



RAPPORT - 15 JUIN 2023

**Estimation de volatilité stochastique et de
paramètres par méthodes de Monte Carlo
Séquentielles**

Élèves :

Antonia GOBILLARD
Kevin ALINEJAD
Théophile SCHMUTZ

Enseignant :

Yohan PETETIN

Table des matières

1	Introduction	2
2	Pré-réquis	4
2.1	Notations	4
3	Méthodes de Monte Carlo et filtrage particulaire	5
3.1	Méthodes de Monte Carlo	5
3.1.1	Cadre mathématique	5
3.1.2	Différents types de problèmes pour méthodes de Monte Carlo	6
3.2	Filtrage particulaire	7
3.2.1	L'échantillonnage séquentiel pondéré	10
3.2.2	Dégénérescence des poids	12
3.2.3	<i>Sampling Importance Resampling</i>	13
3.2.4	Choix de la densité d'importance	14
3.2.5	Convergence des filtres particuliers	14
4	<i>Smoothing</i>	16
4.1	Le <i>Forward-backward</i>	16
5	<i>Parameters learning with SMC</i>	19
5.1	Filtre de Storvik	20
5.2	Le <i>Particle learning and smoothing</i>	21
6	Applications	23
6.1	Modèle linéaire	23
6.2	Modèle de Kitagawa	26
6.3	Modèle de volatilité Stochastique	28
6.3.1	Application théorique	28
6.3.2	Application d'un SIR en supposant les paramètres connus .	28
6.3.3	Application d'un filtre de Storvik	29
6.3.4	Application d'un PLS	31
6.3.5	Application d'un filtre de Storvik puis d'un SIR avec les paramètres estimés	31
6.3.6	Comparaison entre PLS, filtre de Storvik et SIR	32
6.4	Application pour l'estimation de la volatilité du S&P 500	33
	Bibliographie	35

1 Introduction

Les mathématiciens ont depuis longtemps essayé de résoudre les questions soulevées par le monde de la finance. Une des caractéristiques de ces questions est qu'elles font apparaître des dynamiques d'apparence désordonnées, il suffit de penser à la bourse pour s'en convaincre. C'est pourquoi les modèles probabilistes semblent relativement bien adaptés à cette situation. En 1901, la thèse de Louis Bachelier, *Théorie de la spéculation*, portait déjà sur ce thème. Depuis, de nombreux probabilistes se sont penchés sur ces questions raffinant sans cesse les modèles utilisés.

Ainsi, au début des années 1970, Fisher Black, Myron Scholes et Robert Merton ont opéré une avancée majeure en matière d'évaluation d'options. Ces contributions et l'ensemble des développements auxquels elles ont donné lieu sont à l'origine du célèbre modèle de Black-Scholes-Merton, plus communément appelé modèle de Black-Scholes, formule pour le prix d'une option européenne¹ d'achat. Ce modèle a eu un très grand impact sur les méthodes utilisées par les *traders* et les professionnels de la gestion des risques, tant en matière d'évaluation d'options que dans la mise au point de techniques de couverture. Ces travaux ont aussi constitué le point de départ du développement spectaculaire de l'ingénierie financière dans les années 1980 et 1990. En 1997, Robert Merton et Myron Scholes ont été récompensés par le prix Nobel d'économie pour l'importance et la qualité de leurs recherches. Le modèle de Black-Scholes repose sur l'hypothèse que la volatilité du sous-jacent (c'est-à-dire l'ampleur des variations de son prix) est constante au cours du temps. Cependant, dans la réalité, la volatilité est souvent variable et peut avoir un impact significatif sur le prix des options.

Ainsi, pour utiliser le modèle de Black-Scholes de manière efficace, il est crucial d'estimer avec précision la volatilité du sous-jacent. Cela a conduit à un grand nombre de recherches et de développements de techniques pour estimer la volatilité implicite, c'est-à-dire la volatilité que le marché attribue actuellement à l'option, à partir du prix de l'option observé sur le marché. Ainsi, il est crucial de savoir estimer la volatilité d'une option, notamment à l'aide de méthodes de lissage, afin de paramétrer et de faire "tourner" correctement le modèle.

Le modèle de Black-Scholes a été une véritable révolution dans le domaine de l'évaluation d'options, ouvrant la voie à des méthodes plus sophistiquées de couverture et de spéculation. Le modèle s'applique à un large spectre d'actifs financiers, nous pouvons notamment citer les options, qui sont souvent utilisées

1. option exerçable uniquement à l'échéance définie dans le contrat

comme instruments dérivés pour la couverture et la gestion des risques. Un actif financier est un instrument financier qui peut être acheté ou vendu sur un marché financier. Les actifs dérivés sont utilisés comme moyens de transfert de risques.

Les exemples d'actifs financiers incluent les actions, les obligations, les contrats à terme, les options, les devises, les matières premières, etc. Un actif financier est caractérisé par divers paramètres, telle que, par exemple, sa valeur marchande, sa rentabilité, sa liquidité², ou bien son risque. La volatilité est souvent utilisée comme mesure du risque d'un actif financier. La volatilité de la valeur d'un tel actif mesure, de façon schématisée, l'incertitude quant aux variations futures de son cours. Ainsi, plus la volatilité est élevée, plus la probabilité que le cours de l'actif atteigne des sommets ou subisse de fortes pertes est grande, et donc, plus l'actif financier est considéré comme risqué³. À l'inverse, un actif avec une volatilité plus faible est considéré comme plus stable et moins risqué.

On voit alors en quoi l'estimation de la volatilité porte un intérêt certain en finance pour de nombreux acteurs. Qu'il s'agisse d'entreprises, d'investisseurs, de trésoriers ou encore de gestionnaires de portefeuilles, il leur est nécessaire de pouvoir évaluer et maîtriser les risques de pertes financières pour leurs activités selon différents types d'actifs ou bien de correctement paramétrer leurs modèles.

Ainsi, nous nous intéresserons à l'implémentation d'algorithmes de Monte Carlo Séquentiels, souvent appelées méthodes à particules, pour l'estimation de paramètres dans des modèles de volatilité.

2. Un actif financier est dit liquide lorsqu'il peut être acheté ou vendu rapidement sans que cela n'ait d'impacts majeurs sur son prix

3. politique du "tout ou rien"

2 Pré-réquis

2.1 Notations

Soient $(X_n)_{n \geq 0}$ un processus caché et $(Y_n)_{n \geq 0}$ un processus observé. Pour un entier n donné, on suppose que l'on a accès à un jeu d'observations $Y_{0:n} := Y_0, \dots, Y_n$. Un estimateur sera distinguable par son chapeau, ainsi \hat{X}_n est un estimateur de X_n .

Soient $\forall i \in [1, n]$, $X^{(i)} \stackrel{i.i.d}{\sim} X$, la notation *i.i.d* correspond à *variables indépendantes et identiquement distribuées*. On note $p(x_{0:n})$ (respectivement $p(x_{0:n}|y_{0:n})$), la densité par rapport à la mesure de Lebesgue de $x_{0:n}$ (respectivement la densité conditionnelle de $x_{0:n}$ sachant $y_{0:n}$).

Les problèmes statistiques Il existe trois types de problèmes statistiques : la prédiction, le filtrage et le lissage. La *prédiction* est un problème d'inférence bayésienne qui consiste à prédire une variable future, en utilisant les connaissances que nous avons acquises sur les variables passées et présentes. La prédiction consiste à estimer la distribution de probabilité d'une variable future conditionnellement à des observations passées et présentes. Mathématiquement, cela revient à chercher h , fonction mesurable, telle que :

$$\hat{X}_{n+l} = h(Y_{0:n}) \text{ où, } l, n \in \mathbb{N}^*$$

Là où la prédiction permet d'estimer la probabilité de l'état caché futur, le filtrage permet d'estimer la probabilité de l'état caché présentement, en utilisant les observations les plus récentes jusqu'à ce moment-là. Mathématiquement, cela revient à chercher h , fonction mesurable, telle que :

$$\hat{X}_n = h(Y_{0:n}) \text{ où, } n \in \mathbb{N}^*$$

Enfin, alors que la prédiction et le filtrage sont des problèmes d'inférence bayésienne qui permettent d'estimer la distribution de probabilité d'une variable future et présente en utilisant les observations passées et présentes. Le lissage, ou *smoothing* en anglais, est un autre problème qui consiste à estimer la distribution de probabilité d'une variable passée, en utilisant toutes les observations passées et présentes. Le lissage est utilisé pour obtenir une estimation plus précise de la variable passée, en utilisant toutes les informations disponibles. Mathématiquement, cela revient à chercher h , fonction mesurable, telle que :

$$\hat{X}_k = h(Y_{0:n}) \text{ où, } k \in [0, n]$$

3 Méthodes de Monte Carlo et filtrage particulaire

L'objectif de ce chapitre est d'introduire les briques de base du filtrage particulaire et d'exposer quelques algorithmes. Nous introduisons brièvement les méthodes de Monte Carlo usuelles avant d'illustrer comment leur utilisation dans un cadre séquentiel permet d'aboutir au filtre particulaire.

3.1 Méthodes de Monte Carlo

Les méthodes de Monte Carlo sont des techniques d'analyse statistique utilisées pour résoudre des problèmes complexes dans différents domaines tels que la physique, la finance, la biologie, la chimie. Elles tirent leur nom de la ville de Monte Carlo, célèbre pour ses casinos et ses jeux de hasard, car elles reposent sur le principe de la simulation aléatoire.

En effet, les méthodes de Monte Carlo se basent sur le principe de la simulation aléatoire pour résoudre des problèmes numériques. Plus précisément, elles se basent sur l'échantillonnage stochastique, qui consiste à générer des échantillons aléatoires à partir d'une distribution de probabilité donnée, afin de simuler le comportement d'un système complexe. Les méthodes de Monte Carlo utilisent généralement des nombres aléatoires pseudo-aléatoires (ou quasi-aléatoires) pour simuler des échantillons à partir de la distribution de probabilité souhaitée. Ces nombres aléatoires sont générés à partir d'algorithmes déterministes, qui produisent une séquence de nombres qui apparaissent aléatoires, mais qui sont en fait prédéterminés par les paramètres de l'algorithme et la graine aléatoire initiale.

3.1.1 Cadre mathématique

Formellement, en notant $h : \mathcal{X} \rightarrow \mathbb{R}$ et $X \sim p$, nous souhaitons calculer :

$$\psi(h) = \int_{\mathcal{X}} h(x)p(dx) = \mathbb{E}_p[h(X)]$$

Nous pouvons dans ce cas, approcher cette espérance à l'aide de la loi forte des grands nombres. En effet, en simulant n variables aléatoires *i.i.d.*, $X^{(i)} \stackrel{i.i.d.}{\sim} X$, alors :

$$\frac{1}{n} \sum_{i=1}^n h(X^{(i)}) \xrightarrow{p.s.} \mathbb{E}_p[h(X)] = \psi(h) \quad (3.1)$$

Pour tirer les $X^{(i)}(\omega) = x^{(i)}(\omega)$ nous pouvons utiliser la méthode de la fonction de répartition inverse, trouver un changement de variable, appliquer la méthode d'acceptation rejet, ou encore appliquer une méthode MCMC. Ici, nous allons voir l'échantillonnage d'importance.

3.1.2 Différents types de problèmes pour méthodes de Monte Carlo

Principe Le problème que l'on se pose est toujours de calculer la quantité :

$$\psi(h) = \mathbb{E}_p[h(X)] = \int_{\mathcal{X}} h(x)p(dx) \quad (3.2)$$

Ainsi, en se basant sur ce résultat, nous avons trois cas possibles⁴ :

1. cas où l'intégrale (3.2) est incalculable directement, mais il est possible de faire des tirages selon $p(\cdot)$,
2. cas où l'intégrale (3.2) est incalculable directement, et il est impossible de faire des tirages selon $p(\cdot)$,
3. cas où nous ne connaissons p qu'à une constante près.

Premier cas Si l'intégrale est incalculable mais qu'il est possible de tirer selon $p(\cdot)$, on définit un estimateur de $\psi(h)$:

$$\hat{\psi}_n = \frac{1}{n} \sum_{i=1}^n h(X^{(i)}) \quad \text{avec, } X^{(i)} \sim p$$

Deuxième cas Mais il n'est pas nécessairement possible, ou facile, de faire des tirages selon p . Une alternative consiste alors à réécrire $\psi(h)$ comme une espérance selon q , appelée **loi d'importance**, une loi suivant laquelle nous savons simuler et qui a un support qui contient celui de p :

$$\psi(h) = \mathbb{E}_p[h(X)] = \int_{\mathcal{X}} h(x)p(x)dx = \int_{\mathcal{X}} h(x)\frac{p(x)}{q(x)}q(x)dx = \mathbb{E}_q\left[h(X)\frac{p(X)}{q(X)}\right] \quad (3.3)$$

Nous définissons alors un estimateur de $\psi(h)$:

$$\hat{\psi}'_n = \frac{1}{n} \sum_{i=1}^n h(X^{(i)})\frac{p(X^{(i)})}{q(X^{(i)})} = \frac{1}{n} \sum_{i=1}^n h(X^{(i)})\omega^{(i)}$$

4. en réalité il y a quatre cas possibles, mais le cas où l'intégrale est directement calculable a déjà été explicité en (3.1)

Avec $X^{(i)} \sim q$ tirages et en définissant $\omega^{(i)} = \frac{p(X^{(i)})}{q(X^{(i)})}$ les **poids d'importance**.

Nous pouvons remarquer que, par la loi forte des grands nombres, $\hat{\psi}'_n \xrightarrow{p.s.} \psi(h)$. De plus, $\hat{\psi}_n$ est un estimateur sans biais de $\psi(h)$ pour tout q , en effet :

$$\mathbb{E}[\hat{\psi}'_n] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_q \left[h(X^{(i)}) \frac{p(X^{(i)})}{q(X^{(i)})} \right] = \mathbb{E}_p[h(X)] = \psi(h)$$

Mais faut-il encore savoir calculer $p(x^{(i)})$.

Troisième cas Justement, en filtrage optimal, nous connaissons souvent p à une constante près. Soit donc $\tilde{p} \propto p$, supposée connue,

$$\psi(h) = \int h(x) \frac{\tilde{p}(x) / \int \tilde{p}(y) dy}{q(x)} q(x) dx = \frac{\int h(x) \frac{\tilde{p}(x)}{q(x)} q(x) dx}{\int \frac{\tilde{p}(y)}{q(y)} q(y) dy}$$

On définit alors un estimateur de $\psi(h)$:

$$\hat{\psi}_n'' = \frac{\frac{1}{n} \sum_{i=1}^n h(X^{(i)}) \frac{\tilde{p}(X^{(i)})}{q(X^{(i)})}}{\frac{1}{n} \sum_{i=1}^n \frac{\tilde{p}(X^{(i)})}{q(X^{(i)})}} = \sum_{i=1}^n h(X^{(i)}) \underbrace{\frac{\frac{\tilde{p}(X^{(i)})}{q(X^{(i)})}}{\sum_{j=1}^n \frac{\tilde{p}(X^{(j)})}{q(X^{(j)})}}}_{\omega^{(i)}} = \sum_{i=1}^n h(X^{(i)}) \omega^{(i)}$$

Ce qui revient à calculer l'espérance de $h(\cdot)$ non plus selon la densité $p(\cdot)$, mais selon la densité discrète $\sum_{i=1}^n \omega^{(i)} \delta_{x^{(i)}}(dx)$. A noter qu'ici les poids d'importance ont cette expression :

$$\omega^{(i)} = \frac{1}{\sum_{j=1}^n \frac{\tilde{p}(X^{(j)})}{q(X^{(j)})}} \times \frac{\tilde{p}(X^{(i)})}{q(X^{(i)})} = cste \times \frac{\tilde{p}(X^{(i)})}{q(X^{(i)})} \quad (3.4)$$

Nous pouvons désormais estimer des intégrales, suivant le cas présent, à partir de tirages.

3.2 Filtrage particulière

Les filtres particuliers, aussi connus sous le nom de méthodes de Monte-Carlo séquentielles⁵, sont des techniques d'estimation de modèles fondées sur la simulation. Nous cherchons à estimer à chaque instant une fonction d'une variable aléatoire dite cachée, X_n , à partir d'une suite de variables aléatoires appelées observations, $Y_{0:n} = (Y_0, \dots, Y_n)$, fonctions des variables cachées pour tout n . Ainsi, nous souhaitons approcher la distribution de filtrage $p(dx_{0:n}|y_{0:n})$ par un ensemble de particules pondérées $\{(\omega_n^{(j)}, x_n^{(j)})\}_{j=1}^N$.

5. souvent notées SMC

Le modèle Pour des modèles à bruit additif, on se donne une équation d'état ainsi qu'une équation d'observation :

$$\begin{cases} X_n = h(X_{n-1}) + U_n & \text{(équation d'état)} \\ Y_n = g(X_n) + V_n & \text{(équation d'observation)} \end{cases} \quad (3.5)$$

Les U_i et V_i sont de lois connues, les U_i indépendantes entre elles et indépendantes des V_j (elles aussi indépendantes entre elles). On en déduit ainsi la loi de X_i , conditionnellement à la variable X_{i-1} , ainsi que celle de Y_j , conditionnellement à X_j , ce qui caractérise donc parfaitement la loi du couple $(X_{0:n}, Y_{0:n})$ ⁶.

Nous cherchons un moyen efficace de passer d'un estimateur de $p(x_{n-1}|y_{0:n-1})$ à $p(x_n|y_{0:n})$. Nous pouvons remarquer que :

$$\begin{aligned} p(x_n|y_{0:n}) &= \int p(x_{0:n}|y_{0:n}) dx_{0:n-1} \\ &= \int \frac{p(x_{0:n}, y_{0:n})}{p(y_{0:n})} dx_{0:n-1} \\ &= \int \frac{p(x_n, y_n | x_{0:n-1}, y_{0:n-1})}{p(y_n | y_{0:n-1}) p(y_{0:n-1})} \times p(x_{0:n-1}, y_{0:n-1}) dx_{0:n-1} \\ p(x_n|y_{0:n}) &= \int \frac{p(x_n | x_{0:n-1}, y_{0:n-1}) p(y_n | x_{0:n}, y_{0:n-1})}{p(y_n | y_{0:n-1})} \times p(x_{0:n-1}, y_{0:n-1}) dx_{0:n-1} \end{aligned} \quad (3.6)$$

On observe donc que dans le cas général où, $p(x_n, y_n | x_{0:n-1}, y_{0:n-1}) \neq p(x_n, y_n | x_{n-1}, y_{0:n-1})$, l'intégrale ci-dessus ne se simplifie pas en une formule récursive sur $p(x_n|y_{0:n})$.

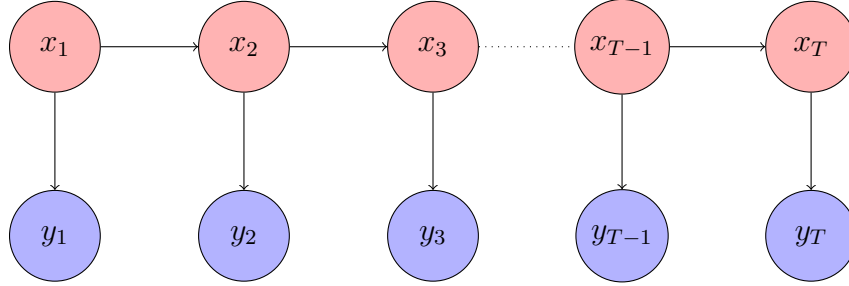
Hypothèse de chaîne de Markov cachée C'est pourquoi, nous supposons que $(X_{0:n}, Y_{0:n})$ est une chaîne de Markov cachée, les propriétés suivantes sont alors vérifiées :

$$p(x_{n+1}|x_{0:n}) = p(x_{n+1}|x_n) \quad (3.7)$$

$$p(y_{0:n}|x_{0:n}) = \prod_{i=0}^n p(y_i|x_{0:n}) \quad (3.8)$$

$$p(y_i|x_{0:n}) = p(y_i|x_i) \text{ pour tout } i \in [0, n] \quad (3.9)$$

6. Si h et g sont linéaires, nous retrouvons un modèle approprié au filtre de Kalman



A partir de ces trois propriétés, nous pouvons voir que la loi du couple $(x_{0:n}, y_{0:n})$ se factorise en :

$$\begin{aligned}
 p(x_{0:n}, y_{0:n}) &= p(x_{0:n}) \times p(y_{0:n}|x_{0:n}) \\
 &= p(x_n, x_{0:n-1}) \times \prod_{i=1}^n p(y_i|x_{0:n}) \text{ d'après l'hypothèse (3.8)} \\
 &= p(x_n|x_{0:n-1})p(x_{0:n-1}) \times \prod_{i=1}^n p(y_i|x_i) \text{ d'après l'hypothèse (3.9)} \\
 &= \prod_{i=1}^n p(x_i|x_{i-1})p(x_0) \times \prod_{i=1}^n p(y_i|x_i) \text{ d'après l'hypthèse (3.7)} \\
 p(x_{0:n}, y_{0:n}) &= \underbrace{p(x_0)p(x_1|x_0) \dots p(x_n|x_{n-1})}_{p(x_{0:n})} \times \underbrace{p(y_0|x_0) \dots p(y_n|x_n)}_{p(y_{0:n}|x_{0:n})} \quad (3.10)
 \end{aligned}$$

Nous pouvons alors simplifier l'intégrale (3.6) d'après les hypothèses (3.7), (3.8), (3.9),

$$p(x_n|y_{0:n}) = \int \frac{p(x_n|x_{0:n-1}, y_{0:n-1})p(y_n|x_{0:n}, y_{0:n-1})}{p(y_n|y_{0:n-1})} \times p(x_{0:n-1}, y_{0:n-1}) dx_{0:n-1}$$

En effet, $p(x_n|x_{0:n-1}, y_{0:n-1})$ se simplifie en,

$$\begin{aligned}
 p(x_n|x_{0:n-1}, y_{0:n-1}) &= \frac{p(x_{0:n}, y_{0:n-1})}{p(y_{x_{0:n-1}, 0:n-1})} \\
 &= \frac{p(y_{0:n-1}|x_{0:n})p(x_{0:n})}{p(x_{0:n-1}, y_{0:n-1})} \\
 &= \frac{p(y_0|x_{0:n})p(y_1|x_{0:n}) \dots p(y_{n-1}|x_{0:n}) \times p(x_n|x_{n-1}) \dots p(x_1|x_0)p(x_0)}{p(y_0|x_{0:n-1})p(y_1|x_{0:n-1}) \dots p(y_{n-1}|x_{0:n-1}) \times p(x_{n-1}|x_{n-2}) \dots p(x_1|x_0)p(x_0)} \\
 &= \frac{p(y_0|x_0)p(y_1|x_1) \dots p(y_{n-1}|x_{n-1}) \times p(x_n|x_{n-1})p(x_{n-1}|x_{n-2}) \dots p(x_1|x_0)p(x_0)}{p(y_0|x_0)p(y_1|x_1) \dots p(y_{n-1}|x_{n-2}) \times p(x_{n-1}|x_{n-2}) \dots p(x_1|x_0)p(x_0)} \\
 p(x_n|x_{0:n-1}, y_{0:n-1}) &= p(x_n|x_{n-1}) \quad (3.11)
 \end{aligned}$$

De même, d'après les hypothèses $p(y_n|x_{0:n}, y_{0:n-1})$ se simplifie en,

$$p(y_n|x_{0:n}, y_{0:n-1}) = \frac{p(x_{0:n}, y_{0:n})}{p(x_{0:n}, y_{0:n-1})} = \frac{p(y_{0:n}|x_{0:n})}{\underbrace{p(y_{0:n-1}|x_{0:n})}_{p(y_n|x_n)}} \frac{p(x_{0:n})}{p(x_{0:n})} = p(y_n|x_n) \quad (3.12)$$

L'intégrale (3.6) se simplifie alors :

$$p(x_n|y_{0:n}) = \int \frac{p(x_n|x_{n-1})p(y_n|x_n)}{p(y_n|y_{0:n-1})} p(x_{0:n-1}, y_{0:n-1}) dx_{0:n-1} \quad (3.13)$$

$$= \frac{p(y_n|x_n)}{p(y_n|y_{0:n-1})} \int p(x_n|x_{n-1})p(x_{n-1}|y_{0:n-1}) dx_{n-1} \quad (3.14)$$

qui cette fois est une expression reliant $p(x_{n-1}|y_{0:n-1})$ et $p(x_n|y_{0:n})$. Toutefois, au vu de la prochaine partie, qui consistera à appliquer des méthodes de Monte Carlo Séquentielles à notre problème, nous allons garder l'expression de $p(x_n|y_{0:n})$ sous la forme d'une intégrale $\int h(x_{0:n})p(x_{0:n-1}, y_{0:n-1})dx_{0:n-1}$, où $h(x_{0:n}) = \frac{p(x_n|x_{n-1})p(y_n|x_n)}{p(y_n|y_{0:n-1})}$.

3.2.1 L'échantillonnage séquentiel pondéré

L'algorithme d'échantillonnage séquentiel pondéré [6] (*Sequential Importance Sampling (SIS)* en anglais) est fondé sur une approximation de la densité conditionnelle $p(dx_{0:n}|y_{0:n})$. Nous allons maintenant appliquer le dernier cas vu en 3.1.2 à la propagation d'une approximation discrète de la mesure a posteriori $p(dx_{0:n}|y_{0:n})$ sous la forme d'une densité discrète $\sum_{i=1}^n \omega^{(i)} \delta_{x^{(i)}}(dx)$ en simulant un échantillon issu d'une loi de densité q , dite loi d'importance ou loi instrumentale, cela nous permettra d'approcher une intégrale de la forme $\int h(x_{0:n})p(x_{0:n}|y_{0:n})dx_{0:n}$ en calculant l'espérance de $h(\cdot)$ non plus selon la densité $p(\cdot)$ mais selon la densité discrète. Pour ce faire, il suffira alors de marginaliser pour obtenir un algorithme de propagation d'une approximation de la mesure de filtrage $p(dx_n|y_{0:n})$. Soit donc,

$$\hat{p}(dx_{0:n-1}|y_{0:n-1}) = \sum_{i=1}^N \omega_{n-1}^{(i)} \delta_{x_{0:n-1}^{(i)}}(dx_{0:n-1}) \quad (3.15)$$

Avec $\omega_{n-1}^{(i)} \propto \frac{p(x_{0:n-1}^{(i)}|y_{0:n-1})}{q(x_{0:n-1}^{(i)}|y_{0:n-1})}$ et $x_{0:n-1}^{(i)} \sim q(x_{0:n-1}|y_{0:n-1})$ pour une certaine densité d'importance q . De manière générale, échantillonner $q(x_{0:n}|y_{0:n})$ a une complexité qui augmente au moins linéairement au cours du temps car la dimension du vecteur $x_{0:n}$ est linéaire à n . Il est donc intéressant d'avoir une méthode d'échantillonnage séquentielle. On souhaite passer de $\hat{p}(dx_{0:n-1}|y_{0:n-1})$ à $\hat{p}(dx_{0:n}|y_{0:n})$, c'est à dire de $\{(\omega_{n-1}^{(j)}, x_{n-1}^{(j)})\}_{j=1}^N$ à $\{(\omega_n^{(j)}, x_n^{(j)})\}_{j=1}^N$.

- Notons qu'en filtrage, le but est d'estimer la loi a posteriori $p(x_n^{(i)}|y_{0:n})$. On a donc uniquement besoin de l'ensemble des particules à l'instant précédent $\{x_{n-1}^{(i)}\}_{i=1}^N$. L'algorithme d'échantillonnage pondéré séquentiel peut alors être simplifié en choisissant une densité d'importance q telle que. Ainsi, considérons l'actualisation des points $x_{0:n-1}^{(i)}$, dans un souci d'économie, nous prenons soin de choisir une densité d'importance qui se factorise en :

$$q(x_{0:n}|y_{0:n}) = q(x_{0:n-1}|y_{0:n-1})q(x_n|x_{0:n-1}, y_{0:n}) \quad (3.16)$$

Il suffira de tirer N particules $x_n^{(i)}$ selon la densité d'importance $q(x_n|x_{0:n-1}^{(i)}, y_{0:n})$.

- Considérons maintenant l'actualisation des poids d'importance, du fait de la factorisation (3.16), nous avons :

$$\frac{p(x_{0:n}^{(i)}|y_{0:n})}{q(x_{0:n}^{(i)}|y_{0:n})} = \frac{p(x_n^{(i)}|x_{n-1}^{(i)})p(y_n|x_n^{(i)})}{p(y_n|y_{0:n-1})q(x_n^{(i)}|x_{0:n-1}^{(i)}, y_{0:n})} \times \frac{p(x_{0:n-1}^{(i)}|y_{0:n-1})}{q(x_{0:n-1}^{(i)}|y_{0:n-1})} \quad (3.17)$$

et donc $\omega_n^{(i)} \propto \omega_{n-1}^{(i)} \frac{p(x_n^{(i)}|x_{n-1}^{(i)})p(y_n|x_n^{(i)})}{q(x_n^{(i)}|x_{0:n-1}^{(i)}, y_{0:n})}$ avec $\sum_{i=1}^N \omega_n^{(i)} = 1$

Ainsi, en utilisant le caractère récursif de la loi d'importance :

$$\omega_n^{(i)} \propto \omega_{n-1}^{(i)} \frac{p(x_n^{(i)}|x_{n-1}^{(i)})p(y_n|x_n^{(i)})}{q(x_n^{(i)}|x_{n-1}^{(i)}, y_n)}$$

La méthode Il reste à déduire des équations précédentes un algorithme de calcul récursif de $\hat{p}(dx_n|y_{0:n})$. A partir de (3.15), nous obtenons :

$$\hat{p}(dx_n|y_{0:n}) = \sum_{i=1}^N \omega_n^{(i)} \delta_{x_n^{(i)}}(dx_n)$$

Il faut donc tirer N particules $x_n^{(1)}, \dots, x_n^{(N)}$ selon la densité d'importance conditionnelle $q(x_n|x_{0:n-1}^{(i)}, y_{0:n})$, et réactualiser les poids comme ci-dessus. Ainsi, le filtre particulière peut s'écrire de la façon suivante :

- à l'instant $n - 1$, soit $\hat{p}(dx_{n-1}|x_{0:n-1}^{(i)}, y_{0:n-1}) = \sum_{i=1}^N \omega_{n-1}^{(i)} \delta_{x_{n-1}^{(i)}}(dx_{n-1})$ une approximation discrète de la mesure de filtrage $p(dx_{n-1}|x_{0:n-1}^{(i)}, y_{0:n-1})$;
- On tire N particules $\{x_n^{(i)}\}_{i=1}^N$ selon la densité d'importance conditionnelle $q(x_n|x_{0:n-1}^{(i)}, y_{0:n})$, on a alors $x_{0:n}^{(i)} = \{x_{0:n-1}^{(i)}, x_n^{(i)}\}$;
- on actualise les poids d'importance selon $\omega_n^{(i)} \propto \omega_{n-1}^{(i)} \frac{p(x_n^{(i)}|x_{n-1}^{(i)})p(y_n|x_n^{(i)})}{q(x_n^{(i)}|x_{n-1}^{(i)}, y_n)}$, avec $\sum_{i=1}^N \omega_n^{(i)} = 1$;

- finalement, $\hat{p}(dx_n|y_{0:n}) = \sum_{i=1}^N \omega_n^{(i)} \delta_{x_n^{(i)}}(dx_n)$ est une approximation discrète de $p(dx_n|y_{0:n})$.

L'algorithme SIS se résume ainsi,

Algorithm 1: Sequential Importance Sampling

Data: x_0

Result: $(\omega_t^{(i)})_{(i,t) \in [1:N] \times [1:T]}, (x_t^{(i)})_{(i,t) \in [1:N] \times [1:T]}$

for $i = 1, \dots, N$ **do**

 tirer $x_0^{(i)} \sim q(x_0|y_0)$;

 calculer $\omega_0^{(i)} \propto x_0 \frac{p(y_0|x_0^{(i)})}{q(x_0^{(i)}|y_0)}$;

end

normaliser les $(\omega_0^{(i)})_{i=1}^N$

for $t = 1, \dots, T$ **do**

for $i = 1, \dots, N$ **do**

 (Prédiction) tirer $x_t^{(i)} \sim q(x_t^{(i)}|x_{t-1}^{(i)}, y_t)$;

 (Correction) calculer $\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(x_t^{(i)}|x_{t-1}^{(i)})p(y_t|x_t^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)}, y_t)}$;

end

 normaliser les $(\omega_t^{(i)})_{i=1}^N$.

end

Les estimées de la moyenne et de la covariance de l'état x_t sont alors respectivement :

$$\mathbb{E}[x_t|y_{0:t}] \approx \sum_{i=1}^N \omega_t^{(i)} x_t^{(i)} = \hat{x}_t \quad (3.18)$$

$$\mathbb{E}[(x_t - \hat{x}_t)^2|y_{0:t}] \approx \sum_{i=1}^N \omega_t^{(i)} (x_t^{(i)} - \hat{x}_t)^2 \quad (3.19)$$

3.2.2 Dégénérescence des poids

L'algorithme d'échantillonnage séquentiel pondéré est la brique de base du filtrage particulière. Cependant, il souffre d'une limitation importante : après quelques itérations, toutes les particules sauf une ont un poids proche de zéro. L'estimation de la loi a posteriori est donc réduite à un dirac, ce qui entraîne la divergence du filtre. Ce phénomène est caractérisé par une augmentation dans le temps de la variance des poids normalisés $\{\omega_t^{(i)}\}_{i=1}^N$. [1] ont notamment montré que lorsque la loi d'importance de densité $q(x_{0:k}|y_{0:k})$ est différente de la loi cible de densité $p(x_{0:k}|y_{0:k})$, la variance des poids ne peut qu'augmenter.

3.2.3 Sampling Importance Resampling

Le ré-échantillonnage a pour objectif de dupliquer les particules dont le poids est suffisamment fort et d'éliminer les particules dont le poids est trop faible afin de ne conserver que les particules les plus significatives. L'ajout de l'étape de ré-échantillonnage dans l'algorithme SIS permet d'obtenir l'algorithme SIR [10] qui évite le phénomène de dégénérescence des poids et permet d'obtenir un filtre stable. Lorsque le ré-échantillonnage intervient à chaque pas de temps, le SIR correspond au bootstrap filter de Gordon, Salmond et Smith [9].

Le rééchantillonnage L'algorithme de rééchantillonnage, ou *sampling* en anglais, consiste à éliminer les particules $x_{0:t}^{(i)}$ de poids faibles et dupliquer les particules de poids forts, nous tirons la trajectoire i avec une probabilité égale à son poids.

Algorithm 2: Sequential Importance Resampling

Data: x_0
Result: $(\omega_t^{(i)})_{(i,t) \in [1:N] \times [1:T]}, (x_t^{(i)})_{(i,t) \in [1:N] \times [1:T]}$
for $i = 1, \dots, N$ **do**
 tirer $x_0^{(i)} \sim q(x_0|y_0)$;
 calculer $\omega_0^{(i)} \propto x_0 \frac{p(y_0|x_0^{(i)})}{q(x_0^{(i)}|y_0)}$;
 normaliser les $(\omega_0^{(i)})_{i=1}^N$.
end
for $t = 1, \dots, T$ **do**
 for $i = 1, \dots, N$ **do**
 (Prédiction) tirer $x_t^{(i)} \sim q(x_t^{(i)}|x_{t-1}^{(i)}, y_t)$;
 (Correction) calculer $\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(x_t^{(i)}|x_{t-1}^{(i)})p(y_t|x_t^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)}, y_t)}$;
 end
 normaliser les $(\omega_t^{(i)})_{i=1}^N$.
 (Ré-échantillonnage) ré-échantillonner les particules ;
end

Bien que le ré-échantillonnage permette de garantir la survie du système de particules, il introduit plusieurs effets indésirables. La duplication des particules de poids fort a tendance à appauvrir le système de particules, ce qui peut être problématique lorsque le bruit de dynamique est faible. Dans ce cas, lorsque la loi d'importance utilisée dans l'étape de prédiction est le noyau de transition $p(x_n|x_{n-1})$, les particules ont tendance à se concentrer autour de quelques zones resserrées provoquant une perte de diversité dans le système de particules (phénomène connu

sous le nom de *sample impoverishment* dans la littérature). Cela augmente le risque qu'aucune des particules ne soit cohérente avec l'observation y_n courante, ce qui, à terme, peut provoquer la divergence du filtre.

3.2.4 Choix de la densité d'importance

Le choix d'une densité d'importance appropriée permet de concevoir des filtres plus robustes. Dans la classe des algorithmes d'échantillonnage pondéré séquentiel, la densité d'importance $q_{opt} = p(x_n|x_{n-1}, y_n) = \frac{p(y_n|x_n)p(x_n|x_{n-1})}{p(y_n|x_{n-1})}$ est optimale. On peut montrer que la variance des poids normalisés correspondant $\omega_n^{(i)}$ est nulle, conditionnellement à $\omega_{n-1}^{(i)}$.

3.2.5 Convergence des filtres particuliers

La littérature sur les méthodes Monte Carlo séquentielles est riche en résultats sur la convergence des filtres particuliers lorsque $N \rightarrow +\infty$. On peut citer notamment [5],[3], [4] ou [8]. On rappelle ci-dessous un résultat établi pour l'algorithme SIR.

Convergence faible [4] Soit ϕ une fonction continue bornée de \mathbb{R}^d . On suppose que la fonction de vraisemblance $x_n \mapsto p(y_n|x_n)$ est bornée. Alors,

$$\lim_{N \rightarrow \infty} \sum_{i=1}^N \omega_n^{(i)} \phi(x_n^{(i)}) = \int_{\mathbb{R}^d} \phi(x_n) p(x_n|y_{0:n}) dx_n \text{ p.s.}$$

Un autre résultat concernant la convergence en norme \mathbb{L}^p s'énonce comme suit :

Convergence \mathbb{L}^p [8] Sous les hypothèses précédentes, pour tout $p \geq 1$, il existe un réel $C_{n,p} > 0$, indépendant de N , tel que pour toute fonction ϕ continue bornée de \mathbb{R}^d ,

$$\mathbb{E} \left[\left| \langle \phi, p(\cdot|y_{0:n}) \rangle - \langle \phi, \hat{p}(\cdot|y_{0:n}) \rangle \right|^p \right]^{\frac{1}{p}} \leq C_{n,p} \frac{\sup_{x \in \mathbb{R}^d} |\phi(x)|}{\sqrt{N}}$$

Où, $\langle \phi, p(\cdot|y_{0:n}) \rangle = \int_{\mathbb{R}^d} \phi(x_n) p(x_n|y_{0:n}) dx_n$ et $\hat{p}(\cdot|y_{0:n}) = \sum_{i=1}^N \omega_n^{(i)} \delta_{x_n^{(i)}}$.

Ce dernier résultat montre que l'erreur \mathbb{L}^p converge à la vitesse $1/\sqrt{N}$. Mais comme la constante $C_{n,p}$ dépend du temps (n), sans hypothèse supplémentaire, $C_{n,p}$ augmente généralement de manière exponentielle avec le temps. Ceci suggère que pour maintenir une erreur bornée, on a en général besoin d'un nombre croissant de particules.

Conclusion de la section 3

Nous avons présenté les briques de base du filtre particulière dans le cadre des modèles à espace d'état. L'algorithme SIS assure théoriquement une convergence vers la loi de filtrage lorsque le nombre de particules est infiniment grand. Cependant, l'utilisation d'un nombre fini de particules peut occasionner la divergence des algorithmes particuliers. En particulier, lorsque le bruit d'observation est faible ou lorsque la dynamique est faiblement bruitée, l'échantillonnage successif des particules selon la loi à priori ne garantit pas la convergence du filtre. Le chapitre suivant propose d'introduire des méthodes de lissage, aussi appelées *smoother* en anglais.

4 *Smoothing*

On s'intéresse désormais à la résolution d'un problème de lissage. Pour rappel, cela consiste à chercher à estimer la distribution de probabilité d'une variable passée, en utilisant toutes les observations passées et présentes. En particulier, on cherche, à partir de la connaissance de $N + 1$ observations $y_{0:N}$ du processus observé, à obtenir la densité à posteriori $p(x_n|y_{0:N})$, où $n \in [[0; N]]$. Le problème est alors le suivant : on souhaite à tout instant n estimer la valeur prise par x_n à partir des observations passées et futures $y_{0:N}$ recueillies depuis l'origine jusqu'à un temps $N \geq n$. On verra que, comme dans le cas du filtrage particulaire, on se ramènera à un calcul récursif. A noter qu'il nous faudra initialiser le lissage avec $p(x_0, y_0)$.

Hypothèses et notations Afin de faciliter les calculs et la compréhension, on pose pour la suite $z_{0:n} := (x_{0:n}, y_{0:n})$. De plus, nous supposons ici aussi que $(x_{0:n}, y_{0:n})$ est une chaîne de Markov cachée. Les propriétés (3.7), (3.8) et (3.9) explicitées s'appliquent donc à $z = \{z_n\}$, le processus est doté d'une structure de chaîne de Markov.

4.1 Le *Forward-backward*

Ainsi, en supposant que z est une chaîne de Markov, on vérifie alors :

$$p(z_n|z_{0:n-1}) = p(z_n|z_{n-1})$$

Par conséquent, le passé et le futur de z sont donc, conditionnellement au présent z_n , indépendants, c'est-à-dire que l'on a :

$$p(z_{0:n-1}, z_{n+1:N}|z_n) = p(z_{0:n-1}|z_n)p(z_{n+1:N}|z_n) \quad (4.1)$$

On en déduit une expression de la densité à posteriori :

$$\begin{aligned} p(x_n|y_{0:N}) &= \frac{p(y_{0:n-1}, z_n, y_{n+1:N})}{\int p(y_{0:n-1}, z_n, y_{n+1:N}) dx_n} \\ &\stackrel{(4.1)}{=} \frac{p(y_{0:n-1}|z_n)p(y_{n+1:N}|z_n)p(z_n)}{\int p(y_{0:n-1}|z_n)p(y_{n+1:N}|z_n)p(z_n) dx_n} \\ p(x_n|y_{0:N}) &= \frac{p(y_{0:n}, x_n)p(y_{n+1:N}|x_n, y_n)}{\int p(y_{0:n}, x_n)p(y_{n+1:N}|x_n, y_n) dx_n} \end{aligned}$$

Ainsi, avec les équations de chaîne de Markov $p(y_{n+1:N}|x_n, y_n) = p(y_{n+1:N}|x_n)$:

$$p(x_n|y_{0:N}) = \frac{p(x_n, y_{0:n})p(y_{n+1:N}|x_n)}{\int p(x_n, y_{0:n})p(y_{n+1:N}|x_n)dx_n}$$

En posant

$$\begin{cases} \alpha(x_n) &:= p(x_n, y_{0:n}) \\ \beta(x_n) &:= p(y_{n+1:N}|x_n) \end{cases} \quad (4.2)$$

$$(4.3)$$

Il vient finalement :

$$p(x_n|y_{0:N}) = \frac{\alpha(x_n)\beta(x_n)}{\int \alpha(x_n)\beta(x_n)dx_n} \quad (4.4)$$

On peut donc calculer la probabilité conditionnelle de x_n sachant toutes les observations $y_{0:N}$ si l'on connaît $\alpha(x_n)$ et $\beta(x_n)$. Or il se trouve que $\alpha(x_n)$ et $\beta(x_n)$ peuvent se réécrire récursivement :

$$\begin{aligned} \alpha(x_{n+1}) &= p(x_{n+1}, y_{0:n+1}) \\ &= \int p(x_n, x_{n+1}, y_{0:n+1})dx_n \\ &= p(y_{n+1}|x_{n+1}) \int p(x_{n+1}|x_n)p(x_n, y_{0:n})dx_n \text{ d'après les hypothèses (3.8) et (3.9)} \\ \alpha(x_{n+1}) &= p(y_{n+1}|x_{n+1}) \int p(x_{n+1}|x_n)\alpha(x_n)dx_n \end{aligned} \quad (4.5)$$

De même,

$$\begin{aligned} \beta(x_n) &= p(y_{n+1:N}|x_n) \\ &= \frac{p(x_n, y_{n+1:N})}{p(x_n)} \\ &= \frac{\int p(x_n, x_{n+1}, y_{n+1:N})dx_{n+1}}{p(x_n)} \\ &= \frac{\int p(x_n, y_{n+2:N}, z_{n+1})dx_{n+1}}{p(x_n)} \text{ d'après} \\ &= \int p(y_{n+2:N}|x_{n+1}, y_{n+1})p(y_{n+1}|x_{n+1}, x_n)p(x_{n+1}|x_n)dx_{n+1} \\ \beta(x_n) &= \int \beta(x_{n+1})p(y_{n+1}|x_{n+1})p(x_{n+1}|x_n)dx_{n+1} \end{aligned} \quad (4.6)$$

Ainsi :

$$\alpha(x_{n+1}) = p(y_{n+1}|x_{n+1}) \int p(x_{n+1}|x_n) \alpha(x_n) dx_n \quad (4.7)$$

$$\beta(x_n) = \int \beta(x_{n+1}) p(y_{n+1}|x_{n+1}) p(x_{n+1}|x_n) dx_n \quad (4.8)$$

Dans ces deux expressions, les densités $p(x_{n+1}|x_n)$ et $p(y_{n+1}|x_{n+1})$ respectivement de transition et de vraisemblance sont connues. Par ailleurs $\alpha(x_0) = p(x_0, y_0)$ et $\beta(x_N) = 1$. Récursivement, il est ainsi possible de calculer les expressions de $\alpha(x_n)$ et $\beta(x_n)$ et donc de la densité cherchée $p(x_n|y_{0:N})$, pour $n \in [0; N]$.

Remarques Pour calculer $p(x_n|y_{0:N})$ via (4.4), il suffit donc de calculer $\alpha(x_n)$ dans le sens direct,

$$\alpha(x_0) = p(x_0, y_0) \longrightarrow \alpha(x_1) \longrightarrow \cdots \longrightarrow \alpha(x_{n-1}) \longrightarrow \alpha(x_n)$$

et $\beta(x_n)$ dans le sens rétrograde,

$$\beta(x_N) = 1 \longrightarrow \beta(x_{N-1}) \longrightarrow \cdots \longrightarrow \beta(x_{n+1}) \longrightarrow \beta(x_n)$$

d'où le nom d'algorithme *forward-backward* ;

5 *Parameters learning with SMC*

Dans le domaine de l'estimation de la volatilité stochastique et des paramètres, les méthodes de Monte Carlo séquentielles (SMC) sont devenues une approche puissante et populaire. Parmi ces méthodes, le filtre de Storvik [11] joue un rôle crucial dans l'apprentissage des paramètres. Cette section se concentre sur l'importance et la pertinence du filtre de Storvik dans le processus d'estimation des paramètres par SMC.

Le filtre de Storvik - également connu sous le nom de filtre bootstrap de Storvik - est un outil d'estimation qui permet de mettre à jour et d'ajuster les paramètres d'un modèle stochastique en utilisant des techniques de simulation Monte Carlo. En exploitant la puissance des échantillons aléatoires, le filtre de Storvik offre une approche flexible et efficace pour l'apprentissage des paramètres.

L'une des raisons pour lesquelles le filtre de Storvik est pertinent dans le contexte de l'estimation de la volatilité stochastique et des paramètres est sa capacité à traiter des modèles complexes. Les modèles financiers, par exemple, peuvent présenter des dynamiques non linéaires et des dépendances complexes entre les variables. Le filtre de Storvik permet d'estimer les paramètres de tels modèles de manière robuste et précise, en prenant en compte les incertitudes et les non-linéarités inhérentes.

De plus, le filtre de Storvik offre une approche séquentielle pour l'apprentissage des paramètres, ce qui signifie qu'il peut mettre à jour les estimations des paramètres au fur et à mesure que de nouvelles observations deviennent disponibles. Cette caractéristique est extrêmement précieuse dans les domaines où les données sont acquises progressivement dans le temps, par exemple dans les séries chronologiques financières. Le filtre de Storvik permet de maintenir les estimations à jour et d'adapter les paramètres du modèle en fonction de l'évolution des données.

Enfin, le filtre de Storvik a l'avantage de fournir des estimations non biaisées des paramètres, même lorsque le modèle est mal spécifié ou lorsque des erreurs de modélisation sont présentes. Cette propriété est cruciale dans la pratique, car les modèles financiers sont souvent simplifiés et approximatifs. Le filtre de Storvik permet donc de surmonter certaines des limitations associées à l'estimation des paramètres dans des environnements réels, offrant ainsi une méthode plus fiable et réaliste pour l'apprentissage des paramètres.

En résumé, le filtre de Storvik est un outil pertinent et puissant dans le contexte de l'estimation de la volatilité stochastique et des paramètres. En combinant les

techniques de simulation de Monte Carlo avec une approche séquentielle, ce filtre permet d'estimer les paramètres de modèles complexes, d'ajuster les estimations au fur et à mesure que l'on acquiert la connaissance de nouvelles observations et de fournir des résultats non biaisés, même dans des situations où les modèles sont approximatifs ou mal spécifiés. Cette section explore plus en détail le fonctionnement et les avantages du filtre de Storvik dans le processus d'apprentissage des paramètres par méthodes de Monte Carlo séquentielles.

5.1 Filtre de Storvik

Cette section présente une approche du filtrage de particules en présence de paramètres inconnus. L'approche habituelle consiste à inclure les paramètres dans le vecteur d'état (x_n, θ) . En raison de la caractéristique non dynamique des paramètres, les échantillons de θ au temps t_n ne prennent que les valeurs données au moment t_{n-1} . Étant donné que certaines de ces valeurs deviennent très improbables lorsque de nouvelles observations arrivent, il en résultera un appauvrissement de l'ensemble des valeurs distinctes de θ . L'approche donnée en [11] est basée sur une approche différente. Supposons que la distribution a posteriori de θ sachant $y_{1:n}$ et $x_{1:n}$ dépend d'une statistique suffisante $S_n = S_n(x_{1:n}, y_{1:n})$, où S_n est facile à mettre à jour de manière récursive.

Supposons qu'un ensemble approximatif de particules S_{n-1} est disponible à partir de la distribution postérieure $p(x_{1:n-1}, y_{1:n-1})$. Là encore, l'ensemble de particules S_{n-1} doit être actualisé en un nouvel ensemble de particules S_n au temps t_n . Même si seul le processus $\{x_n\}$ ainsi que les statistiques suffisantes $\{S_n\}$ seront stockés, la simulation est simplifiée si θ est inclus en tant que variable auxiliaire dans l'étape de simulation. L'approche est basée sur ce qui suit :

$$\begin{aligned}
 p(x_{1:n}, \theta | y_{1:n}) &= \frac{p(x_{1:n}, \theta, y_n | y_{1:n-1})}{p(y_n | y_{1:n-1})}, \text{ on notera } C_n = p(y_n | y_{1:n-1})^{-1} \\
 &= C_n p(x_{1:n-1} | y_{1:n-1}) p(\theta | x_{1:n-1}, y_{1:n-1}) p(x_n | x_{1:n-1}, y_{1:n-1}, \theta) \\
 &\quad \times p(y_n | x_{1:n}, y_{1:n-1}, \theta) \\
 &= C_n p(x_{1:n-1} | y_{1:n-1}) p(\theta | S_{n-1}) p(x_n | x_{n-1}, \theta) p(y_n | x_n, \theta) \quad (5.1)
 \end{aligned}$$

A noter que $C_n = p(y_n | y_{1:n-1})^{-1}$ est une constante ne dépend pas des $x_{1:n}$ ou de θ . Ainsi, nous pouvons reprendre les méthodes de filtrage particulaire vues précédemment, 2,1, en ajoutant une étape supplémentaire pour simuler θ . Selon Geir Storvik, l'approche la plus simple serait de simuler $x_{1:n-1}$ avec $\hat{p}(x_{1:n-1} | y_{1:n-1})$, θ de $p(\theta | S_{n-1})$, x_n de $p(x_n | x_{1:n-1}, \theta)$. L'aspect important de cette approche est que le paramètre θ simulé au temps t_n ne dépend pas des valeurs simulées aux points

de temps précédents. Cela permet d'éviter le problème de l'appauvrissement. En principe, l'existence d'une statistique suffisante de faible dimension pour θ n'est pas nécessaire car seule l'évaluation ou la simulation de $p(\theta|x_{1:n}, y_{1:n})$ est nécessaire, comme l'ont noté Liu et Chen [7]. Cependant, pour que le filtre fonctionne rapidement et ne soit pas de plus en plus complexe au fil du temps, il est nécessaire que $p(\theta|x_{1:n}, y_{1:n})$ ne dépende que de $(x_{1:n}, y_{1:n})$ tout au long de S_n .

Algorithm 3: Storvik's filter

Data: x_0
Result: $(\omega_t^{(i)})_{(i,t) \in [1:N] \times [1:T]}, (x_t^{(i)})_{(i,t) \in [1:N] \times [1:T]}, \theta$
for $i = 1, \dots, N$ **do**
 tirer $x_0^{(i)} \sim q(x_0|y_0)$;
 calculer $\omega_0^{(i)} \propto x_0^{(i)} \frac{p(y_0|x_0^{(i)})}{q(x_0^{(i)}|y_0)}$;
 normaliser $\omega_0^{(i)}$
end
for $t = 1, \dots, T$ **do**
 for $i = 1, \dots, N$ **do**
 tirer $\theta \sim f_{t,1}(\theta|x_{0:t-1}^{(i)}, y_{0:t-1})$;
 tirer $x_t^{(i)} \sim f_{t,2}(x_t|x_{0:t-1}^{(i)}, y_t, \theta)$ calculer
 $\omega_t^{(i)} = \omega_{t-1}^{(i)} \times \frac{p(\theta|S_{t-1}^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)}, \theta)p(y_t|x_t^{(i)}, \theta)}{f_{t,1}(\theta|x_{0:t-1}^{(i)}, y_{1:t})f_{t,2}(x_t^{(i)}|x_{0:t-1}^{(i)}, y_t, \theta)}$
 end
 for $i = 1, \dots, N$ **do**
 tirer un indice j de $[1 : N]$ avec une probabilité proportionnelle à
 $\omega_t^{(j)}$
 $x_{1:t}^{(i)} \leftarrow x_{1:t}^{(j)}$
 $S_t^{(i)} \leftarrow T(S_t^{(j)}, x_t^{(i)}, y_t)$;
 end
end

Ici, $f_{t,1}(\theta|x_{0:t-1}^{(i)}, y_{1:t})$ et $f_{t,2}(x_t^{(i)}|x_{0:t-1}^{(i)}, y_t, \theta)$ sont des densités d'importances respectivement de θ et x_t . Typiquement, $f_{t,1}(\theta|x_{0:t-1}^{(i)}, y_{1:t}) = f_{t,1}(\theta|S_{t-1}^{(i)}|y_t)$ afin que la simulation et les calculs soient computationnellement rapides. $T(\cdot)$ est une fonction qui mets à jour la statistique suffisante.

5.2 Le Particle learning and smoothing

Dans le lissage particulière avec des paramètres inconnus, nous souhaitons estimer les états et les paramètres conditionnellement à l'ensemble des données $y_{0:T}$

et tirer des échantillons $(x_{0:T}^{(i)}, \theta^{(i)})$ de la postériorité conjointe $p(x_{0:T}, \theta | y_{0:T})$, où T représente le nombre d'étapes temporelles.

Carvalho et al. (2010) [2] ont montré qu'une *backward pass* peut être exécutée après l'algorithme de filtrage et *particle learning*, et que les particules filtrées peuvent être ré-échantillonnées pour obtenir des tirages de la distribution de lissage. L'idée est basée sur la règle de Bayes et la décomposition de la distribution de lissage postérieure conjointe comme suit,

$$p(x_{0:T}, \theta | y_{0:T}) = p(x_T, \theta | y_{0:T}) \prod_{t=1}^{T-1} p(x_t | x_{t+1}, \theta, y_{0:t})$$

Où, $p(x_t | x_{t+1}, \theta, y_{0:t}) \propto p(x_{t+1} | x_t, \theta) p(x_t | \theta, y_{0:t})$

Algorithm 4: Particle Learning and Smoothing (PLS)

(Forward Filter) Exécuter l'algorithme de *particle learning* pour générer des échantillons $(x_t^{(i)}, \theta^{(i)})_{i=1}^N$ à partir de $p(x_t, \theta | y_{0:t})$ à chaque fois $t = 1, \dots, N$.

(Backward smoother) Sélectionner un couple $(x_T^{(i)}, \theta^{(i)})_{i=1}^N$ de la première étape, et simuler en arrière :

for $t = T - 1, \dots, 1$ **do**

 Ré-échantillonner les particules $(x_t^{(j)})_{j=1}^N$ de la première étape avec les poids proportionnels à $\omega_t^{(j)} \propto p(x_{t+1}^{(j)} | x_t^{(j)}, \theta^{(i)})$ pour générer $x_t^{(i)}$

end

6 Applications

Ainsi, nous avons vu comme algorithmes :

- le *Sequential Importance Sampling* (SIS) (1),
- le *Sequential Importance Resampling* (SIR) (2),
- le *Storvik's filter* (3),
- le *Particle Learning and Smoothing* (PLS) (4).

Généralement, pour chaque modèle, nous allons appliquer le SIR en supposant les paramètres du modèle connus. Puis, dans un soucis de réalisme, nous appliquerons le filtre de Storvik afin d'estimer les paramètres du modèle ainsi que l'état caché, cependant, il s'avère que le Storvik donne des résultats moyens concernant l'estimation des états cachés, c'est pourquoi, nous appliquerons le *Backward smoother* (l'application consécutive du filtre de Storvik et du *backward Smoother* formeront le PLS), ou un SIR classique avec les paramètres du modèle estimés.

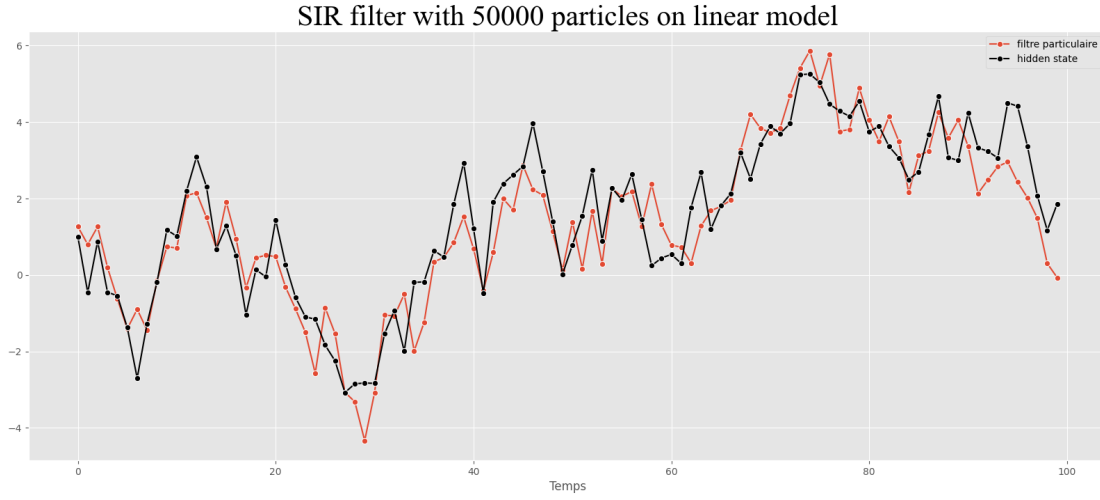
A noter que le papier de S. Godsill, A. Doucet et C. Andrieu [1] souligne le fait que le PLS n'est pas très optimal, cet article et apporte quelque amélioration appelée PLSa.

6.1 Modèle linéaire

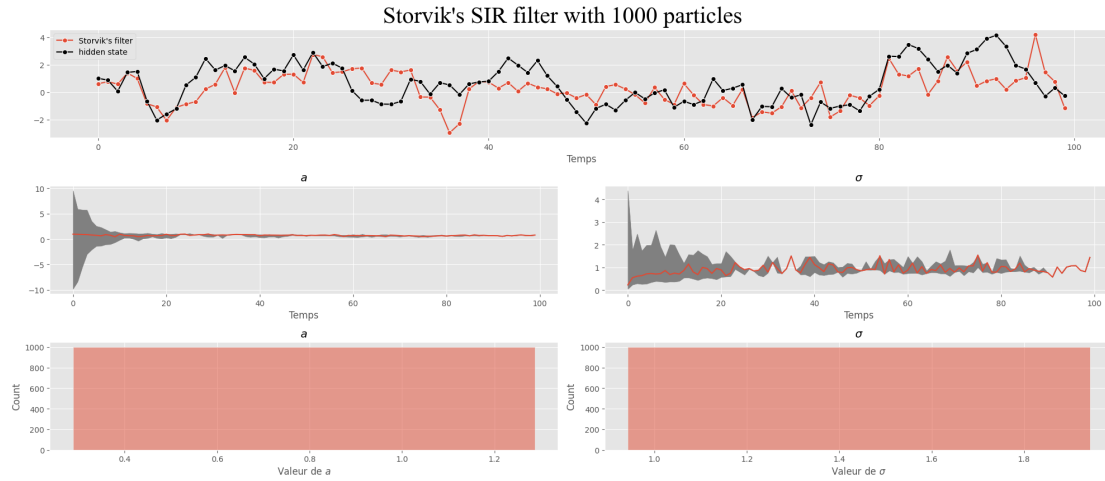
Les articles de Doucet [1] et Storvik [11] introduisent le modèle linéaire comme étant :

$$\begin{cases} x_t &= ax_{t-1} + \sigma\epsilon_t, \text{ avec } \epsilon_t \sim \mathcal{N}(0, 1), \\ y_t &= x_t + \nu_t, \text{ avec } \nu_t \sim \mathcal{N}(0, 1) \end{cases}$$

Dans la suite, on fixera $a = 0.9$ et $\sigma = 1$.

FIGURE 1 – Application du SIR sur le modèle linéaire avec $N = 50000$ particules

En supposant désormais, comme c'est le cas en réalité, que les paramètres sont inconnus ($\theta = (a \ \sigma^2)^T$ inconnu et à estimer), on obtient grâce à l'algorithme de Storvik :

FIGURE 2 – Application du filtre de Storvik sur le modèle linéaire avec $N = 1000$ particules

D'après le papier de Storvik [11], le procédé pour mettre à jour σ^2 et a est :

$$a \sim \mathcal{N}(m_k, \sigma^2 C_k) \quad \sigma^2 \sim \mathcal{IG}(\frac{\nu_k}{2}, \frac{d_k}{2})$$

Où les statistiques suffisantes sont mises à jour de la manière suivante,

- $D_k = x_{k-1}^2 C_{k-1} + 1$
- $C_k = C_{k-1} - C_{k-1}^2 x_{k-1}^2 \frac{1}{D_k}$
- $m_k = m_{k-1} + C_{k-1} x_{k-1} \frac{1}{D_k} (x_k - x_{k-1} m_{k-1})$
- $d_k = d_{k-1} + (x_k - x_{k-1} m_{k-1})^2 \frac{1}{D_k}$
- $\nu_k = \nu_{k-1} + q$, avec $q = \dim x_k$

Estimated parameters with Storvik's SIR filter
with 1000 particles and 100 iterations on linear model

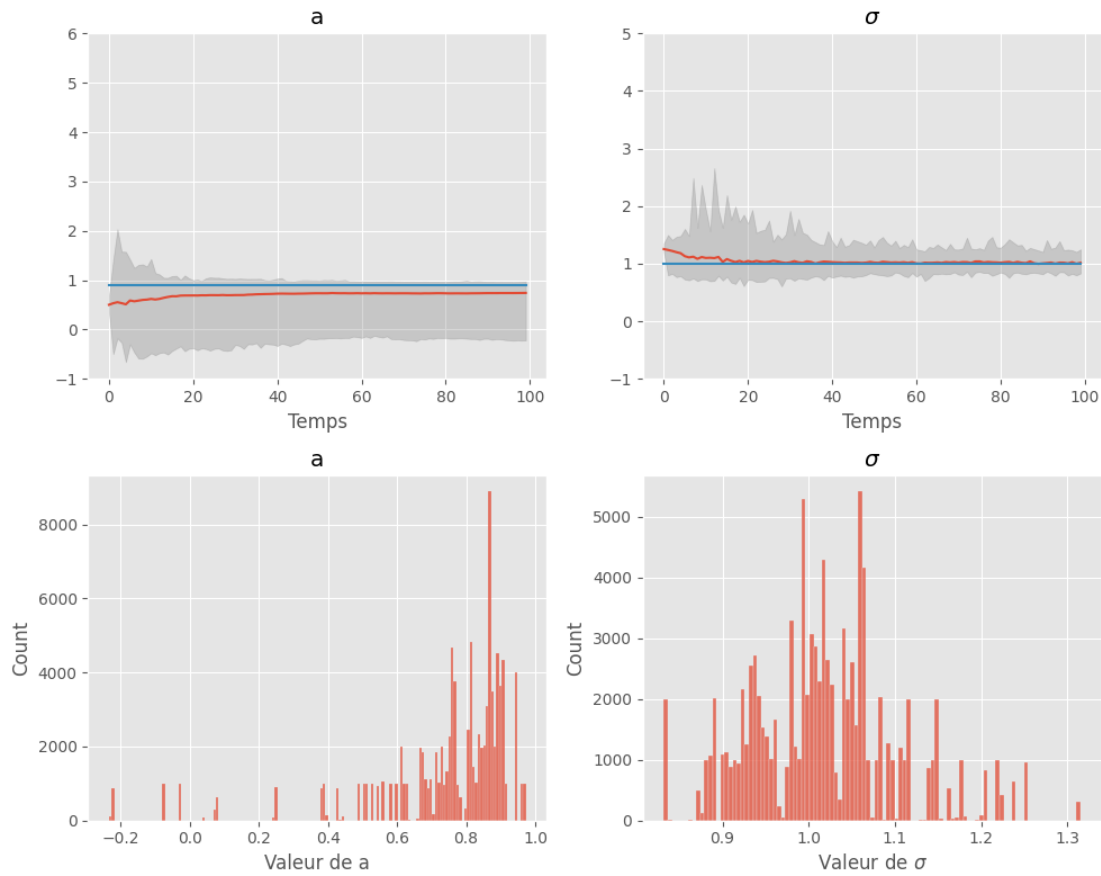
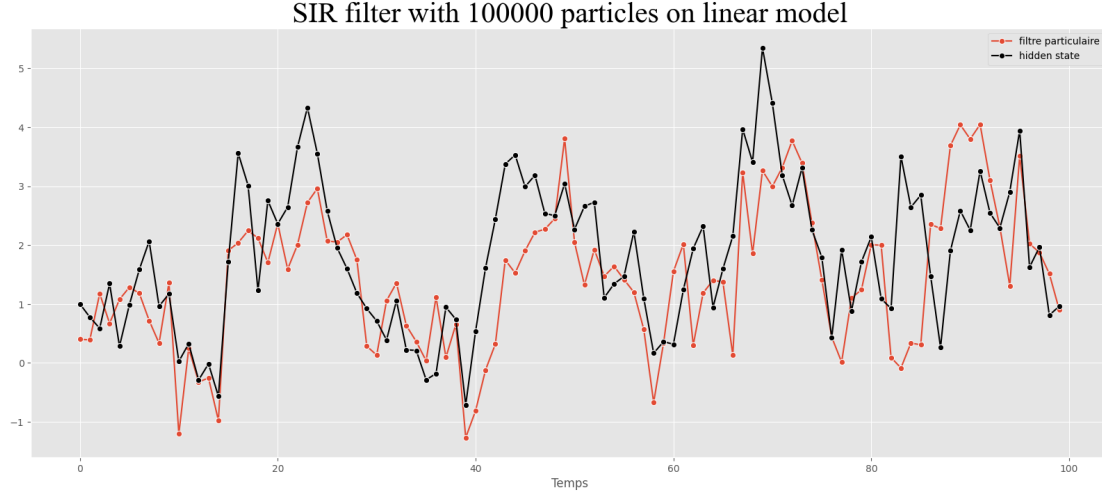


FIGURE 3 – Filtre de Storvik sur 100 trajectoires différentes, chaque trajectoire étant estimée avec 1000 particules

On obtient alors $a_{estim} = 0.85$ et $\sigma_{estim} = 1.21$. On peut donc maintenant run un SIR classique avec ces paramètres estimés comme précédemment.

FIGURE 4 – SIR classique avec les paramètres estimées a_{estim} et σ_{estim}

6.2 Modèle de Kitagawa

Le modèle de Kitagawa, développé par Sadanori Konishi et Genshiro Kitagawa, est un modèle théorique important. Il a été proposé pour résoudre des problèmes d'estimation statistique, tels que l'estimation de l'état d'un système dynamique à partir d'observations bruitées. Il sert souvent de *benchmark* afin de tester des méthodes d'estimation d'états cachés. L'un des avantages majeurs du modèle de Kitagawa est sa capacité à gérer des systèmes non linéaires. Contrairement aux approches classiques qui supposent souvent des relations linéaires entre les variables, ce modèle permet de modéliser des systèmes réels plus complexes, où les interactions entre les variables peuvent être non linéaires. Cela en fait un outil précieux dans de nombreux domaines où les phénomènes étudiés sont intrinsèquement non linéaires.

$$\begin{cases} x_t = \alpha x_{t-1} + \beta \frac{x_{t-1}}{1+x_{t-1}^2} + \gamma \cos(1.2(t-1)) + \omega_t, \text{ avec } \omega_t \sim \mathcal{N}(0, W), \\ y_t = \frac{x_t^2}{20} + \nu_t, \text{ avec } \nu_t \sim \mathcal{N}(0, V), \end{cases}$$

En y appliquant l'algorithme SIR énoncé précédemment, avec des paramètres fixés $\alpha=0.5$, $\beta=25$, $\gamma=8$, $V=5$ et $W=1$, on obtient :

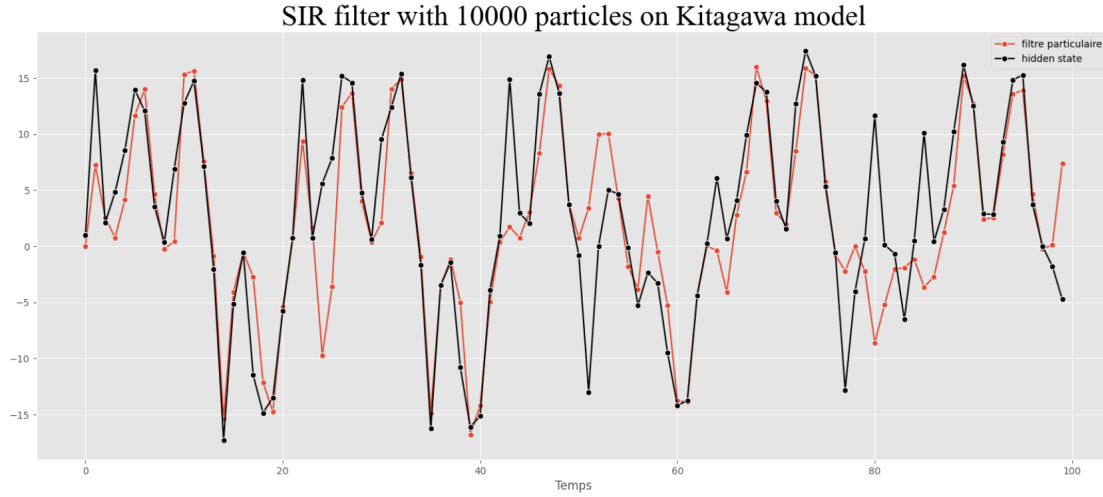


FIGURE 5 – SIR avec 10000 particules

En réalité, les paramètres des modèles sont inconnus. On applique donc ensuite l'algorithme de Storvik qui permet d'estimer simultanément les paramètres du modèle et l'état caché.

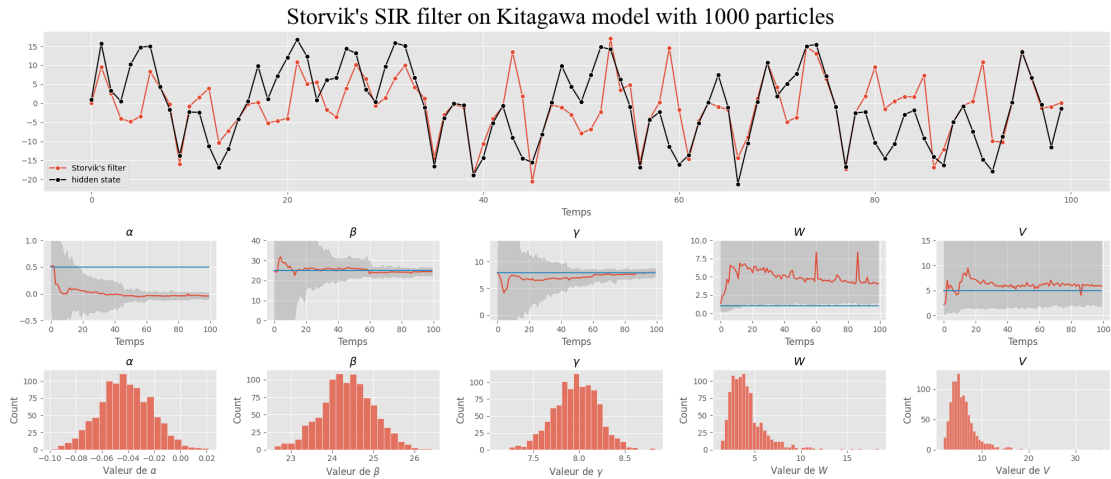


FIGURE 6 – Application du Storvik avec 1000 particules

On obtient alors $\alpha_{estim} = -0.08$, $\beta_{estim} = 26.49$, $\gamma_{estim} = 8.20$, $W_{estim} = 3.21$, $V_{estim} = 4.37$. En faisant tourner avec ses paramètres estimés l'algorithme du SIR on a alors le profil d'estimation suivant :

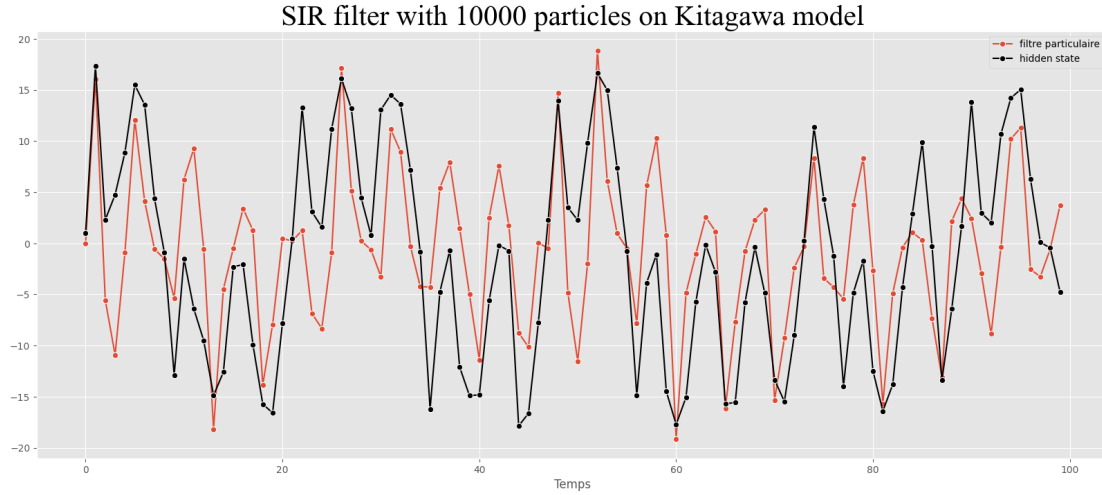


FIGURE 7 – SIR classique avec les paramètres estimés $\alpha_{estim}, \beta_{estim}, \gamma_{estim}, W_{estim}$ et V_{estim}

6.3 Modèle de volatilité Stochastique

6.3.1 Application théorique

$$\begin{cases} x_t = \alpha + \beta x_{t-1} + \omega_t, \text{ avec } \omega_t \sim \mathcal{N}(0, W), \\ y_t = \exp\left(\frac{x_t}{2}\right) \nu_t, \text{ avec } \nu_t \sim \mathcal{N}(0, 1), \end{cases}$$

Ce modèle, contrairement au précédent, modélise la volatilité comme un processus indépendant. Nous fixons les paramètres à $\alpha = 0$, $\beta = 0.97$ et $W = 1$.

6.3.2 Application d'un SIR en supposant les paramètres connus

En y appliquant l'algorithme SIR (2), avec des paramètres fixés on obtient :

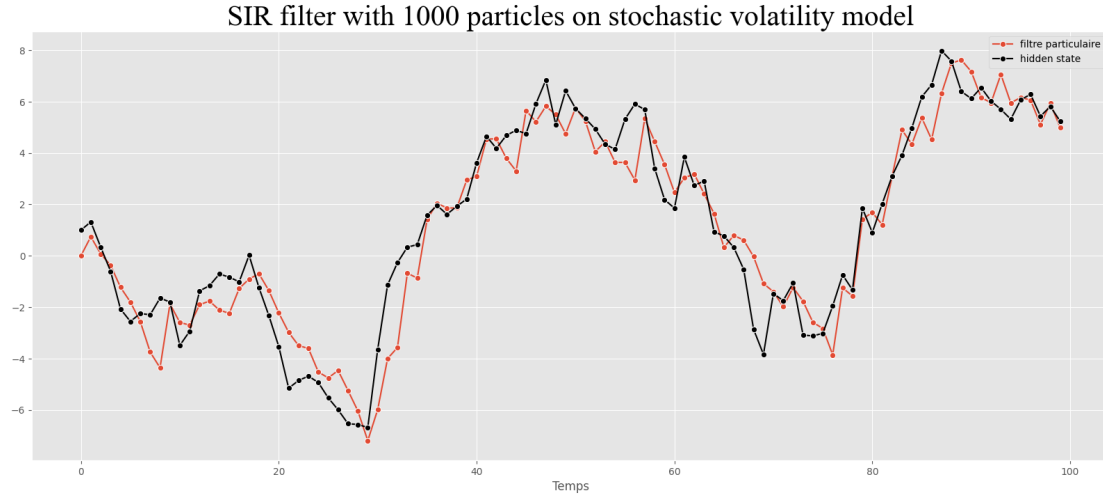


FIGURE 8 – Application du SIR avec 1000 particules

6.3.3 Application d'un filtre de Storvik

En réalité, les paramètres des modèles sont inconnus. On applique donc ensuite l'algorithme de Storvik qui permet d'estimer simultanément les paramètres du modèle et l'état caché.

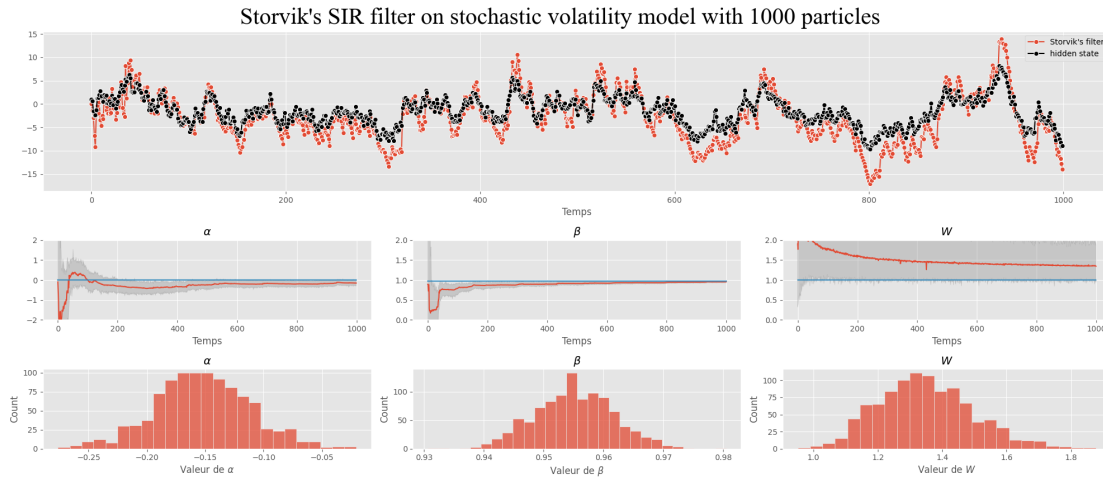


FIGURE 9 – Exemple d'application du Storvik avec 1000 particules

On trouve ainsi, $W_{estim} = 1.35$, $\alpha_{estim} = -0.152$ et $\beta_{estim} = 0.96$. α_{estim} et β_{estim} convergent vers les vraies valeurs de β , α . Cependant, nous pouvons voir que W_{estim} converge vers une valeur supérieure à W . Nous retrouverons cette tendance tout le long de l'application.

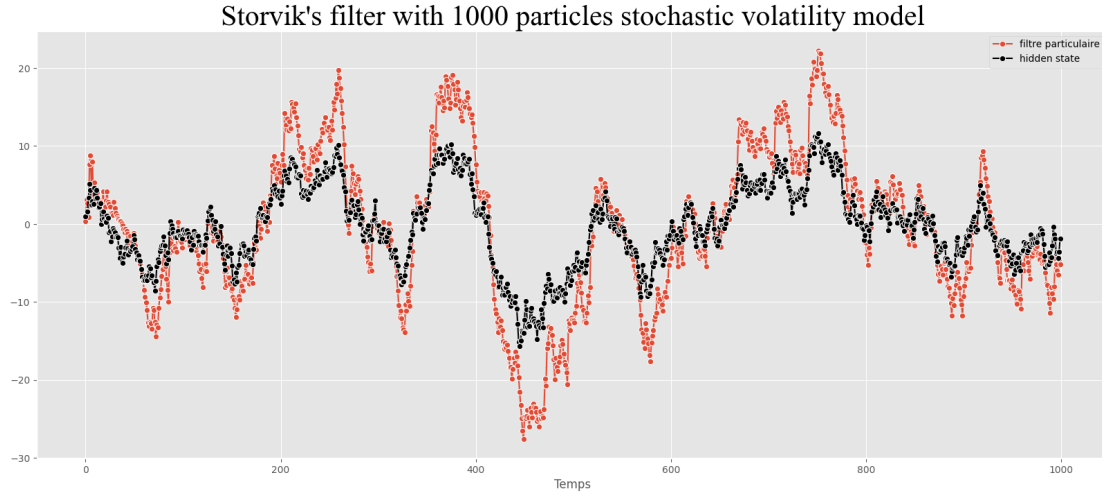


FIGURE 10 – États cachés et son estimation par un filtre de Storvik avec 1000 particules

Application d'un filtre de Storvik sur plusieurs trajectoires Nous pourrions nous contenter de run une seule fois le filtre de Storvik sur une plage de données grande pour estimer les paramètres du modèles, mais voici un exemple d'estimation des paramètres du modèle moyenné sur 10 trajectoires :

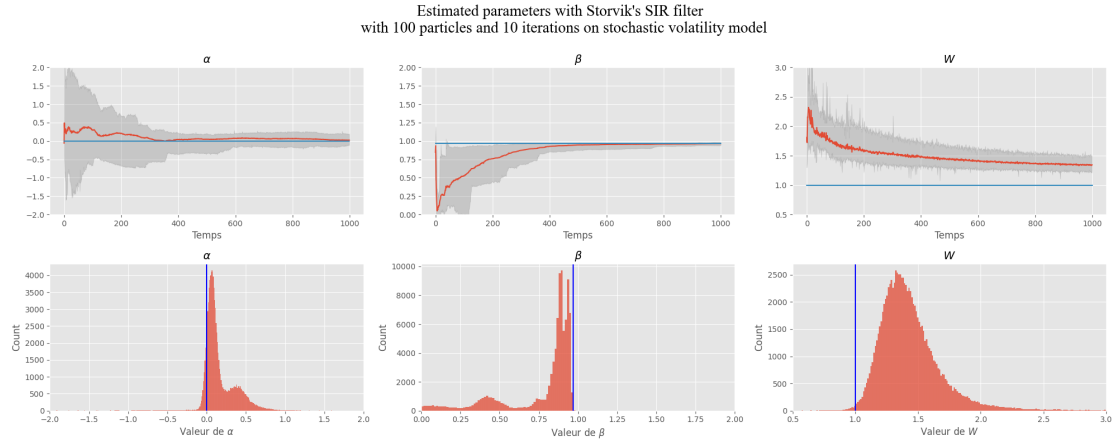


FIGURE 11 – Estimation des paramètres du modèle sur 10 trajectoires avec le Storvik, en gris : intervalle de confiance à 95%

On peut voir que la convergence est particulièrement rapide, il n'est donc pas nécessaire d'avoir plus d'itération pour estimer les paramètres du modèle avec le filtre de Storvik. On trouve $W_{estim} = 1.35$, $\alpha_{estim} = 0.025$ et $\beta_{estim} = 0.97$.

6.3.4 Application d'un PLS

Ici, nous appliquons un PLS (4) :

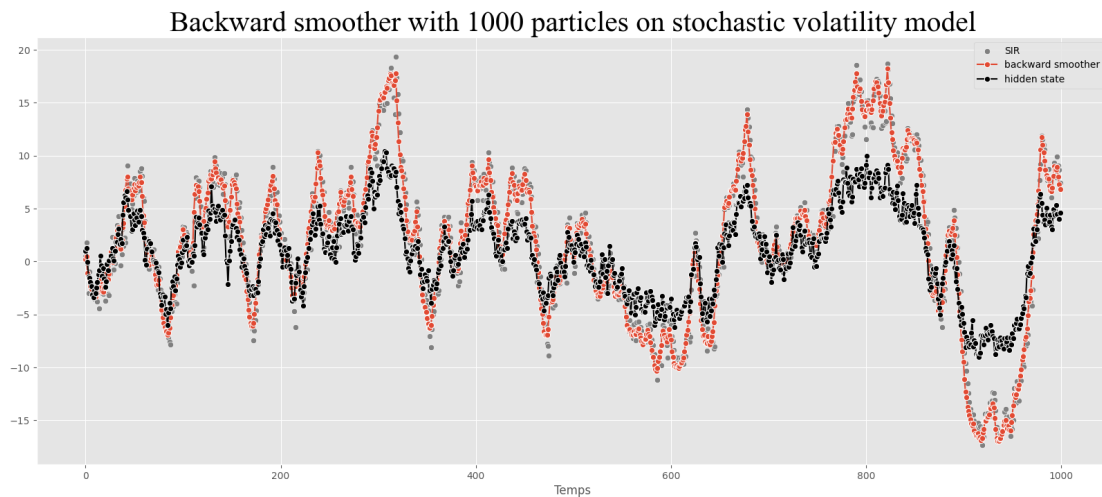


FIGURE 12 – Application d'un PLS

On trouve $W_{estim} = 1.27$, $\alpha_{estim} = 0.04$ et $\beta_{estim} = 0.98$. Comme annoncé dans l'article [1], le PLS n'améliore pas vraiment l'estimation obtenue par le *forward filter* appliqué (ici un Storvik) et donne de résultats médiocres de manière générale. Cet article présente plusieurs améliorations du PLS que nous n'avons pas eu le temps d'explorer. En contrepartie de cette relative efficacité du PLS, nous verrons qu'appliquer deux *forward filter* consécutivement : un Storvik, puis un SIR avec les paramètres estimés du Storvik, nous obtiendrons une estimation de l'état caché bien meilleure qu'avec un PLS.

6.3.5 Application d'un filtre de Storvik puis d'un SIR avec les paramètres estimés

Ici nous allons tester de run un Storvik pour estimer les paramètres du modèle, puis, au lieu de lancer un backward smoother, nous relançons un SIR avec les paramètres estimés. Avec le Storvik, nous trouvons $W_{estim} = 1.43$, $\alpha_{estim} = 0.20$ et $\beta_{estim} = 0.95$.

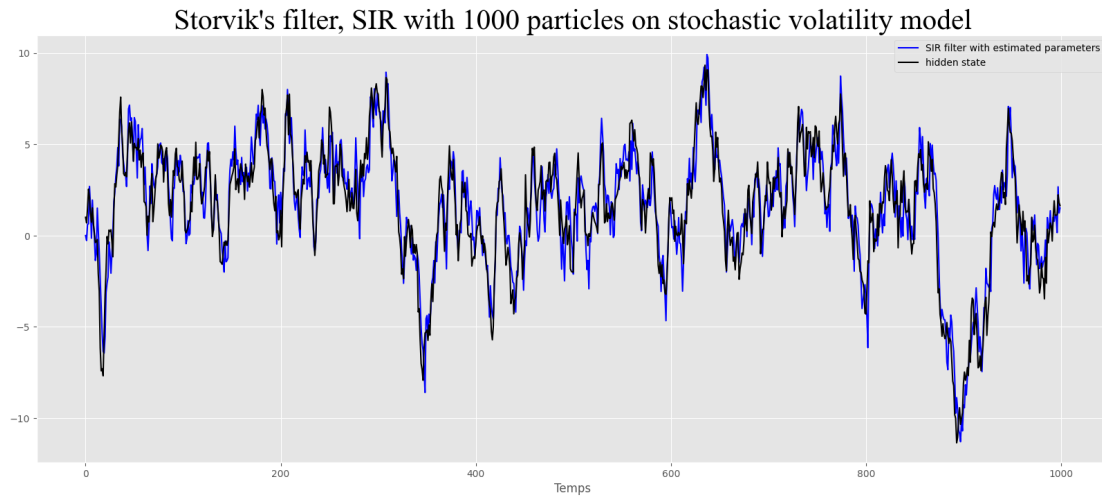


FIGURE 13 – Hidden states et SIR avec les paramètres estimés par un Storvik avec 1000 particules

Nous pouvons voir que cela fonctionne particulièrement bien, cette façon de run deux forward filters (un Storvik puis un SIR en utilisant les paramètres estimés du Storvik) outperforms le PLS. Nous allons confirmer cela ci-dessous.

6.3.6 Comparaison entre PLS, filtre de Storvik et SIR

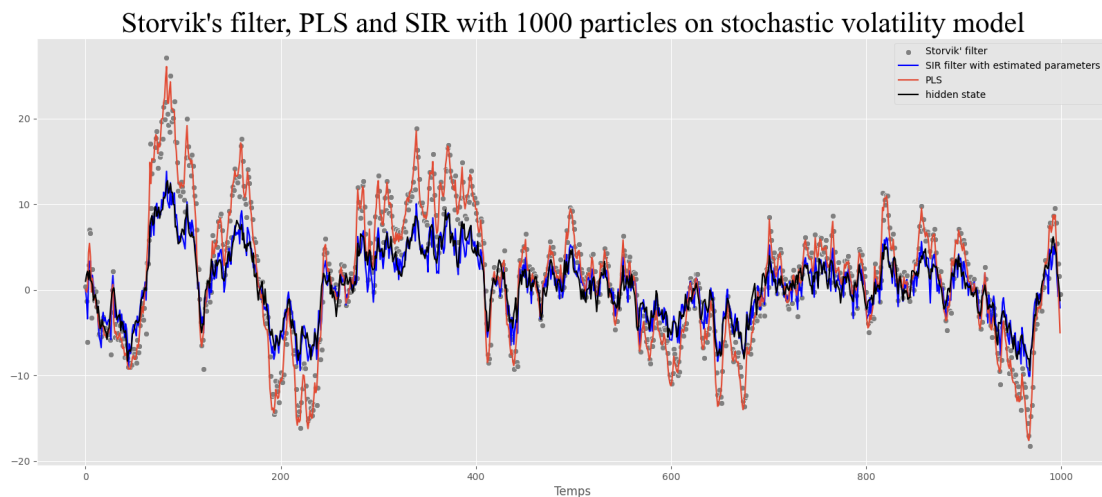


FIGURE 14 – Comparaison entre le filtre de Storvik, le PLS et le SIR avec les paramètres estimés par le Storvik

Avec le Storvik, nous trouvons $W_{estim} = 1.26$, $\alpha_{estim} = 9.637 \times 10^{-5}$ et $\beta_{estim} = 0.98$. Il est clair ici que le double forward filters donne des résultats bien meilleurs qu'un PLS, ou encore simplement d'un Storvik.

6.4 Application pour l'estimation de la volatilité du S&P 500

Enfin pour terminer ce rapport, appliquons ce que nous avons vu pour essayer d'estimer la volatilité d'un indice financier, le S&P 500. Comme nous l'avons vu dans l'introduction, connaître la volatilité du marché est un élément important, ne serait-ce que pour faire fonctionner des modèles préétablies (par exemple le modèle de Black-Schole).

On se place dans une période relativement volatile : pendant la crise financière de 2008. L'état caché que l'on cherche à estimer est donc la volatilité X_t et nous possédons une volatilité observée Y_t (qui est obtenu en par le cours de clôture de l'indice). On suppose donc que le X_t et Y_t suivent le modèle indiqué dans la section 6.3.1. Enfin on a :

$$Y_t = \log\left(\frac{P_t}{P_{t-1}}\right)$$

où P_t est le prix de clôture du jour t .

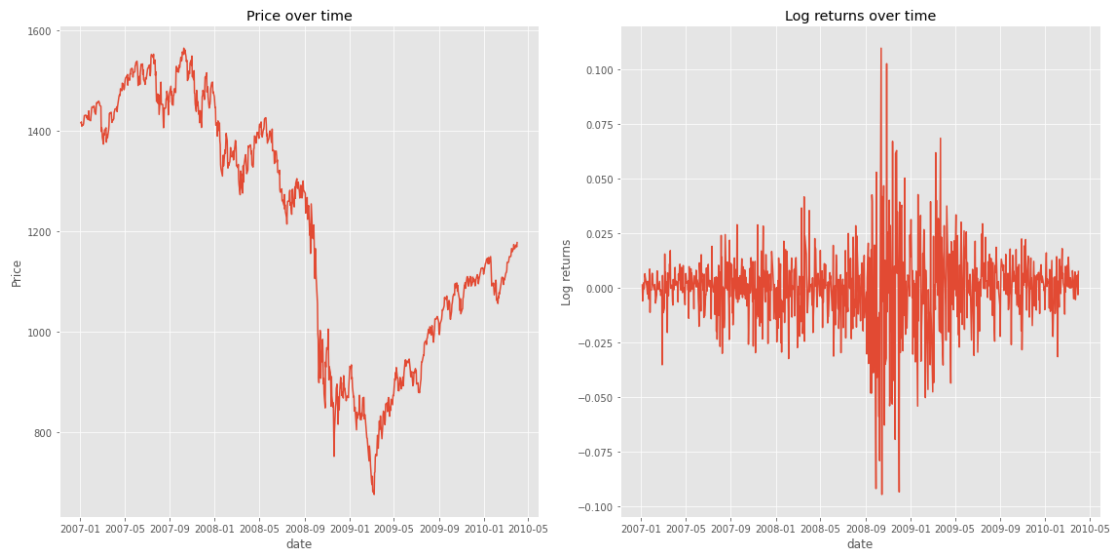


FIGURE 15 – Variation du prix de l'indice et de Y_t de 2007 à 2010

On remarque qu'on observe un très forte volatilité vers le dernier trimestre de

2008. On ne connaît évidemment pas les paramètres du modèle, il faut donc les estimer, puis en appliquer le SIR de nouveau sur ces paramètres,

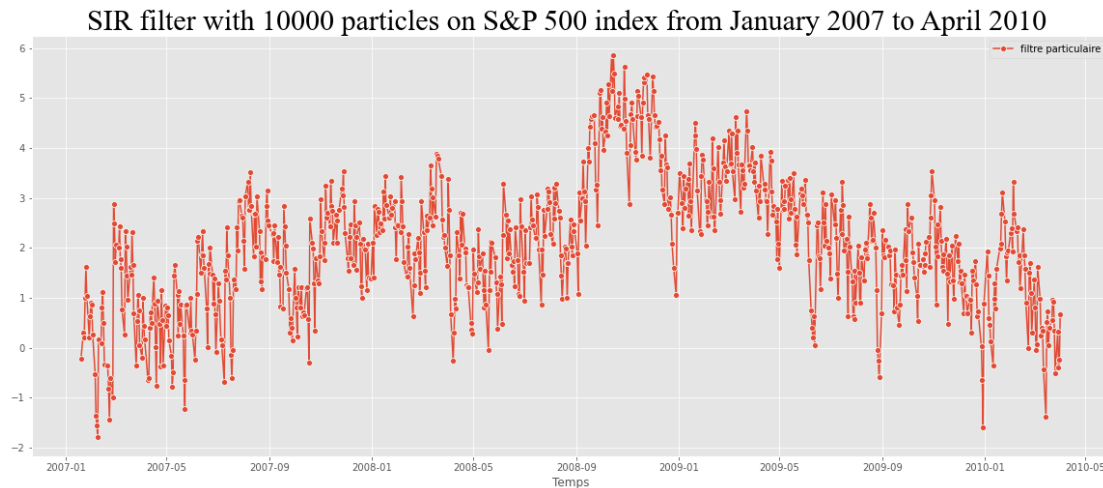


FIGURE 16 – État caché.

Une tendance notable émerge en matière de volatilité durant la période de 2008 à 2009, une phase notoirement tumultueuse sur les marchés financiers. Durant cette période, les estimations fournies par le filtre particulaire indiquent une volatilité cachée particulièrement élevée. Ce phénomène est à la fois intuitif et conforme à nos attentes, étant donné que des périodes de crises financières sont généralement caractérisées par une volatilité accrue, qui se reflète dans une estimation du processus caché substantiellement plus grande.

Néanmoins, ce qui est particulièrement frappant, c'est l'efficacité du filtre particulaire à réagir à l'évolution des conditions de marché. En effet, à mesure que la volatilité du marché se calme, le filtre particulaire, de façon assez impressionnante, parvient à faire retomber ses estimations du processus caché à des niveaux plus modérés, plus en accord avec des périodes de stabilité relative.

Ceci est particulièrement évident lorsque l'on compare ces estimations avec celles obtenues lors des périodes dites "plus calmes", par exemple, au début et à la fin des séquences graphiques. On observe que les estimations du processus caché reviennent à des niveaux similaires à ceux d'avant la crise financière, suggérant que le filtre particulaire fait preuve d'une remarquable capacité d'adaptation aux variations du marché, à la fois en période de turbulences et de stabilité.

Bibliographie

- [1] S. GODSILL A. DOUCET et C. ANDRIEU. “On sequential Monte Carlo sampling methods for Bayesian filtering”. In : *Statistics and Computing* 10 (1999), p. 197-208.
- [2] Lopes CARVALHO Johannes et POLSON. “Particle learning and smoothing”. In : *Statistical Science* 25 (2010), p. 88-106.
- [3] D. CRISAN. “Exact rates of convergence for a branching particle approximation to the solution of the zakai equation”. In : *Annals of Probability* 32 (2003), p. 819-838.
- [4] D. CRISAN et A. DOUCET. “A survey of convergence results on particle filtering methods for practitioners”. In : *IEEE Transactions on Signal Processing* 50 3 (2002), p. 736-746.
- [5] P. Del Moral D. CRISAN et T. J. LYONS. “Interacting particle systems approximations of the Kushner Stratonovitch equation, Advances in Applied Probability”. In : *International Journal of Control* (1999), p. 819-838.
- [6] J. E. HANDSCHIN et D. Q. MAYNE. “Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering”. In : *International Journal of Control* 9.5 (1969), p. 547-559.
- [7] J. S. LIU et R. CHEN. “Sequential Monte Carlo methods for dynamic systems”. In : *J. Amer. Statist. Assoc.* 93.443 (1998), p. 1032-1044.
- [8] P. Del MORAL et L. MICLO. “Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering”. In : *Séminaire de Probabilités, Lecture Notes in Mathematics* 1729.XXXIV (2000), p. 1-145.
- [9] D.J. Salmond N.J. GORDON et A.F.M. SMITH. “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”. In : *IEE Proceedings F Radar and Signal Processing* 140 2 (1993).
- [10] D. B. RUBIN. “MUsing the SIR algorithm to simulate posterior distributions, Bayesian Statistics”. In : *Oxford University Press* 3 (1988).
- [11] G. STORVIK. “Particle Filters for State-Space Models With the Presence of Unknown Static Parameters”. In : *IEEE* 50.2 (2002).