5.21) si $S_{21} = 0$, ce qui donne $d - cb = 0$. Cela signifie que pour $(R_1,R_2) = (r_1,r_2)$ donnés, les covariances $a,b,c,$ et e peuvent être les mêmes dans les deux modèles et la différence se situe au niveau de d : elle est quelconque dans le modèle (5.20) général, alors qu'elle est égale à cb dans le modèle (2.21) (on peut noter que cela signifie que X_1 et Y_2 sont indépendantes conditionnellement à Y_1). Il est important de

remarquer que cela implique que les lois de (R_1, R_2), (X_1, Y_1, X_2), et (Y_1, X_2, Y_2) sont strictement les mêmes dans le modèle (5.20) et son « approximation », consistant à prendre pour un « nouveau » d la quantité cb. Sachant que les deux modèles sont stationnaires cela implique que les lois de (R_n, R_{n+1}), (X_n, Y_n, X_{n+1}), et (Y_n, X_{n+1}, Y_{n+1}), sont les mêmes pour tout $n = 1, \ldots, N-1$.

Considérons le modèle (5.20) complet, avec la loi de la chaîne de Markov stationnaire des sauts définie par $p(r_1, r_2)$. On suppose toutes les moyennes nulles. La loi du processus triplet est donc donnée par $p(r_1, r_2)$ et les matrices de covariance

$$\Gamma^{Z_1^2}(r_1^2) = \begin{array}{c} \\ X_1 \\ Y_1 \\ X_2 \\ Y_2 \end{array} \begin{array}{c} \begin{array}{cccc} \overset{X_1}{} & \overset{Y_1}{} & \overset{X_2}{} & \overset{Y_2}{} \end{array} \\ \begin{bmatrix} 1 & b(r_1^2) & a(r_1^2) & d(r_1^2) \\ b(r_1^2) & 1 & e(r_1^2) & c(r_1^2) \\ a(r_1^2) & e(r_1^2) & 1 & b(r_1^2) \\ d(r_1^2) & c(r_1^2) & b(r_1^2) & 1 \end{bmatrix} \end{array} = \begin{bmatrix} \Gamma(r_1^2) & A(r_1^2)^T \\ A(r_1^2) & \Gamma(r_1^2) \end{bmatrix}, \tag{5.29}$$

avec

$$\Gamma = \begin{bmatrix} 1 & b(r_1^2) \\ b(r_1^2) & 1 \end{bmatrix} \text{ et } A = \begin{bmatrix} a(r_1^2) & e(r_1^2) \\ d(r_1^2) & c(r_1^2) \end{bmatrix}.$$

Cela donne

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} \alpha(r_n^{n+1}) & \beta(r_n^{n+1}) \\ \gamma(r_n^{n+1}) & \delta(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} + \begin{bmatrix} S_{11}(r_n^{n+1}) & S_{12}(r_n^{n+1}) \\ S_{21}(r_n^{n+1}) & S_{22}(r_n^{n+1}) \end{bmatrix} \begin{bmatrix} U_{n+1} \\ V_{n+1} \end{bmatrix} \tag{5.30}$$

Finalement, la distribution du modèle est donnée par la loi $p(r_1, r_2)$ et les corrélations $a(r_1^2)$, $b(r_1^2), c(r_1^2), d(r_1^2)$, et $e(r_1^2)$.

Dans les simulations ci-après on va considérer $b(r_1^2) = b$, $a(r_1^2) = a(r_2)$, $c(r_1^2) = c(r_2)$, $d(r_1^2) = d(r_2)$, et $e(r_1^2) = e(r_2)$. On va donc simuler des réalisations du modèle $T_1^N = (T_1, ..., T_N)$ dans toute sa généralité (5.30), et estimer, par des filtrages adaptatifs, (R_1^N, X_1^N) à partir de Y_1^N. Nous allons considérer trois méthodes:

(i) la méthode de référence, appelée MR, dans laquelle $R_1^N = r_1^N$ est considérée comme connue. Sachant donc $R_1^N = r_1^N$, la seule variable cachée $X_1^N = x_1^N$ est estimée classiquement par le filtre de Kalman;

(ii) la méthode utilisant l'extension du filtrage particulaire au modèle (5.20) décrite dans la sous-section 5.2.2, notée FP ;

(iii) la nouvelle méthode, note NM, consistant à poser $\gamma(r_n) = 0$ pour tout r_n, et à appliquer la méthode exacte décrite dans la sous-section précédente.

Dans toutes les expérimentations on suppose que chaque r_n prend pour valeur 0 ou 1.

On va considérer deux séries d'expériences.

Dans la première la loi $p(r_1, r_2)$ est fixée et donnée par $p(r_1 = 0, r_2 = 0) = p(r_1 = 1, r_2 = 1) = 0.4$, and $p(r_1 = 0, r_2 = 1) = p(r_1 = 1, r_2 = 0) = 0.1$. Alors nous considérons cinq cas correspondant à cinq modèles suivants. Dans tous les cas, nous avons

$b = 0.3$, $a(0) = 0.1$, $a(1) = 0.5$, $c(0) = 0.4$, $c(1) = 0.9$, $e(0) = 0.75$, $e(1) = 0.33$.

Alors on choisit cinq couples $(d^i(0), d^i(1))$, $i = 1, ..., 4$, tells que les covariances $\gamma(r_n) = [d(r_n) - bc(r_n)]/(1 - b^2)$ correspondantes vérifient :

Cas 1 : $\gamma(0) = \gamma(1) = 0.0$;

Cas 2 : $\gamma(0) = \gamma(1) = 0.1$;

Cas 3 : $\gamma(0) = \gamma(1) = 0.2$;

Cas 4 : $\gamma(0) = \gamma(1) = 0.3$;

Cas 5 : $\gamma(0) = \gamma(1) = 0.4$.

Ainsi nous faisons croitre $\gamma = \gamma(0) = \gamma(1)$, ce qui a pour effet le fait que le modèle (5.20) selon lequel on simule les données est de plus en plus différent du modèle "approché" (5.21) qui est utilisé pour appliquer la méthode NM. En principe, cette croissance devrait implique la dégradation des résultats obtenus avec la NM et l'objet principal de l'expérimentation est d'étudier cette dégradation. Ainsi que nous nous allons le voir, le résultat intéressant est que cette dégradation n'est pas significative.

Notons qu'il n'a pas été possible d'aller au-delà de $\gamma = 0.4$ car la matrice (5.29) n'est plus définie positive.

Donc dans chaque cas nous simulons $T_1^N = t_1^N = (x_1^N, r_1^N, y_1^N)$ selon le modèle (5.30) et (x_1^N, r_1^N) sont cherchés avec les trois méthodes MR (où r_1^N est donné), FP, et NM. La différence entre x_1^N et l'estimée $\hat{x}_1^N$ est mesurée par $\varepsilon(x_n, \hat{x}_n) = \frac{1}{N}\sqrt{\sum_{n=1}^{N}(x_n - \hat{x}_n)^2}$, et la différence entre r_1^N et l'estimée $\hat{r}_1^N$ est mesurée par le taux d'erreurs $\tau(r_n, \hat{r}_n) = \frac{1}{N}\sqrt{\sum_{n=1}^{N}|r_n - \hat{r}_n|}$ (rappelons que les r_n valent 0 ou 1).
La longueur des échantillons considérés est $N = 200$, et dans la méthode FP nous avons utilisé 500 particules.

Les résultats obtenus sont présentés dans le Tableau 5.1, et deux familles de trajectoires sont présentées sur les Figures 5.2 et 5.3.

	Case 1	Case 2	Case 3	Case 4	Case 5
γ	0.0	0.1	0.2	0.3	0.4
	Erreur $\varepsilon(x_n,\hat{x}_n)=\frac{1}{N}\sqrt{\sum_{n=1}^{N}(x_n-\hat{x}_n)^2}$				
MR	0.0576	0.0574	0.0587	0.0584	0.0576
FP	0.0605	0.0597	0.0610	0.0604	0.0609
NM	0.0595	0.0593	0.0606	0.0600	0.0597
	Tau d'erreurs $\tau(r_n,\hat{r}_n)=\frac{1}{N}\sqrt{\sum_{n=1}^{N}\lvert r_n-\hat{r}_n\rvert}$				
MR	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
FP	25.80%	25.00 %	26.10 %	27.45%	25.30%
NM	25.15%	25.35 %	25.70 %	28.10%	25.65%
	Temps de calculs				
MR	0.15	0.14	0.16	0.16	0.13
FP	123.58	113.17	108.18	157.28	113.56
NM	0.50	0.48	0.29	0.75	0.47

Tableau 5.1. Les erreurs quadratiques et les taux d'erreurs obtenus avec les Méthodes MR, FP, et NM.

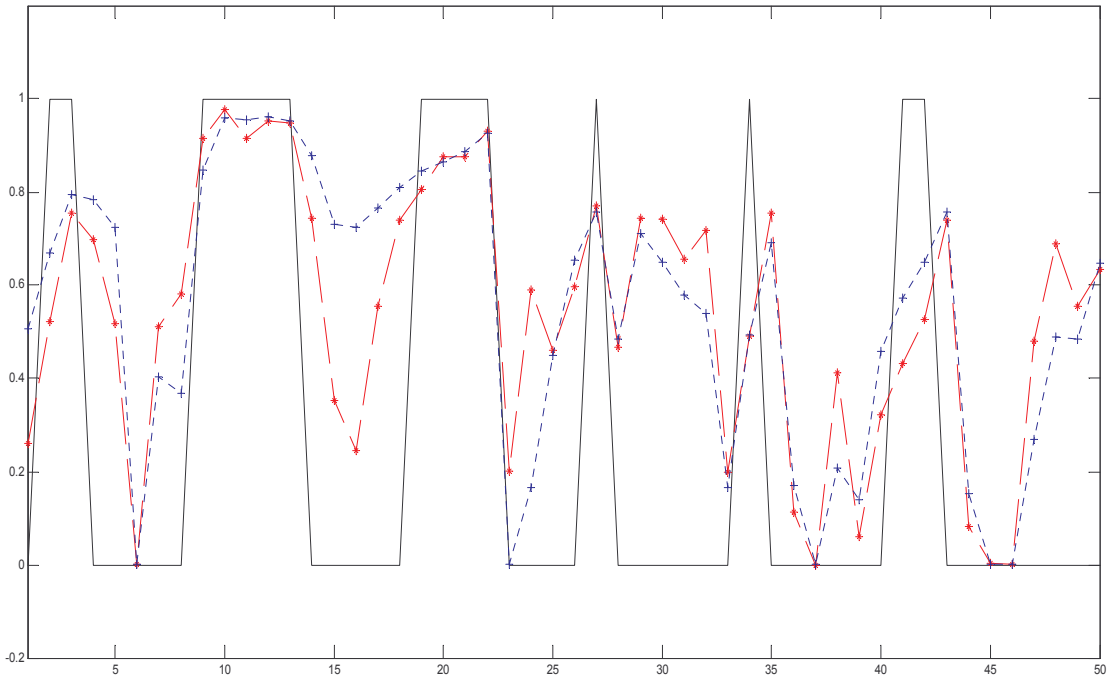


Figure 5.2. Réalisation de R_1^N, estimée de la loi $p(r_t=1|y_1^t)$ par FP (en rouge, trais longs), et son calcul dans NM (en bleu, trais courts).

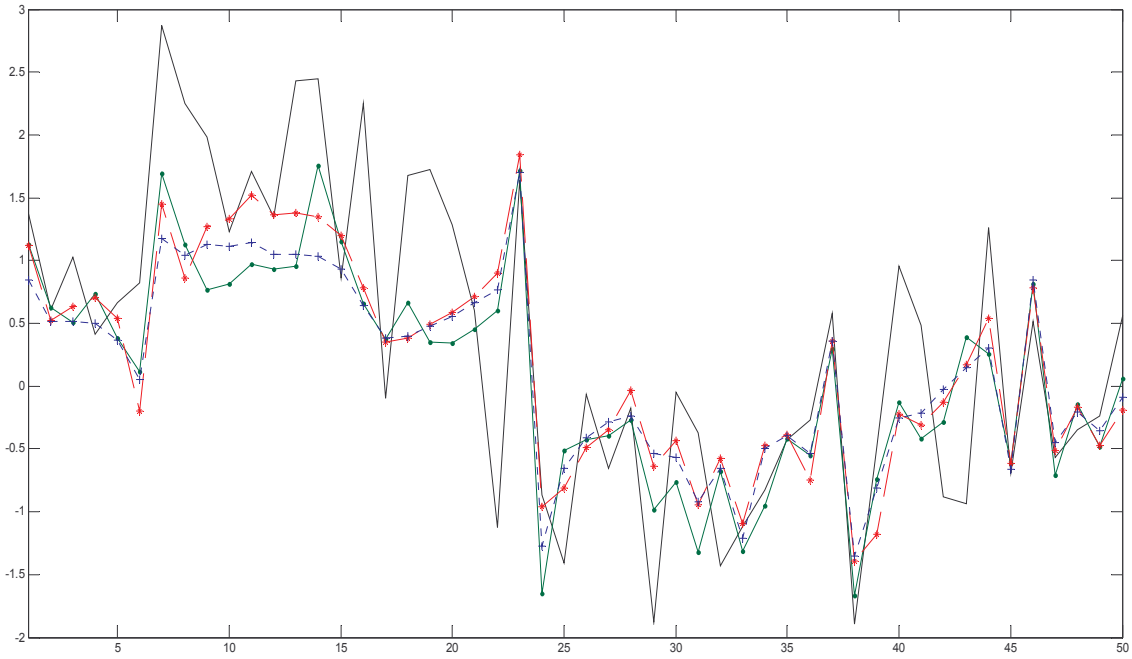


Figure 5.3. Cinquante derniers points de la vraie trajectoire (en noir, trait plein), restauration par MR (en vert), restauration par FP en rouge (traits longs), et restauration par NM (en bleu, traits cours).

Nous proposons les conclusions suivantes:

1. On observe que lorsque γ croît, ce qui fait que le modèle (5.21) utilise dans la méthode NM s'éloigne du modèle (5.20) utilise pour simuler les données, les résultats obtenus avec la méthode NM ne se dégradent pas de manière significative. De plus, malgré le haut niveau de bruit les résultats obtenus avec les méthodes NM et FP sont proches des résultats « optimaux » obtenus avec la méthode MR. Cela est très encourageant car cela montre que le filtrage optimal est robuste par rapport au paramètre γ, au moins dans le contexte considéré.

2. Le filtrage particulaire assure également de très bonnes performances, qui sont très proches de celles du filtre optimal (la méthode MR). Cependant, le temps d'exécution, environ deux cents fois plus cours dans la méthode NM, plaide en faveur de cette dernière.

Dans la deuxième série d'expérimentions nous considérons le cas $\gamma = 0.4$, qui est donc le plus défavorable à la nouvelle méthode NM, et nous faisons varier $p(r_1, r_2)$. Soit $p(r_1, r_2)$ écrit sous la forme suivante $p(r_1, r_2) = p(r_1)p(r_2|r_1)$. Dans tous les cas $p(r_1) = 0.5$, et $p(r_2 = 0|r_1 = 0) = p(r_2 = 1|r_1 = 1) = q$ varie. L'objectif de cette expérience est de regarder si la variation des paramètres de la chaîne des sauts modifie les conclusions ci-dessus.

Les résultats sont présentés dans le Tableau 5.2.

	$q = 0.9$	$q = 0.8$	$q = 0.7$	$q = 0.6$	$q = 0.5$	$q = 0.4$	$q = 0.3$	$q = 0.2$
	Erreur $\varepsilon(x_n, \hat{x}_n) = \frac{1}{N}\sqrt{\sum_{n=1}^{N}(x_n - \hat{x}_n)^2}$							
MR	0.0538	0.0508	0.0544	0.0589	0.0519	0.0513	0.0569	0.0505
FP	0.0546	0.0522	0.0563	0.0619	0.0543	0.0548	0.0633	0.0531
NM	0.0551	0.0514	0.0553	0.0609	0.0547	0.0535	0.0605	0.0530
	Tau d'erreurs $\tau(r_n, \hat{r}_n) = \frac{1}{N}\sqrt{\sum_{n=1}^{N}\lvert r_n - \hat{r}_n\rvert}$							
FP	20%	23%	21%	30%	26%	26%	28%	22%
NM	24%	25%	22%	30%	25%	27%	27%	23%

Tableau 5.2. Les erreurs quadratiques et les taux d'erreurs obtenus avec les méthodes MR, FP, et NM. Dans tous les cas $\gamma = 0.4$, et $p(r_2 = 0 | r_1 = 0) = p(r_2 = 1 | r_1 = 1) = q$.

Nous constatons que les conclusions générales restent vraies indépendamment des paramètres de la chaîne des sauts.
Bien entendu, les expérimentations présentées sont très partielles et d'autres études sont nécessaires pour valider l'intérêt des nouveaux modèles.

5.4 Conclusion

Nous avons étendu le filtrage particulaire, utilisé dans les modèles classiques (5.19), à un modèle général (5.20) dans lequel le couple (chaîne cachée, chaîne observée) est markovien conditionnellement aux sauts. La nouveauté de ce modèle est que la chaîne cachée n'est plus nécessairement markovienne conditionnellement aux sauts. Ensuite, nous avons considéré un cas particulier de cette généralisation dans lequel le filtrage exact avec une complexité linéaire en temps est possible. Il s'agit d'une contribution originale, même si le modèle présenté peut également être vu comme appartenant à la famille « modèles cachés conditionnellement linéaires à sauts markoviens » proposée dans [9]. Nous avons présenté une étude des simulations pour comparer les deux méthodes (filtrage particulaire dans le modèle général et filtrage exact dans un modèle particulier, vu comme un modèle approché du modèle général).
Les conclusions générales de notre étude sont les suivantes :

(a) Le très bon comportement du filtrage particulaire, avéré dans les modèles classiques (5.19), est préservé dans le modèle général (5.20) ;

(b) La qualité du filtrage, ainsi que celle de l'estimation des sauts, par la nouvelle méthode (5.21) mentionnée ci-dessus sont comparables à celles obtenues avec le filtrage particulaire, même dans les cas où dans le modèle général servant à générer les données les valeurs de $\gamma(\lambda)$ pour $\lambda \in \Omega$ sont très différentes de 0 (la méthode est robuste par rapport au paramètre γ) ;

(c) Le temps d'exécution de la méthode utilisant le modèle (5.20) est environ deux cent fois plus court que celle utilisant le filtrage particulaire (pour 500 particules).

Conclusion générale

Nous avons abordé dans cette thèse le problème de filtrage statistique optimal dans des systèmes markoviens à sauts. Nous avons apporté les contributions suivantes :

(i) dans le modèle classique, où l'on utilise habituellement le filtrage particulaire, nous avons proposé deux méthodes sous-optimales originales. Leur intérêt, par rapport au filtrage particulaire, réside essentiellement dans leur rapidité. En effet, dans le cadre des simulations effectuées, elles se montrent plus rapide et de l'efficacité comparable à celle de filtrage particulaire ;

(ii) nous avons étendu le modèle classique à un modèle plus général et nous avons proposé de généraliser le filtrage particulaire classique au nouveau modèle. Par ailleurs, un cas particulier original du nouveau modèle, dans lequel le filtrage optimal exact est possible, a été proposé et étudié. Ce dernier point nous semble particulièrement prometteur car la nouvelle méthode utilisant ce modèle est très rapide et, du moins dans le cadre de nos simulations, donne des résultats comparables aux résultats optimaux obtenus à sauts connus.

(iii) nous avons proposé des méthodes d'estimation des paramètres de type EM et ICE adaptatives, permettant un filtrage non supervisé. En particulier, la nouvelle m éthode de type ICE est applicable dans les modèles récents avec des bruits à mémoire longue.

Une réflexion théorique sur les nouveaux modèles et la qualité des résultats qu'ils procurent est probablement parmi les perspectives le plus intéressantes.

Bibliographie

[1] N. Abbassi, D. Benboudjema, and W. Pieczynski, Kalman filtering approximations in triplet Markov Gaussian switching models, Statistical Signal Processing Workshop (SSP 2011), Nice, France, June 28-30, 2011.

[2] C. Andrieu, C. M. Davy, and A. Doucet, Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions, IEEE Trans. on Signal Processing, Vol. 51, No. 7, pp. 1762-1770, 2003.

[3] D. Benboudjema, M. Malainin, and W. Pieczynski, Exact Kalman filtering in pairwise Gaussian switching systems, Applied Stochastic Models and Data Analysis (ASMDA 2011), June 7-10, Roma, Italy, 2011.

[4] O. Cappé, E. Moulines, and T. Ryden, Inference in hidden Markov models, Springer, 2005.

[5] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, Discrete time Markov jump linear systems, New York, Springer-Verlag, 2005.

[6] A. Doucet, N. J. Gordon, and V. Krishnamurthy, Particle filters for state estimation of Jump Markov Linear Systems, IEEE Trans. on Signal Processing, Vol. 49, pp.613-624, 2001.

[7] P. Giordani, R. Kohn, and D. van Dijk, A unified approach to nonlinearity, structural change, and outliers, Journal of Econometrics, 137, 112-133, 2007.

[8] W. Pieczynski and F. Desbouvries, Kalman filtering using Pairwise Gaussian Models, International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong-Kong, 2003.

[9] W. Pieczynski, Exact filtering in Markov marginal switching hidden models, Comptes Rendus Mathematique, Vol. 349, No. 9-10, pp. 587-590, May 2011.

[10] J. K. Tugnait, Adaptive estimation and identification for discrete systems with Markov jump parameters, IEEE Trans. on Automatic Control, AC-25, 1054-1065, 1982.

[11] H. W. Sorenson, Kalman filtering techniques, in advances in Control Systems, Theory and Appl., C. T. Leondes, Ed., Vol.3, pp.219-92, Acad. Press, 1996.

[12] R. J. Meinhold and N. D. Singpurwalla, Understanding the Kalman filter,The American Statistician, Vol. 37, No. 2, pp. 123-127, May 1983.

[13] A. C. Harvey, Forecasting, structural time series models and the Kalman filter, Cambridge University Press, 1989.

[14] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings, Vol. 2, No. 140, pp. 107-113, 1993.

[15] D. T. Pham, K. Dahia, and C. Musso, A Kalman-Particle Kernel Filter and its application to terrain navigation, in Proceeding of the Fusion 2003 Conference, (Cairns, Australia), July 2003.

[16] D. T. Pham, K. Dahia, and C. Musso, A Kalman-Particle Kernel Filter for efficient nonlinear filtering, IEE Proceeding, 2004.

[17] K. Dahia, and C. Musso, D. T. Pham, and J. P. Guibert, Application of the Kalman-Particle Kernel Filter to the updated inertial navigation system, in 12 The European Signal Proceeding Conference, (Vienna, Austria), Sept. 2004.

[18] G. Favier, Filtrage, modélisation et identification des systèmes linéaires stochastiques à temps discret, Editions du CNRS, 1982.

[19] P. Kaminski, A. Bryson Jr., and S. Schmidt, Discrete square root filtering: A survey of current techniques, IEEE Transactions on Automatic Control, Vol. 16, No. 6, pp. 727-736, December 1971.

[20] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, Journal of Computational and Graphical Statistics, Vol. 5, No. 1, pp. 1-25, 1996.

[21] J. S. Liu and R. Chen, Sequential Monte Carlo methods for dynamic systems, Journal of the American Statistical Association, Vol.93, No. 443, pp. 1032-44, September 1998.

[22] A. Doucet, S. J. Godsill, and C. Andrieu, Sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and computing, Vol. 10, pp. 197-208, 2000.

[23] P. M. Djurié, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, Particle filtering, IEEE Signal Processing Magazine, Vol. 20, No. 5, pp.19-38, September 2003.

[24] M. M Dempster, N. M Laird, and D. B. Rubin, Maximum Likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B-39, pp 1-38, 1977.

[25] G. Celeux et J. Diebolt, l'algorithme SEM: un algorithme d'apprentissage probabiliste pour la reconnaissance de mélanges de densités, Revue de Statistique Appliquée, Vol. 34, No. 2, 1986.

[26] W. Pieczynski, Chaînes de Markov Triplet, Comptes Rendus de l'Académie des Sciences-Mathématique, Série I, Vol. 335, Issue 3, pp. 275-278, 2002.

[27] F. Desbouvries and W. Pieczynski, Particle Filtering in Pairwise and Triplet Markov Chains, Processing of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP 2003), Grado-Gorizia, Italy, June 8-11, 2003.

[28] N. Brunel and W. Pieczynski, Unsupervised signal restoration using Copulas and Pairwise Markov chains, IEEE Workshop on Statistical Signal Processing (SSP 2003), Saint Louis, Missouri, September 28-October 1, 2003.

[29] F. Desbouvries and W. Pieczynski. Modèle de Markov triplet et filtrage de Kalman. Comptes Rendus de l'Académie des Sciences-Mathématique, 336(8): 667-670, 2003.

[30] F. Desbouvries and W. Pieczynski. Particle filtering with pairwise Markov processes. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'03), Hong-Kong, 2003.

[31] G. D. Fornay. The Viterbi algorithm. Procceedings of the IEEE, 61(3) :268-277, 1973.

[32] M. Duflo. Algorithmes Stochastiques, Mathématiques et Applications 23, SMAI, Springer-Verlag Berlin Heidelberg 1996.

[33] F. Salzenstein and W.Pieczynski, Parameter estimation in hidden fuzzy Markov random fields and image segmentation, Graphical Models and Image Processing, Vol. 59, No. 4, pp. 205-220, 1997.

[34] F. Salzenstein and W. Pieczynski, Sur le choix de méthode de segmentation statistique d'images. Traitement du Signal, Vol. 15, No. 2, pp.119-128, 1998.

[35] C. F. J. Wu, On the convergence properties of the EM algorithm, Annals of Statistics, 11 :95-103, 1983.

[36] R. S. Bucy, Lectures on discrete time filtering, Signal Processing and Digital Filtering, Springer Verlag, 1994.

[37] P. Del Moral, G. Rigal et G. Salut, Estimation et commande optimales non linéaires. La résolution particulaire en estimation filtrage. Résultats expérimentaux. Rapport No. 2, DRET, SM.MCY/685.92/A, 18 Mars 1992.

[38] N. Bouleau, Probabilités de l'ingénieur, variables aléatoires et simulation, Hermann, 2002.

[39] J. Jacod et P. Protter, L'essentiel en théorie des probabilité,. Cassini, 2003.

[40] J. Lapuyade-Lahorgue. Sur diverses extensions des chaînes de Markov cachées avec applications au traitement des signaux radar, Thèse de l'Université Paris VI, 2009.

[41] Ben-Mabrouk-Mohamed, Modèles de Markov triplets en restauration des signaux, Thèse de l'Université Paris VI, 2011.

[42] P. Lanchantin, Chaines de Markov triplets et segmentation non supervisée de signaux. Thèse de l'Institut National des Télécommunications, 2006.

[43] B. Ait El-Fquih, Estimation Bayésienne non supervisée dans les chaînes de Markov triplets continues, Thèse de l'Institut National des Télécommunications, 2007.

[44] D. L. Alspach and H. W. Sorenson, Nonlinear Bayesian Estimation Using Gaussian Sum Approximations, IEEE Transaction on Automatic Control, AC-17(4) :439-448, August 1972.

[45] P. Lanchantin and W. Pieczynski, Unsupervised non stationary image segmentation using triplet Markov chains. In advanced Concepts for Intelligent Vision Systems (ACIVS 04), Brussels, Belgium, Aug. 31-Sept., 2004.

[46] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski, Unsupervised segmentation of triplet Markov chains with long-memory noise, Signal Processing, Vol. 5, No. 88, pp. 1134-1151, May 2008.

[47] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski, Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise, Signal Processing, Vol. 91, No. 2, pp. 163-175, 2011.

[48] J. P. Delmas, Relations entre les algorithmes d'estimation itératives EM et ICE avec exemple d'application, GRETSI, pages 185-188, 1995.

[49] B. Benmiloud and W. Pieczynski, Estimation des paramètres dans les chaines de Markov cachées et segmentation d'images, Traitement du Signal, Vol. 12, No. 5, pp. 433-454, 1995.

[50] M. Ben Mabrouk and W. Pieczynski, Unsupervised segmentation of random discrete data using triplet Markov chains, In International Symposium on Applied Stochastic Models and Data Analysis, ASMADA, Crete, Grece, May 2007.

[51] W. Pieczynski, Chaines de Markov triplets, Comptes rendus de l'Académie des Sciences - Mathématique, Vol. 335, Issue 3, pp. 275-278, 2002.

[52] W. Pieczynski. Pairwise Markov chains, IEEE Trans. on Patter Analysis and Machine Intelligence, Vol. 25, No. 5, pp. 34-639, 2003.

[53] W. Pieczynski, Triplet partially Markov chains and trees, In 2nd International Symposium on Image/Video Communication over mobile networks (ISIVC04), Brest, France, 7 -9 July 2004.

[54] W. Pieczynski, Convergence of the Iterative conditional estimation and application to mixture proportion identification, In IEEE Statistical Signal Processing Workshop, Madison, Wisconsin, USA, 26-29 August 2007.

[55] W. Pieczynski, Exact smoothing in hidden conditionally Markov switching chains, In XIII International Conference Applied Stochastic Models and Data Analysis, Vilnius Lithuania, June 30-July 3 2009.

[56] B. Ristic, S. Arulampalam, and N. Gordon, Beyond the Kalman Filter - Particle filters for tracking applications, Artech House, Boston, MA (2004).

[57] J. K. Tugnait, Adaptive estimation and identification for discrete systems with Markov jump parameters, IEEE Trans. on Automatic Control, AC-25, 1054-1065 (1982).

[58] O. Zoeter, and T. Heskes, Deterministic approximate inference techniques for conditionally Gaussian state space models, Statistical Computation, 16, pp. 279-292, 2006.

[59] P. Doukhan, G. Oppenheim, and M. S. Taqqu, Long-Range Dependence, Birkhauser, 2003.

[60] O Cappé, E. Moulines, and T. Ryden, Inference in hidden Markov models, Springer, Series in Statistics, Springer, 2005.

[61] N. Abbassi, D. Benboudjema, and W. Pieczynski, Kalman filtering approximations in triplet Markov Gaussian switching models, IEEE Workshop on Statistical Signal Processing (SSP2011), Nice, France, June 28-30, 2011

[62] N. Abbassi and W. Pieczynski, Long memory based approximation of filtering in non linear switching systems, Stochastic Modeling Techniques and Data Analysis international conference (SMTDA '10), Chania, Greece, June 8-11, 2010.

[63] N. Bardel, N. Abbassi, F. Desbouvries, W. Pieczynski and F. Barbaresco, A Bayesian filtering algorithm in Jump-Markov systems with application to Track-Before-Detect, IEEE International Radar Conference, Washington DC, USA, May 10-14, 2010.

[64] N. Abbassi et W. Pieczynski, Filtrage exact partiellement non supervisé dans les modèles cachés à sauts markoviens, GRETSI 2009, Dijon, 8-11 septembre 2009.

[65] W. Pieczynski and N. Abbassi, Exact filtering and smoothing in short or long memory stochastic switching systems, 2009 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2009), September 2-4, Grenoble, France, 2009.

[66] W. Pieczynski, N. Abbassi, and M. Ben Mabrouk, Exact Filtering and Smoothing in Markov Switching Systems Hidden with Gaussian Long Memory Noise, XIII International Conference Applied Stochastic Models and Data Analysis, (ASMDA 2009), June 30- July 3, Vilnius Lithuania, 2009.

[67] N. Abbassi and W. Pieczynski, Exact filtering in semi-Markov jumping system, Sixth International Conference of Computational Methods in Sciences and Engineering (ICCMSE 2008), September, 25-30, Hersonissos, Crete, Greece, 2008.

[68] W. Pieczynski, Statistical image segmentation, Machine Graphics and Vision, Vol.1, No 1-2, pp. 261-268, May 1992.

[69] M. Broniatowski et G. Celeux, Reconnaissance de mélanges de densités par un algorithme d'apprentissage probabiliste, Data Analysis and Informatics, North-Holland, 1983.

[70] L. E. Baum and T. Petrie, Statistical Inference for Probabilistic Functions of Finite State Markov Chains, Ann. Math. Statist. Vol. 37, No. 6, 1554-1563, 1966.

[71] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chain, Ann. of Math. Statistics, Vol. 41, pp. 164–171, 1970.

[72] P. A. Devijver. Baum's forward-backward algorithm revisited, Pattern Recognition, Vol. 3, No. 6, pp. 369–373, 1985.

[73] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society Series B, 39, pp. 1–38, 1977.

[74] G. Celeux and J. Diebolt, A stochastic approximation type em algorithm for the mixture problem. In Stochastics and Stochastics Reports, Vol. 41, pp. 119–134, 1992.

[75] W. Pieczynski, Sur la convergence de l'estimation conditionnelle itérative, *Comptes Rendu de l'Académie des Sciencess-Mathématique*, Vol. 346, No. 7-8, pp. 457-460, Avril 2008.

[76] S. J. Julier and J. K. Uhlmann, Unscented filtering and Nonlinear Estimation, Proc. IEEE, Vol. 92, pp . 401-42, March 2004.