



HAL
open science

Algorithmes de restauration bayésienne mono- et multi-objets dans des modèles markoviens

Yohan Petetin

► **To cite this version:**

Yohan Petetin. Algorithmes de restauration bayésienne mono- et multi-objets dans des modèles markoviens. Mathématiques générales [math.GM]. Institut National des Télécommunications, 2013. Français. NNT : 2013TELE0032 . tel-00939083

HAL Id: tel-00939083

<https://theses.hal.science/tel-00939083v1>

Submitted on 30 Jan 2014

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

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



École Doctorale en Sciences Mathématiques de Paris-Centre ED386

THÈSE DE DOCTORAT
Discipline : Mathématiques Appliquées

présentée par

Yohan PETETIN

pour obtenir le grade de Docteur de Télécom SudParis et de l'Université Pierre et Marie Curie

**Algorithmes de restauration Bayésienne mono- et multi-
objet(s) dans des modèles Markoviens**

dirigée par François DESBOUVRIES

Soutenue le 27 Novembre 2013 devant le jury composé de :

Pr. Jean-Yves TOURNERET	INP-ENSEEIH	Président
Pr. Christophe ANDRIEU	Bristol University	Rapporteur
Pr. Nicolas CHOPIN	CREST / ENSAE	Rapporteur
Dr. Jacques BLANC-TALON	DGA - MRIS	Examineur
Pr. Michel BRONIATOWSKI	Université de Paris VI	Examineur
Pr. Emmanuel DUFLOS	École Centrale de Lille	Examineur
Pr. François ROUEFF	Télécom ParisTech	Examineur
Pr. François DESBOUVRIES	Télécom SudParis	Directeur de thèse

Thèse n° 2013TELE0032

Remerciements

Je tiens tout d'abord à exprimer, à double titre, ma gratitude envers mon directeur de thèse, François Desbouvries. Sa pédagogie, sa rigueur et la passion pour les mathématiques appliquées qu'il m'a transmises lorsque j'étais étudiant ingénieur à Télécom SudParis m'ont poussé à me lancer dans cette grande aventure qu'est la thèse de doctorat. Grâce à son soutien, j'ai pu mener à bien mon projet de recherche. Je le remercie par ailleurs de m'avoir accordé tout le temps nécessaire à la préparation de cette thèse ; l'aboutissement de ce travail est en grande partie dû à tous les conseils que j'ai pu recevoir de sa part.

Je souhaite également remercier Wojciech Pieczynski, directeur du département Communications Images et Traitement de l'Information (CITI) de Télécom SudParis, pour la confiance qu'il m'a accordée en me permettant d'exercer, en plus de mon activité de recherche, une activité d'enseignement extrêmement formative au sein de Télécom SudParis, et la possibilité d'encadrer des étudiants pour leurs projets de fin d'études dans le parcours de spécialisation dont il est responsable.

Je remercie Emmanuel Monfrini pour nos innombrables discussions, qu'elles soient scientifiques ou non, et je le remercie de m'avoir intégré dans les différents projets qu'il a eu l'occasion d'encadrer durant ces dernières années. Sa bonne humeur constante et sa forte implication dans les enseignements dont il est responsable ont été (et restent) un modèle à suivre pour les années à venir.

De manière plus générale, je remercie toute l'équipe du département CITI, Randal Douc pour ses connaissances fines des méthodes de Monte Carlo, Frédéric Lehman, Marc Castella, Jean-Pierre Delmas, Roger Lamberti, François Simin, Marc Uro, Luc Leclère et Julie Bonnet. Leur sympathie ont fait de mon cadre de travail le plus agréable qui soit.

Parmi les autres personnes que j'ai côtoyées durant ces trois années, je me dois de citer certains thésards de Télécom SudParis, Yazid, Fethi, Florian, Mustapha ; tous ont contribué à rendre ce travail agréable. Je remercie également, pour leur bonne humeur, les trois dernières promotions ingénieur de Télécom SudParis (2010-2013, 2011-2014 et 2012-2015) que j'ai pu sensibiliser à l'Analyse Numérique, au Traitement du Signal, et à la Statistique Mathématique. J'espère qu'ils ne me tiendront pas rigueur des séances de Travaux Pratiques à huit heures du matin.

Enfin, et il s'agit du point le plus important, je remercie mes proches et mes amis qui m'ont soutenu activement pendant toutes ces années. Mes parents, mon frère et Yara ont été ma source de motivation la

plus importante, et leur présence constante durant ces trois dernières années m'a permis de travailler dans les meilleures conditions possibles.

Résumé

Cette thèse est consacrée au problème d'estimation bayésienne pour le filtrage statistique, dont l'objectif est d'estimer récursivement des états inconnus à partir d'un historique d'observations, dans un modèle stochastique donné. Les modèles stochastiques considérés incluent principalement deux grandes classes de modèles : les modèles de Markov cachés et les modèles de Markov à sauts conditionnellement markoviens. Ici, le problème est abordé sous sa forme générale dans la mesure où nous considérons le problème du filtrage mono- et multi objet(s), ce dernier étant abordé sous l'angle de la théorie des ensembles statistiques finis et du filtre « Probability Hypothesis Density ».

Tout d'abord, nous nous intéressons à l'importante classe d'approximations que constituent les algorithmes de Monte Carlo séquentiel, qui incluent les algorithmes d'échantillonnage d'importance séquentiel et de filtrage particulière auxiliaire. Les boucles de propagation mises en jeu dans ces algorithmes sont étudiées et des algorithmes alternatifs sont proposés. Les algorithmes de filtrage particulière dits « localement optimaux », c'est à dire les algorithmes d'échantillonnage d'importance avec densité d'importance conditionnelle optimale et de filtrage particulière auxiliaire pleinement adapté sont comparés statistiquement, en fonction des paramètres du modèle donné.

Ensuite, les méthodes de réduction de variance basées sur le théorème de Rao-Blackwell sont exploitées dans le contexte du filtrage mono- et multi-objet(s). Ces méthodes, utilisées principalement en filtrage mono-objet lorsque la dimension du vecteur d'état à estimer est grande, sont dans un premier temps étendues pour les approximations Monte Carlo du filtre Probability Hypothesis Density. D'autre part, des méthodes de réduction de variance alternatives sont proposées : bien que toujours basées sur le théorème de Rao-Blackwell, elles ne se focalisent plus sur le caractère spatial du problème mais plutôt sur son caractère temporel.

Enfin, nous abordons l'extension des modèles probabilistes classiquement utilisés. Nous rappelons tout d'abord les modèles de Markov couple et triplet dont l'intérêt est illustré à travers plusieurs exemples pratiques. Ensuite, nous traitons le problème de filtrage multi-objets, dans le contexte des ensembles statistiques finis, pour ces modèles. De plus, les propriétés statistiques plus générales des modèles triplet sont exploitées afin d'obtenir de nouvelles approximations de l'estimateur bayésien optimal (au sens de l'erreur quadratique moyenne) dans les modèles à sauts classiquement utilisés ; ces approximations peuvent produire des estimateurs de performances comparables à celles des approximations particulières, mais ont l'avantage d'être moins coûteuses sur le plan calculatoire.

Mots-clefs

Filtrage particulière ; méthodes de Monte Carlo ; Rao-Blackwell ; Probability Hypothesis Density ; filtrage multi-objets ; modèles de Markov cachés ; modèles de Markov couple et triplet.

Abstract

This thesis focuses on the Bayesian estimation problem for statistical filtering which consists in estimating hidden states from an historic of observations over time in a given stochastic model. The considered models include the popular Hidden Markov Chain models and the Jump Markov State Space Systems ; in addition, the filtering problem is addressed under a general form, that is to say we consider the mono- and multi-object filtering problems. The latter one is addressed in the Random Finite Sets and Probability Hypothesis Density contexts.

First, we focus on the class of particle filtering algorithms, which include essentially the sequential importance sampling and auxiliary particle filter algorithms. We explore the recursive loops for computing the filtering probability density function, and alternative particle filtering algorithms are proposed. The “locally optimal” filtering algorithms, i.e. the sequential importance sampling with optimal conditional importance distribution and the fully adapted auxiliary particle filtering algorithms, are statistically compared in function of the parameters of a given stochastic model.

Next, variance reduction methods based on the Rao-Blackwell theorem are exploited in the mono- and multi-object filtering contexts. More precisely, these methods are mainly used in mono-object filtering when the dimension of the hidden state is large ; so we first extend them for Monte Carlo approximations of the Probability Hypothesis Density filter. In addition, alternative variance reduction methods are proposed. Although we still use the Rao-Blackwell decomposition, our methods no longer focus on the spatial aspect of the problem but rather on its temporal one.

Finally, we discuss on the extension of the classical stochastic models. We first recall pairwise and triplet Markov models and we illustrate their interest through several practical examples. We next address the multi-object filtering problem for such models in the random finite sets context. Moreover, the statistical properties of the more general triplet Markov models are used to build new approximations of the optimal Bayesian estimate (in the sense of the mean square error) in Jump Markov State Space Systems. These new approximations can produce estimates with performances alike those given by particle filters but with lower computational cost.

Keywords

Particle filtering ; Monte Carlo methods ; Rao-Blackwell ; Probability Hypothesis Density ; multi-object filtering ; Hidden Markov models ; Pairwise and Triplet Markov models.

Table des matières

Remerciements	3
Résumé	5
Abstract	7
Notations	13
Avant propos	15
1 Introduction	19
1.1 Filtrage statistique mono-objet	20
1.1.1 Modèles de Markov cachés	20
1.1.2 Modèles de Markov cachés à sauts markoviens	21
1.2 Filtrage statistique multi-objets	22
1.2.1 Problèmes et contraintes	23
1.2.2 Formulation statistique du problème de filtrage multi-objets	23
1.2.3 Critère de performance	26
1.3 Contributions de cette thèse	27
1.4 Publications de l’auteur	30
1.4.1 Articles de revues internationales avec comité de lecture	30
1.4.2 Articles de conférences internationales avec actes et comité de lecture	30
2 Méthodes de Monte Carlo séquentielles	33
2.1 Le filtrage particulaire et le filtrage particulaire auxiliaire	34
2.1.1 Principe de l’échantillonnage d’importance	34
2.1.2 Considérations pratiques	36
2.1.3 Le filtrage particulaire auxiliaire	37
2.2 Algorithmes de filtrage particulaire direct, par prédiction et par lissage	39
2.2.1 Quelques mécanismes de simulations	40
2.2.2 Filtrage direct	41
2.2.3 Filtrage prédictif	42
2.2.4 Filtrage lissé	44
2.3 Implémentation pratique des algorithmes de filtrage particulaire	45

2.3.1	Simulations dans un modèle linéaire et gaussien	46
2.3.2	Le PS-APF	46
2.3.3	Discussion	48
2.3.4	Simulations dans le modèle de Kitagawa	49
2.3.5	Scénario de poursuite angle-distance	50
2.3.6	Modèle semi-linéaire	53
2.4	Comparaison théorique d’algorithmes de filtrage localement optimaux	53
2.4.1	Estimateurs MC et mélanges de lois de probabilité	54
2.4.2	Algorithmes localement optimaux SIS, SIR et FA	57
2.4.3	Comparaison de $\hat{\Theta}_n^{\text{SIR},N'}$ et de $\hat{\Theta}_n^{\text{FA},N'}$	59
2.4.4	Comparaison de $\hat{\Theta}_n^{\text{SIR},N'}$ et $\hat{\Theta}_n^{\text{SIS}}$	60
2.4.5	Comparaison de $\hat{\Theta}_n^{\text{SIS}}$ et $\hat{\Theta}_n^{\text{FA},N'}$	63
2.4.6	Simulations	67
2.5	Conclusion	75
3	Techniques de réduction de variance Rao-Blackwellisées pour les méthodes SMC	77
3.1	État de l’art	77
3.2	Filtres particuliers PHD marginalisés pour le filtrage multi-objets bayésien	80
3.2.1	Quelques rappels sur le filtre PHD	81
3.2.2	Le filtre M-PHD pour les modèles partiellement linéaires et gaussiens	82
3.2.3	Implémentation pratique du filtre M-PHD	85
3.2.4	D’autres applications du filtre M-PHD	89
3.2.5	Simulations	91
3.3	Méthodes Monte Carlo conditionnelles temporelles	96
3.3.1	Estimateurs CMC pour les modèles HMC	97
3.3.2	Estimateurs CMC pour des modèles à sauts (classe 2)	104
3.3.3	Estimateurs CMC pour des modèles à sauts (classe 1)	107
3.3.4	Estimateurs CMC pour le filtrage multi-objets	111
3.4	Conclusion	115
4	Modèles de Markov couple et triplet	119
4.1	Généralités sur les modèles de Markov couple et triplet	120
4.1.1	Les limitations des modèles HMC	120
4.1.2	Modèles de Markov couple	121
4.1.3	Filtrage dans les modèles de Markov couple	122
4.1.4	De l’importance des modèles couple	123
4.1.5	Modèles de Markov couple physiquement contraints	125
4.1.6	Modèles de Markov triplet	127
4.1.7	Filtrage dans les modèles de Markov triplet	128
4.2	Filtrage multi-objets pour les chaînes de Markov couple	130
4.2.1	Les limites de l’approche classique	131
4.2.2	Un filtre PHD pour les modèles PMC	132
4.2.3	Une implémentation par mélanges de gaussiennes du filtre PMC-PHD	133
4.2.4	Apport du PMC-PHD : simulations	137

4.3	Une classe d'algorithmes de filtrage exact rapide pour des modèles de Markov à sauts	140
4.3.1	Une classe particulière de modèles de Markov triplet	141
4.3.2	Une solution approchée du problème de filtrage	143
4.3.3	Discussion	144
4.3.4	Analyses de performances et simulations	145
4.4	Conclusion	150
Conclusion et perspectives		155
A Conditionnement dans les vecteurs aléatoires gaussiens		157
A.1	Résultat 1	157
A.2	Résultat 2	157
B Preuves des résultats du Chapitre 2		159
B.1	Preuve de la Proposition 2.1	159
B.2	Preuve de la Proposition 2.2	160
B.3	Preuve de la Proposition 2.3	160
C Preuves des résultats du Chapitre 4		163
C.1	Preuve de la Proposition 4.1	163
C.2	Preuve des Propositions 4.2 à 4.3	164
C.3	Preuve de la Proposition 4.4 (filtrage exact dans les modèles de Markov triplet)	165
C.4	Preuve de la Proposition 4.8	165
D Un pisteur robuste basé sur une approche CPHD		167
D.1	L'implémentation par mélange de gaussiennes du CPHD	167
D.1.1	Prédiction CPHD	168
D.1.2	Mise à jour CPHD	168
D.2	Un pisteur robuste basé sur une approche CPHD	169
D.2.1	Clustering	169
D.2.2	Gating	170
D.2.3	Loi de probabilité du nombre de cibles	170
D.2.4	Intensité des mesures de fausses alarmes	170
D.2.5	Gestion des pistes	170
D.2.6	Initialisation des pistes	171
D.3	Simulations	171
Bibliographie		182

Notations

Abréviations

Nous présentons ici la liste des abréviations qui seront utilisées tout au long de cette thèse. Les abréviations couramment utilisées dans la littérature scientifique seront tirées de la langue anglaise. Lors de leur première apparition dans le texte principal, nous donnerons leur traduction en français et en anglais.

APF	Filtrage Particulaire Auxiliaire
ARCH	AutoRégressif Conditionnellement Hétéroscédastique
CMC	Monte Carlo Conditionnel
CPHD	Cardinalized Probability Hypothesis Density
DKL	Divergence de Kullback-Leibler
EKF	Filtre de Kalman Etendu
EQM	Erreur Quadratique Moyenne
EMM	Exact Matching Moment
FA-APF	Filtrage Particulaire Auxiliaire Pleinement Adapté
FK	Filtre de Kalman
FP	Filtrage/Filtre Particulaire
HMC	Chaîne de Markov Cachée
IMM	Multi-Modèles Interactifs
JMSS	Système d'espace d'état à sauts Markoviens
MC	Monte Carlo
MCMC	Monte Carlo par Chaîne de Markov
M-PHD	Probability Hypothesis Density Marginalisé
ND	Normal-Discret
OSPA	Optimal SubPattern Assignment
PHD	Probability Hypothesis Density
PMC	Chaîne de Markov couple
RB	Rao-Blackwell/Blackwellisé
SIS	Échantillonnage d'importance séquentiel
SIR	Échantillonnage d'importance avec Rééchantillonnage
SMC	Monte Carlo Séquentiel
SV	Volatilité Stochastique
UKF	Filtre de Kalman Sans Parfum
UT	Transformation Sans Parfum
cf.	confer
i.i.d.	indépendant et identiquement distribué
v.a.	variable aléatoire

Notations mathématiques

Dans tout ce document, les matrices et les vecteurs (éventuellement aléatoires) sont représentés par des majuscules grasses, et les variables aléatoires scalaires par des majuscules non grasses. Les réalisations de vecteurs aléatoires sont représentées par des minuscules grasses et les scalaires (éventuellement des réalisations de variables aléatoires réelles) par des minuscules non grasses.

\otimes	Produit de Kroenecker
\propto	Proportionnel à
\mathbf{A}^{-1}	Matrice Inverse de \mathbf{A}
\mathbf{A}^T	Matrice transposée de \mathbf{A}
$\text{diag}(\mathbf{A}_1, \mathbf{A}_2)$	Matrice diagonale par blocs formée par les matrices \mathbf{A}_1 et \mathbf{A}_2
\mathbf{I}_q	Matrice identité de dimension q
$p(\cdot)$	Densité de probabilité
$\mathcal{N}(\mathbf{x}; \mathbf{m}; \mathbf{P})$	Distribution Normale de moyenne \mathbf{m} et matrice de covariance \mathbf{P} , prise en \mathbf{x}
$\mathcal{N}(\cdot; \mathbf{m}; \mathbf{P})$	Loi normale de moyenne \mathbf{m} et matrice de covariance \mathbf{P}
$\mathbf{X} \sim p(\mathbf{x})$	\mathbf{X} suit la loi de densité de probabilité $p(\mathbf{x})$
$\mathbf{x}^i \sim p(\mathbf{x})$	\mathbf{x}^i est tiré selon la loi de densité de probabilité $p(\mathbf{x})$
$D_{\text{KL}}(p_1, p_2)$	DKL entre deux distributions de densités $p_1(\cdot)$ et $p_2(\cdot)$
\mathbf{X}_n	v.a représentant un état caché
\mathbf{Y}_n	v.a représentant une observation associée à un état caché
\mathbf{Z}_n	v.a représentant une mesure quelconque (détection ou fausse alarme)
ξ_n	Réalisation $(\mathbf{x}_n, \mathbf{y}_n) = \xi_n$ du couple (X_n, Y_n)
$\mathbf{x}_{0:n}$	Séquence de vecteurs $(\mathbf{x}_0, \dots, \mathbf{x}_n)$
Ξ_n	RFS des états cachés
\mathcal{X}_n	Réalisation $\{\mathbf{x}_{1,n}, \dots, \mathbf{x}_{m,n}\}$ du RFS Ξ_n
Ψ_n	RFS des observations
\mathcal{Z}_n	Réalisation du RFS Ψ_n
$ \mathcal{X} $	Cardinal de l'ensemble \mathcal{X}
$\mathbb{1}_V(\cdot)$	Fonction indicatrice sur l'ensemble V

Avant propos

Le filtrage statistique, qui consiste à estimer de façon séquentielle des données cachées à partir de données observées, est un problème de traitement statistique du signal qui englobe des applications aussi vastes que la poursuite d'objets mobiles, l'estimation de données financières ou le traitement de données en biologie. Les modélisations probabilistes pour traiter ce problème reposent sur des modèles stochastiques dans lesquels les données cachées sont considérées comme des variables aléatoires (v.a.), et les données observées comme des réalisations de v.a., auxquelles nous avons donc accès. Dans les modèles stochastiques dynamiques les plus souvent utilisés, l'état caché suit une dynamique markovienne connue et les observations sont reliées aux états par l'intermédiaire d'une loi donnée. Par conséquent, l'estimation de l'état inconnu peut être ramené à un problème d'inférence bayésienne (nous cherchons donc à minimiser, en moyenne, une fonction de risque donnée) dans une chaîne de Markov cachée, généralement non linéaire et/ou non gaussienne. Les techniques bayésiennes propagent dans le temps la loi conditionnelle de l'état sachant les observations. Néanmoins, le calcul de l'estimateur bayésien optimal est impossible dans de nombreux modèles car les formules de propagation impliquées font intervenir des intégrales incalculables et/ou des sommes dont le nombre de termes est exponentiel en temps. Il convient alors de recourir à des approximations afin d'approcher l'estimateur bayésien optimal : de nombreuses solutions approchées ont été (et continuent à être) proposées.

Parmi les approximations existantes, celles qui sont couramment utilisées sont les algorithmes de filtrage particulaire (FP), présentés et étudiés dans de nombreuses contributions (voir par exemple [55] [97] [43] [33]). Ces algorithmes reposent sur l'utilisation séquentielle de méthodes de Monte Carlo (*Sequential Monte Carlo* = SMC) et consistent à remplacer la propagation exacte de la loi conditionnelle par la propagation d'une mesure empirique construite sur des points tirés approximativement selon cette loi ; l'estimateur ponctuel bayésien optimal, qui est souvent le moment d'ordre un ou deux de cette loi a posteriori, est alors remplacé par la moyenne empirique de l'approximation particulaire de cette loi. Finalement, dans ces techniques SMC, les mesures empiriques propagées dans le temps ne sont qu'un intermédiaire de calcul, l'objectif étant l'estimateur ponctuel lui-même. Ces techniques calculent en définitive un estimateur de l'estimateur bayésien optimal. Par conséquent, à l'approximation inhérente à tout estimateur statistique s'ajoute une approximation due à son calcul approché. L'utilisation des méthodes SMC soulève néanmoins deux problèmes :

- comment propager récursivement une approximation de l'estimateur statistique optimal ?
- Quels sont les outils nous permettant de réduire la variance induite par le caractère aléatoire de ces méthodes ?

Considérons maintenant les cas où, par exemple, la cinématique de l'état peut changer de façon brutale (objet manoeuvrant, changement de régime lié à l'environnement en géologie) et les observations dépendent de l'environnement dans lequel se situe le capteur (observation d'une scène sur la mer ou dans le ciel). Une des approches classiques dans un tel cas consiste à introduire un troisième processus latent, dit processus de saut, qui modélise les changements de paramètres cinématiques ou des paramètres de l'environnement. Plus précisément, le modèle le plus souvent utilisé est un système d'état classique dont les paramètres dépendent d'une chaîne de Markov exogène [13] [85]. Ces modèles sont donc hiérarchiques par nature : on se donne d'abord la loi (markovienne) du saut, puis celle du signal caché conditionnellement à ces sauts, et enfin celle des observations conditionnellement aux sauts et au signal caché.

Bien que physiquement justifiée, une telle approche peut, dans certains cas, limiter structurellement l'efficacité des algorithmes d'inférence statistique. En effet, même dans le cas simple où conditionnellement aux variables de saut, la chaîne de Markov cachée est un système d'état classique linéaire et gaussien, le calcul de l'estimateur bayésien optimal (au sens du risque quadratique) est un problème NP-complet. Des techniques de calcul approché (algorithmes pseudo-bayésiens généralisés (*Generalized pseudo-Bayes algorithms*) [118], algorithmes multi-modèles interactifs (*Interacting Multiple Models* = IMM) [13], algorithmes de filtrage particulière (FP) [44]) sont donc indispensables. Le choix de l'approximation utilisée est donc crucial et influence la précision des estimateurs utilisés dans ces modèles probabilistes. Cependant, les limitations structurelles de ces modèles posent le problème de leur pertinence réelle, et l'existence de modèles qui représentent également une certaine réalité physique et dans lesquels il est aisé de calculer exactement des estimateurs bayésiens est une question ouverte.

Une difficulté supplémentaire peut s'ajouter au problème d'estimation précédent. Dans de nombreuses situations (surveillance d'un aéroport, flux vidéo d'images de foule, par exemple), le nombre d'objets devient aléatoire et dépend du temps, de nouveaux objets pouvant apparaître ou disparaître à tout instant. Par ailleurs, les mesures observées aux bornes d'un capteur peuvent provenir des objets pistés (on parlera alors d'objets détectés), ou correspondre à un fouillis parasite auquel cas nous parlerons de mesures de fausses alarmes. De plus, un objet peut ne pas être détecté, auquel cas il n'existe pas d'observation associée à cet objet. L'extension au cas multi-objets du filtrage bayésien optimal mono-cible n'est donc pas immédiate, car au problème d'estimation s'ajoute le problème de l'appariement dynamique entre observations et états cachés.

De nombreuses solutions ont été proposées (algorithme de plus proche voisin (*Global Nearest Neighbour algorithm*) [11], algorithme de filtrage probabiliste joint à association de données (*Joint Probabilistic Data Association* = JPDA) [106], algorithme de pistage par hypothèses multiples (*Multiple Hypothesis Tracking* = MHT) [111] [16] et leurs variantes). Cependant, ces travaux pionniers souffrent de plusieurs limitations majeures : ils ne peuvent être utilisés que pour un nombre d'objets et d'observations limités ; le coût de calcul de l'algorithme JPDA croît exponentiellement avec le nombre d'objets et d'observations ; enfin, le filtre MHT optimal, de par sa nature combinatoire, gère un nombre total d'hypothèses qui croît exponentiellement avec le temps, ce qui le rend inutilisable et nécessite le développement d'algorithmes sous-optimaux.

Une approche plus récente, fondée sur la statistique des ensembles finis aléatoires (*Random Finite Sets* = RFS), consiste à éviter le problème d'appariement objets-observations en considérant qu'à chaque instant, les objets inconnus sont représentés par autant de vecteurs d'état, dont le nombre est lui-même aléatoire et dépendant du temps [56] [81]. La loi de probabilité de cet ensemble fini de vecteurs aléatoires est donc caractérisée par la loi du nombre d'objets, et conditionnellement à ce nombre d'objets, la loi jointe de ces différents objets. Cependant, la complexité algorithmique d'un filtre de Bayes optimal qui propagerait la

« densité multi-objets » a posteriori, nécessite, une nouvelle fois, le recours à des approximations. Parmi celles-ci, le filtre *Probability Hypothesis Density* (PHD) [80] cherche à propager dans le temps un moment de la densité multi-objets a posteriori, qui correspond à la densité spatiale de l'espérance mathématique du nombre d'objets. Le PHD permet donc d'estimer à la fois le nombre d'objets (par intégration de cette densité spatiale) et leur état inconnu (par la recherche de maxima locaux). La propagation du PHD a posteriori fait néanmoins intervenir des intégrales et/ou des sommes dont le nombre de termes croît exponentiellement ; là encore se pose donc un problème de choix d'approximation.

Nous venons de soulever différentes difficultés inhérentes aux problèmes de filtrage mono- et multi objet(s) que nous allons aborder dans cette thèse. Notre travail est donc organisé de la manière suivante :

Le Chapitre 1 introduit les différents points essentiels abordés dans ce travail. Les modèles statistiques classiquement utilisés ainsi que les problématiques de filtrage mono- et multi-objet(s) y sont exposés de façon plus précise que dans cette introduction où n'apparaît aucune notation mathématique. Après avoir précisé les différents points importants, nous présentons en détail la structure de notre travail en trois chapitres.

Le Chapitre 2 est consacré aux méthodes de FP. Plus précisément, notre réflexion porte en premier lieu sur l'emploi des différentes boucles de propagation permettant de calculer la loi de filtrage a posteriori et/ou l'estimateur bayésien optimal. L'implémentation pratique et les performances des différents algorithmes obtenus sont discutées à partir de diverses simulations informatiques. Enfin, certains des algorithmes présentés font également l'objet d'une analyse statistique afin de prévoir leur comportement en fonction du modèle probabiliste utilisé, et d'optimiser leur performance.

Le Chapitre 3 est dédié aux techniques de réduction de variance pour les méthodes SMC, dans le cadre du filtrage mono- et multi-objets. Les méthodes de réduction de variance en FP sont principalement basées sur le théorème de Rao-Blackwell (RB) et ont été abordées dans diverses contributions [23] [44] [24]. Leur objectif est de proposer un estimateur SMC basé sur des tirages de v.a. dont la dimension est plus petite que celle des tirages des algorithmes de FP classiques du Chapitre 2. L'estimation de l'état inconnu combine donc une méthode particulière avec une méthode de calcul exact, conditionnellement à l'approximation particulière. Dans ce chapitre, nous prolongeons les travaux antérieurs en adaptant dans un premier temps ces techniques RB au contexte du filtrage multi-objets, dans lequel non seulement les états sont inconnus mais également le nombre d'objets. D'autre part, nous proposons une utilisation alternative originale du théorème de RB pour les problèmes de filtrage mono- et multi-objets : plutôt que de chercher à réduire la variance des estimateurs d'intérêt en considérant un découpage *spatial* du vecteur d'état inconnu, comme cela est fait habituellement, nous exploitons l'aspect par nature séquentiel des algorithmes SMC (chaque nouveau jeu de données contient le jeu précédent) et nous considérons donc un découpage *temporel* de la séquence inconnue de vecteurs d'états.

Le Chapitre 4 se focalise sur la modélisation probabiliste choisie en amont du problème d'estimation, modélisation qui n'est toujours qu'une représentation approchée d'une réalité physique. Dans un premier temps, nous remettons en cause la représentation classique des modèles de Markov cachés et des modèles de Markov à sauts Markoviens en considérant des modèles plus généraux que constituent les modèles couples et triplets [94] [93]. Dès lors que l'intérêt de ces modèles est justifié, il convient de se demander si les algorithmes d'estimation bayésienne peuvent être utilisés dans ces modèles plus généraux ; en particulier, la formulation habituelle du problème de filtrage multi-objets dans le cadre des RFS ne permet pas de prendre en compte de façon immédiate les propriétés plus générales de ces modèles. À cette fin, nous proposons une reformulation du problème de filtrage multi-objets capable de prendre en compte des modèles statistiques

plus complexes. Enfin, nous réfléchissons à l'utilisation des modèles triplets pour le problème d'estimation bayésienne dans les modèles à sauts linéaires et gaussiens classiquement utilisés. Plutôt que de chercher à affiner le calcul de l'estimateur dans ce modèle donné, il peut être tout aussi pertinent de travailler en amont sur le modèle lui-même, c'est à dire d'inclure, dès sa conception, des propriétés structurelles permettant ensuite le développement d'algorithmes efficaces [95]. En résumé, notre méthode d'approximation alternative approche le modèle originel par un modèle triplet qui partage certaines propriétés physiques pertinentes mais dans lequel l'estimateur bayésien optimal (au sens du risque quadratique) est calculable exactement avec une complexité linéaire en nombre d'observations.

Introduction

Sommaire

1.1 Filtrage statistique mono-objet	20
1.2 Filtrage statistique multi-objets	22
1.3 Contributions de cette thèse	27
1.4 Publications de l’auteur	30

Le problème très général abordé dans cette thèse est le problème du *filtrage statistique*, qui consiste à estimer de façon récursive des données cachées, à partir de données observées. Les applications qui s’appuient sur cette problématique générale sont nombreuses et il convient d’en présenter une liste nécessairement non exhaustive :

- détection et poursuite d’objet(s) mobile(s) [62] [12] [23] [25] ;
- estimation de volatilité en finance [97] [19] ;
- détection de rupture en géologie ou en biologie [47] [21] ;
- détection de sources de signaux à bande étroite en communications numériques [10] ou extraction de signaux dans des canaux bruités [23] ;
- problème de cartographie et localisation simultanées en robotique [45] [87] ;
- problème de prévisions météorologiques [14].

Plus précisément, ces nombreuses applications reposent sur deux problèmes différents de filtrage statistique :

- dans le cadre du filtrage mono-objet, nous souhaitons estimer le vecteur d’état au cours du temps d’un *unique* état caché ; à chaque instant, nous observons *une* réalisation d’une v.a. reliée de façon probabiliste à l’état caché que l’on souhaite estimer ;
- dans le contexte du filtrage multi-objets, le problème est plus compliqué. D’une part, le nombre d’états cachés à estimer est *inconnu* (il peut être supérieur à 1) et évolue dans le temps (des objets apparaissent tandis que d’autres disparaissent) ; d’autre part, les réalisations de v.a. observées dont nous disposons à un instant donné ne sont pas nécessairement dues à la présence d’un état caché et, par ailleurs, certains états cachés peuvent n’être associés à *aucune* observation.

Dans un premier temps, il convient de présenter séparément ces deux problématiques, que nous ramènerons ensuite à un problème commun. Par la suite, certaines des solutions proposées seront donc communes aux filtrages mono- et multi-objet(s).

1.1 Filtrage statistique mono-objet

Nous considérons d'abord le problème du filtrage mono-objet. À chaque instant n , nous souhaitons estimer l'état caché d'un objet, noté $\mathbf{X}_n \in \mathbb{R}^p$, à partir d'observations $\mathbf{Y}_{0:n} = \{\mathbf{Y}_i\}_{i=0}^n$, où $\mathbf{Y}_n \in \mathbb{R}^q$. Parmi les modèles les plus fréquemment utilisés pour relier $\mathbf{X}_{0:n}$ et $\mathbf{Y}_{0:n}$, nous rencontrons les modèles de Markov cachés et modèles de Markov à sauts markoviens que nous définissons ci-après.

1.1.1 Modèles de Markov cachés

$\{\mathbf{X}_n, \mathbf{Y}_n\}_{n \geq 0}$ est une chaîne de Markov cachée (*Hidden Markov Chain* = HMC) si la loi jointe du couple $(\mathbf{X}_{0:n}, \mathbf{Y}_{0:n})$ s'écrit

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\mathbf{x}_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1}) \prod_{i=0}^n g_i(\mathbf{y}_i | \mathbf{x}_i) \quad (1.1.1)$$

(pour des raisons de simplicité de la notation, nous ne distinguerons pas les variables des densités de probabilités et les réalisations de v.a.). Plus précisément, dans le modèle (1.1.1), le processus $\{\mathbf{X}_n\}_{n \geq 0}$ est une chaîne de Markov définie par la densité de \mathbf{X}_0 , $p(\mathbf{x}_0)$, et les transitions $f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1})$. De plus, conditionnellement à tous les états $\mathbf{X}_{0:n} = \mathbf{x}_{0:n}$, les observations $\mathbf{Y}_{0:n}$ sont indépendantes, et l'observation \mathbf{Y}_i dépend seulement de l'état caché \mathbf{X}_i . Ces modèles s'avèrent d'une grande utilité en pratique pour deux raisons. D'une part, les densités $f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1})$ décrivent l'évolution de l'état caché dans le temps tandis que les lois $g_i(\mathbf{y}_i | \mathbf{x}_i)$ mesurent la vraisemblance de l'observation \mathbf{y}_i avec l'état \mathbf{x}_i . Par conséquent, le choix de ces lois est intrinsèque au problème que l'on se donne (poursuite d'un objet mobile, estimation de la volatilité du prix d'une action, etc...) [19]. D'autre part, dans le modèle (1.1.1), il est possible (au moins théoriquement) de calculer récursivement la loi de filtrage $p(\mathbf{x}_n | \mathbf{y}_{0:n})$ (nous pourrions noter, pour des raisons de concision, $p(\mathbf{x}_i | \mathbf{y}_{0:j}) = p_{i|j}$) à partir des lois $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$ et $g_n(\mathbf{y}_n | \mathbf{x}_n)$. Ce calcul se fait en deux étapes, une étape de prédiction et une étape de mise à jour [5] :

$$p(\mathbf{x}_n | \mathbf{y}_{0:n-1}) = \int f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}, \quad (1.1.2)$$

$$p(\mathbf{x}_n | \mathbf{y}_{0:n}) = \frac{g_n(\mathbf{y}_n | \mathbf{x}_n) p(\mathbf{x}_n | \mathbf{y}_{0:n-1})}{\int g_n(\mathbf{y}_n | \mathbf{x}_n) p(\mathbf{x}_n | \mathbf{y}_{0:n-1}) d\mathbf{x}_n}. \quad (1.1.3)$$

La donnée de cette loi permet alors de proposer des estimateurs de $f(\mathbf{X}_n)$, où $f(\cdot)$ est une fonction d'intérêt. Considérons par exemple le critère [15] [103]

$$J(\Theta) = E((\Theta - f(\mathbf{X}_n))^T (\Theta - f(\mathbf{X}_n))), \quad (1.1.4)$$

(qui correspond à l'erreur quadratique moyenne (EQM) $E((\Theta - f(X_n))^2)$ dans le cas scalaire). La minimisation de ce critère, lorsque $\mathbf{y}_{0:n}$ est fixé, nous conduit à calculer l'estimateur

$$\Theta_n = E(f(\mathbf{X}_n) | \mathbf{y}_{0:n}) = \int f(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{y}_{0:n}) d\mathbf{x}_n. \quad (1.1.5)$$

Le calcul exact de $p(\mathbf{x}_n | \mathbf{y}_{0:n})$ ainsi que celui de l'intégrale Θ_n n'est généralement pas faisable, excepté dans le cas particulier important des HMC linéaires et gaussiens,

$$f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n), \quad (1.1.6)$$

$$g_n(\mathbf{y}_n | \mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n), \quad (1.1.7)$$

où le filtre de Kalman (FK) [67] [68] [59] est applicable.

Dans les autres cas, des approximations numériques comme le filtre de Kalman étendu (*Extended Kalman Filter* = EKF) [62] ou le filtre de Kalman sans parfum (*Unscented Kalman Filter* = UKF) [66] [62] approchent les deux premiers moments de la loi $p_n|n$. Le FP (ou méthodes SMC) [127] [79] [97] [43] [18] fournit une alternative aux méthodes d'approximations numériques et est un outil puissant qui permet d'approcher le moment Θ_n par un estimateur $\hat{\Theta}_n$ dont l'expression repose sur des simulations probabilistes. Dans cette classe d'approximations, la loi de filtrage $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ est approchée par une loi discrète notée $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_n^i}$ dont le support est donné par des points $\{\mathbf{x}_n^i\}_{i=1}^N$ tirés aléatoirement et dont les poids $\{w_n^i\}_{i=1}^N$ sont calculés à partir des paramètres du modèle (1.1.1), des $\{\mathbf{x}_n^i\}_{i=1}^N$ et des observations $\mathbf{y}_{0:n}$. L'estimateur $\hat{\Theta}_n$ de Θ_n est alors donné par $\hat{\Theta}_n = \sum_{i=1}^N w_n^i f(\mathbf{x}_n^i)$.

1.1.2 Modèles de Markov cachés à sauts markoviens

Dans certaines applications, il peut être utile de considérer plusieurs modèles HMC afin de modéliser différents régimes possibles, que ce soit pour la dynamique de l'état caché ou pour le lien entre les observations et les états cachés [2] [4] [118] [69] [23] [47] [20]. C'est ainsi que les modèles de Markov cachés à sauts markoviens (*Jump Markov State Space System* = JMSS) permettent de prendre en compte plusieurs modèles HMC en utilisant un troisième processus de Markov discret $R_n \in \{1, \dots, K\}$ appelé « mode » ou « régime ». Nous faisons donc dépendre un modèle HMC (1.1.1) d'un processus discret $\mathbf{R}_{0:n}$ permettant de modéliser les différents régimes possibles associés à l'état caché \mathbf{X}_n . À titre d'exemple, dans le cas d'un objet manoeuvrant dont on chercherait à estimer les états cachés, il peut être intéressant de considérer à chaque instant trois modèles différents afin de prendre en compte les différents régimes possibles de l'objet caché : trajectoire rectiligne, manoeuvre à gauche et manoeuvre à droite. La loi du triplet $(\mathbf{X}_{0:n}, \mathbf{Y}_{0:n}, \mathbf{R}_{0:n})$ s'écrit donc

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n}) = p(r_0) \prod_{i=1}^n p(r_i|r_{i-1}) p(\mathbf{x}_0|r_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}, r_i) \prod_{i=0}^n g_i(\mathbf{y}_i|\mathbf{x}_i, r_i). \quad (1.1.8)$$

En d'autres termes $\{R_n\}_{n \geq 0}$ est une chaîne de Markov, et conditionnellement à $\mathbf{R}_{0:n} = \mathbf{r}_{0:n}$, $(\mathbf{X}_{0:n}, \mathbf{Y}_{0:n})$ est une chaîne de Markov cachée décrite par les transitions $f_{i|i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}, r_i)$ et les vraisemblances $g_i(\mathbf{y}_i|\mathbf{x}_i, r_i)$. Comme dans les modèles HMC, le choix de ces lois dépend du problème considéré et caractérise ce que nous appellerons par la suite « les propriétés physiques du modèle ».

Dans ces modèles, les équations de filtrage s'écrivent

$$p(\mathbf{x}_n, r_n|\mathbf{y}_{0:n-1}) = \sum_{r_{n-1}} \int p(r_n|r_{n-1}) f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}, r_n) p(\mathbf{x}_{n-1}, r_{n-1}|\mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}, \quad (1.1.9)$$

$$p(\mathbf{x}_n, r_n|\mathbf{y}_{0:n}) = \frac{g_n(\mathbf{y}_n|\mathbf{x}_n, r_n) p(\mathbf{x}_n, r_n|\mathbf{y}_{0:n-1})}{\sum_{r_n} \int g_n(\mathbf{y}_n|\mathbf{x}_n, r_n) p(\mathbf{x}_n, r_n|\mathbf{y}_{0:n-1}) d\mathbf{x}_n}, \quad (1.1.10)$$

d'où nous déduisons

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{r_n} p(\mathbf{x}_n, r_n|\mathbf{y}_{0:n}), \quad (1.1.11)$$

$$\Theta_n = \int f(\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n}) d\mathbf{x}_n = \sum_{r_n} \int f(\mathbf{x}_n) p(\mathbf{x}_n, r_n|\mathbf{y}_{0:n}) d\mathbf{x}_n. \quad (1.1.12)$$

Il reste à calculer (1.1.12). Dans les modèles à sauts linéaires et gaussiens, qui vérifient

$$f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}, r_n) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n(r_n)\mathbf{x}_{n-1}; \mathbf{Q}_n(r_n)) \text{ et} \quad (1.1.13)$$

$$g_n(\mathbf{y}_n|\mathbf{x}_n, r_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n(r_n)\mathbf{x}_n; \mathbf{R}_n(r_n)), \quad (1.1.14)$$

la loi $p(\mathbf{x}_n|\mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ est calculable. En effet, à sauts connus, ce modèle devient le modèle (1.1.6)-(1.1.7) dans lequel le FK peut être utilisé. Cependant, le calcul de Θ_n dans (1.1.12) est un problème NP-complet [118] à cause des sommes qui interviennent dans (1.1.13)-(1.1.14).

Par conséquent, qu'il s'agisse de modèles à sauts linéaires et gaussiens, dans lesquels le calcul de Θ_n est un problème NP-complet, ou de modèles à sauts plus généraux, dans lesquels le calcul de Θ_n fait en plus intervenir des intégrales qui ne sont pas calculables, des solutions sous-optimales sont nécessaires. Plusieurs approximations ont été proposées pour l'estimation de Θ_n dans ces modèles et nous les détaillerons tout au long de ce travail. Il est en particulier possible de distinguer deux classes d'approximations. D'une part, des approximations numériques qui propagent un mélange de gaussiennes dont les paramètres sont calculés par le FK ou l'EKF/UKF ont été proposées [118] [85]. D'autre part, les techniques de FP ont été adaptées pour ce problème et peuvent être divisées en deux classes [4] [44] :

- dans les modèles (1.1.8), le couple $((\mathbf{X}_n, \mathbf{R}_n), \mathbf{Y}_n)$ est une chaîne de Markov cachée (1.1.1) caractérisée par les transitions $p(r_i|r_{i-1})f_{i|i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{r}_i)$ et les vraisemblances $g_i(\mathbf{y}_i|\mathbf{x}_i, \mathbf{r}_i)$; par conséquent, les méthodes MC utilisées dans les modèles HMC restent valables pour les modèles à sauts, à l'adaptation près $\mathbf{X}_n \leftarrow (\mathbf{X}_n, \mathbf{R}_n)$;
- dans la seconde classe de méthodes, nous réécrivons

$$\Theta_n = \sum_{\mathbf{r}_{0:n}} \underbrace{p(\mathbf{r}_{0:n}|\mathbf{y}_{0:n})}_{\text{FP}} \int f(\mathbf{x}_n) \underbrace{p(\mathbf{x}_n|\mathbf{r}_{0:n}, \mathbf{y}_{0:n})}_{\text{FK}} d\mathbf{x}_n. \quad (1.1.15)$$

L'équation (1.1.15) suggère d'estimer Θ_n en deux temps. D'abord, la loi $p(\mathbf{r}_{0:n}|\mathbf{y}_{0:n})$ est approchée par une méthode MC. Ensuite, conditionnellement à cette approximation particulière, l'intégrale $\int f(\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{r}_{0:n}, \mathbf{y}_{0:n})d\mathbf{x}_n$ est calculée exactement. Par conséquent, contrairement à la classe précédente, l'approximation MC ne concerne que la loi $p(\mathbf{r}_{0:n}|\mathbf{y}_{0:n})$ et repose donc sur des tirages probabilistes de dimension plus petite. Sous certaines conditions sur les lois utilisées pour tirer les particules, l'estimateur qui en résulte a une variance asymptotique (lorsque le nombre de tirages MC tend vers l'infini) plus petite que l'estimateur résultant de la première classe d'approximations MC [24] [44], mais la contrepartie est que cet estimateur ne peut être calculé que si l'intégrale dans (1.1.15) est calculable. L'utilisation de tirages de dimension plus petite est connue sous le nom de FP « marginalisés » ou « Rao-Blackwellisés » (dont il sera question avec plus de détails au Chapitre 3).

1.2 Filtrage statistique multi-objets

Le calcul ou l'approximation de Θ_n dans les modèles (1.1.1) ou (1.1.8) permet d'estimer l'état caché d'un unique objet. Lorsque l'on cherche à estimer les états cachés de plusieurs objets, le problème est plus complexe et passe par une formulation statistique adaptée que nous présentons à présent. Le Paragraphe 1.2.1 décrit les contraintes inhérentes à cette problématique plus générale, puis, dans le Paragraphe 1.2.2, nous présentons brièvement les principales méthodes de filtrage multi-objets.

1.2.1 Problèmes et contraintes

Contrairement à la section précédente, il ne s'agit plus ici d'estimer une séquence d'états d'un unique objet, mais d'estimer les séquences d'états d'objets dont le nombre est inconnu et évolue dans le temps : des objets peuvent apparaître (naissance) ou disparaître (mort) au cours du temps. D'autre part, contrairement au problème précédent, il se peut qu'un objet ne soit pas détecté, c'est à dire que nous n'ayons pas accès à une réalisation de la v.a. \mathbf{Y}_n associée à un état caché \mathbf{X}_n . Enfin, les observations dont nous disposons ne sont pas nécessairement associées à un état caché : de telles observations seront appelées « fausses alarmes » ou « fouillis ».

Les solutions les plus anciennes pour traiter ce problème reposent sur les techniques de filtrage mono-objet combinées à un mécanisme d'association objets-observations visant à optimiser l'association entre un état caché et une observation ; ces solutions incluent les algorithmes JPDA [11] [106] et MHT [111] [16]. Ces solutions ne seront pas discutées dans ce travail puisque nous allons nous pencher sur une classe de solutions plus récentes basées sur la théorie des Ensembles Statistiques Finis (ou RFS) [56] [54] [80]. Ces solutions considèrent les objets cachés ainsi que les observations comme des RFS, c'est à dire des ensembles de variables aléatoires dont le cardinal est également aléatoire et varie dans le temps. Le paragraphe suivant formalise le problème de filtrage multi-objets sous l'angle de la théorie des RFS et décrit les solutions associées.

1.2.2 Formulation statistique du problème de filtrage multi-objets

Ensembles statistiques finis

A l'instant n , nous modélisons les objets cachés ainsi que les observations par deux RFS,

$$\Xi_n = \{\mathbf{X}_{1,n}, \dots, \mathbf{X}_{N(n),n}\}, \quad (1.2.1)$$

$$\Psi_n = \{\mathbf{Z}_{1,n}, \dots, \mathbf{Z}_{M(n),n}\}, \quad (1.2.2)$$

où $N(n)$ et $M(n)$ représentent respectivement le nombre (aléatoire) d'objets et d'observations à l'instant n . Une réalisation du RFS Ξ_n est donc la réalisation du nombre d'objets de ce RFS, et conditionnellement à ce nombre, une réalisation des v.a. qui le composent. Nous la noterons $\Xi_n = \mathcal{X}_n = \{\mathbf{x}_{1,n}, \dots, \mathbf{x}_{m,n}\}$ (m représente ici une réalisation de $N(n)$). De même, une réalisation du RFS Ψ_n sera notée \mathcal{Z}_n . Il convient ici de justifier l'emploi d'une nouvelle notation $\mathbf{Z}_{i,n}$, pour modéliser la i -ème observation. Dans la section précédente, les observations étaient notées \mathbf{Y}_n ; mais, dans le problème de filtrage multi-objets, rappelons qu'une observation peut être associée à un état caché mais aussi être une mesure de fausse alarme. La notation \mathbf{Z} sera donc utilisée lorsque nous ne présumerons pas de la nature de l'observation (fausse alarme ou détection), tandis que la notation \mathbf{Y} fera toujours allusion à une observation aléatoire associée à un état caché \mathbf{X} .

La modélisation (1.2.1)-(1.2.2) a l'avantage de ramener le problème de filtrage multi-objets à une formulation très simple : estimer à chaque instant n le RFS Ξ_n (c'est à dire estimer le nombre d'éléments de ce RFS ainsi que les valeurs prises par les v.a. qui le composent) à partir de l'historique des réalisations de $\Psi_n = \mathcal{Z}_n$, noté $\mathcal{Z}_{0:n}$. Nous avons besoin, pour cela, de généraliser la notion de densité de probabilité et d'établir un lien entre les RFS Ξ_n et Ξ_{n-1} puis entre Ξ_n et Ψ_n .

Conditionnellement à une réalisation du RFS $\Xi_{n-1} = \mathcal{X}_{n-1}$, le RFS Ξ_n est composé des objets de \mathcal{X}_{n-1}

ayant survécu ainsi que des objets naissants ; nous pouvons donc écrire

$$\Xi_n = \left[\bigcup_{\mathbf{x}_{n-1} \in \mathcal{X}_{n-1}} \mathbf{S}_{n|n-1}(\mathbf{x}_{n-1}) \right] \cup \left[\bigcup_{\mathbf{x}_{n-1} \in \mathcal{X}_{n-1}} \mathbf{B}_{n|n-1}(\mathbf{x}_{n-1}) \right] \cup \Gamma_n, \quad (1.2.3)$$

où $\mathbf{S}_{n|n-1}(\mathbf{x}_{n-1})$ modélise le comportement de l'état \mathbf{x}_{n-1} entre les instants $n-1$ et n ($\mathbf{S}_{n|n-1}(\mathbf{x}_{n-1}) = \{\mathbf{X}_n\}$ si l'état \mathbf{x}_{n-1} survit, ou $\mathbf{S}_{n|n-1}(\mathbf{x}_{n-1}) = \emptyset$ si l'état \mathbf{x}_{n-1} ne survit pas) et où $\mathbf{B}_{n|n-1}(\mathbf{x}_{n-1})$ est le RFS des objets issus de la séparation de l'état \mathbf{x}_{n-1} (dans la suite, pour des enjeux de concision des équations, nous ne considérerons pas ce phénomène dit de « spawning », c'est à dire que nous prendrons $\mathbf{B}_{n|n-1}(\mathbf{x}_{n-1}) = \emptyset$, quel que soit \mathbf{x}_{n-1}) ; enfin, Γ_n désigne l'ensemble des objets naissants à l'instant n .

De même, conditionnellement à une réalisation $\Xi_n = \mathcal{X}_n$, le RFS Ψ_n s'écrit

$$\Psi_n = \mathbf{K}_n \cup \left[\bigcup_{\mathbf{x}_n \in \mathcal{X}_n} \Omega(\mathbf{x}_n) \right], \quad (1.2.4)$$

où \mathbf{K}_n désigne l'ensemble des mesures de fausses alarmes et $\bigcup_{\mathbf{x}_n \in \mathcal{X}_n} \Omega(\mathbf{x}_n)$ l'ensemble des mesures associées aux objets de \mathcal{X}_n détectés. Nous supposons dans un premier temps que le modèle suivi par un objet et l'observation qui lui est associée (dont, rappelons le, on peut ne pas observer une réalisation) suit le modèle HMC (1.1.1).

Densité multi-objets

Maintenant que le lien entre les différents RFS a été établi, introduisons quelques outils nécessaires pour caractériser la loi d'un RFS. En particulier, la densité multi-objets $p(\mathcal{X})$ d'un RFS Ξ [54] contenant des v.a. à valeurs dans \mathbb{R}^p généralise la notion de densité de probabilité ; c'est une fonction positive de \mathcal{X} (où $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ désigne une réalisation de Ξ) à valeurs dans \mathbb{R} qui peut se réécrire

$$p(\mathcal{X}) = m! \Pr(|\Xi| = m) p(\mathbf{x}_1, \dots, \mathbf{x}_m), \quad (1.2.5)$$

où $p(\mathbf{x}_1, \dots, \mathbf{x}_m)$ désigne une densité de probabilité reliée à $(\mathbf{X}_1, \dots, \mathbf{X}_m)$ (voir par exemple [81, p. 364]). Le terme $m!$ exprime le fait que $p(\cdot)$ est symétrique par rapport à ses arguments. L'intégrale d'ensemble de la densité multi-objets $p(\mathcal{X})$ (qui sera caractérisée par la notation $\delta\mathcal{X}$) est définie par

$$\int p(\mathcal{X}) \delta\mathcal{X} \triangleq p(\emptyset) + \frac{1}{i!} \sum_{i=1}^{+\infty} \int p(\{\mathbf{x}_1, \dots, \mathbf{x}_i\}) d\mathbf{x}_{1:i}, \quad (1.2.6)$$

et est égale à 1.

De la même manière que le filtre bayésien (1.1.2)-(1.1.3) propage la densité de la variable aléatoire \mathbf{X}_n conditionnellement à $\mathbf{Y}_{0:n} = \mathbf{y}_{0:n}$, le filtre bayésien multi-objets propage la densité multi-objets du RFS Ξ_n conditionnellement à $\Psi_{0:n} = \mathcal{Z}_{0:n}$ [54] :

$$p(\mathcal{X}_n | \mathcal{Z}_{0:n-1}) = \int \bar{f}_{n|n-1}(\mathcal{X}_n | \mathcal{X}_{n-1}) p(\mathcal{X}_{n-1} | \mathcal{Z}_{0:n-1}) \delta\mathcal{X}_{n-1}, \quad (1.2.7)$$

$$p(\mathcal{X}_n | \mathcal{Z}_{0:n}) = \frac{\bar{g}_n(\mathcal{Z}_n | \mathcal{X}_n) p(\mathcal{X}_n | \mathcal{Z}_{0:n-1})}{\int \bar{g}_n(\mathcal{Z}_n | \mathcal{X}_n) p(\mathcal{X}_n | \mathcal{Z}_{0:n-1}) \delta\mathcal{X}_n}, \quad (1.2.8)$$

où $\bar{f}_{n|n-1}(\mathcal{X}_n | \mathcal{X}_{n-1})$ et $\bar{g}_n(\mathcal{Z}_n | \mathcal{X}_n)$ désignent les transitions et vraisemblances multi-objets. Ces fonctions dépendent des transitions $f_{n|n-1}$ et vraisemblances g_n qui caractérisent l'évolution dynamique de chaque objet ainsi que la manière dont l'observation intrinsèque à chaque état est produite [54] [56] [81].

Cependant les intégrales d'ensembles (1.2.7)-(1.2.8) ne peuvent pas être calculées en général. Une approximation de ces équations par une méthode SMC [122] n'est envisageable que si le nombre d'objets à estimer est faible, auquel cas les intégrales d'ensemble peuvent être aisément approximées car la somme (à priori infinie) de (1.2.6) se réduit à la somme d'un petit nombre de termes. Dans le cas général, il faut utiliser d'autres approximations ; le filtre PHD que nous définissons maintenant [80] [81] est l'une d'entre elles.

Le PHD

Plutôt que de propager la densité de filtrage multi-objets $p(\mathcal{X}_n | \mathcal{Z}_{0:n})$, Mahler [80] a proposé de propager un moment d'ordre un de cette densité multi-objets, le PHD. Le PHD, ou intensité, $v_n(\mathbf{x})$ du RFS Ξ_n (ou de $p(\mathcal{X}_n)$), conditionnellement à $\mathcal{Z}_{0:n}$, est une fonction de \mathbb{R}^p à valeurs dans \mathbb{R}^+ qui vérifie la propriété suivante : pour tout $S \in \mathbb{R}^p$,

$$\int_S v_n(\mathbf{x}) d\mathbf{x} = \int |S \cap \mathcal{X}_n| p(\mathcal{X}_n | \mathcal{Z}_{0:n}) \delta \mathcal{X}_n = E(|S \cap \mathcal{X}_n| | \mathcal{Z}_{0:n}), \quad (1.2.9)$$

où $|S \cap \mathcal{X}_n|$ désigne le nombre d'éléments de \mathcal{X}_n appartenant à S . En d'autres termes, $v_n(\mathbf{x})$ est la densité spatiale de l'espérance du nombre d'objets. Ainsi, si l'on connaît $v_n(\mathbf{x})$, il est possible d'estimer le nombre d'objets en calculant $\int v_n(\mathbf{x}) d\mathbf{x}$ ainsi que les états cachés de chaque objet en recherchant les maxima locaux de la fonction $v_n(\mathbf{x})$. En définitive, le problème de filtrage bayésien multi-objets peut être approximé par le problème de la propagation du moment d'ordre un $v_n(\mathbf{x})$ de la densité multi-objets $p(\mathcal{X}_n | \mathcal{Z}_{0:n})$.

Il reste à calculer $v_n(\mathbf{x})$. Notons que $v_n(\mathbf{x})$ n'est plus fonction du RFS \mathcal{X}_n mais de $\mathbf{x} \in \mathbb{R}^p$, où \mathbb{R}^p est l'ensemble dans lequel les v.a. qui constituent \mathcal{X} prennent leur valeur. Sous certaines hypothèses, il se trouve que $v_n(\mathbf{x})$ est calculable récursivement. Notons $p_{s,n}(\mathbf{x})$ la probabilité de survie d'un état \mathbf{x} entre les instants $n-1$ et n ; $p_{d,n}(\mathbf{x})$ la probabilité de détection d'un état \mathbf{x} à l'instant n ; $\gamma_n(\mathbf{x})$ l'intensité du RFS des naissances à l'instant n ; $\kappa_n(\mathbf{z})$ l'intensité du RFS des mesures de fausses alarmes. Sous les hypothèses suivantes

A.1 Les objets évoluent et génèrent des observations indépendamment les uns des autres ;

A.2 le RFS régissant les mesures de fausses alarmes est de Poisson (voir Remarque 1.1 ci-dessous) et ces mesures sont indépendantes des mesures dues aux objets ;

A.3 conditionnellement à $\mathcal{Z}_{0:n-1}$, le RFS Ξ_n est de Poisson (voir Remarque 1.1 ci-dessous),

le calcul récursif du PHD se fait en deux étapes et dépend du PHD de la densité multi-objets $p(\mathcal{X}_n | \mathcal{Z}_{0:n-1})$, noté $v_{n|n-1}(\mathbf{x})$:

$$v_{n|n-1}(\mathbf{x}) = \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x} | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1} + \gamma_n(\mathbf{x}), \quad (1.2.10)$$

$$v_n(\mathbf{x}) = (1 - p_{d,n}(\mathbf{x})) v_{n|n-1}(\mathbf{x}) + \sum_{\mathbf{z} \in \mathcal{Z}_n} \frac{p_{d,n}(\mathbf{x}) g_n(\mathbf{z} | \mathbf{x}) v_{n|n-1}(\mathbf{x})}{\kappa_n(\mathbf{z}) + \int p_{d,n}(\mathbf{x}) g_n(\mathbf{z} | \mathbf{x}) v_{n|n-1}(\mathbf{x}) d\mathbf{x}}. \quad (1.2.11)$$

Remarque 1.1. Un RFS Ξ est dit de Poisson si sa densité multi-objets s'écrit, pour $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$,

$$p(\mathcal{X}) = e^{-\lambda} \lambda^m p(\mathbf{x}_1) \cdots p(\mathbf{x}_m), \quad (1.2.12)$$

où $p(\mathbf{x}_i)$ est une densité de probabilité. La loi de probabilité du nombre d'objets est alors donnée par $\Pr(|\Xi| = m) = \frac{e^{-\lambda} \lambda^m}{m!}$ [81], qui est donc une loi de Poisson de paramètre λ . De plus, pour ces RFS, le PHD (ou moment d'ordre 1) s'écrit $v(\mathbf{x}) = \lambda p(\mathbf{x})$. Nous avons donc $\int v(\mathbf{x}) d\mathbf{x} = \lambda$ et nous retrouvons le fait que l'intégrale du PHD est égale à l'espérance du nombre d'objets dans Ξ . Il est possible de supprimer

les hypothèses de Poisson **A.2** et **A.3**, auquel cas les équations (1.2.10)-(1.2.11) deviennent plus complexes puisqu'il faut, en plus, propager la loi de probabilité du nombre d'objets dans Ξ_n conditionnellement à $\mathcal{Z}_{0:n}$. Le filtre résultant est appelé filtre PHD cardinalisé (*Cardinalized PHD* = CPHD) [82] et sera utilisé en Annexe D.2.

En pratique, bien que moins complexes que les équations (1.2.7)-(1.2.8), les équations (1.2.10)-(1.2.11) ne peuvent pas être calculées de façon exacte. En effet, même dans le cas où les intégrales sont calculables, $v_n(\mathbf{x})$ est un mélange qui croît exponentiellement, à cause de la somme $\sum_{\mathbf{z} \in \mathcal{Z}_n}$ dans (1.2.11).

Par conséquent, comme en filtrage mono-objet, et même si une formule récursive exacte est disponible (cf. (1.2.10)-(1.2.11)), l'utilisation de techniques d'approximation est indispensable pour pouvoir approcher le PHD $v_n(\mathbf{x})$ et donc estimer le nombre d'objets ainsi que leur état caché. Ces approximations peuvent également être divisées en deux classes. Nous distinguons dans un premier temps les approximations par mélange de gaussiennes [121] ; $v_n(\mathbf{x})$ est approché par un mélange gaussien dont les paramètres sont déduits du FK ou de l'EKF/UKF. D'autre part, les approximations basées sur l'utilisation de tirages MC peuvent également être adaptées au contexte multi-objets [122] [26]. Le PHD $v_n(\mathbf{x})$ est approché par une densité discrète dont le support est donné par des points $\{\mathbf{x}_n^i\}_{i=1}^{L_n}$, pondérés par $\{w_n^i\}_{i=1}^{L_n}$ (dans ce contexte où le nombre d'objets évolue, la taille du support est amenée à évoluer avec le temps, d'où la notation L_n). La somme de ces poids, généralement différente de 1, fournit alors un estimateur MC du nombre d'objets. En revanche, l'approximation discrète rend plus difficile la recherche de maxima locaux et donc l'extraction des états cachés ; des techniques de clustering peuvent néanmoins être utilisées dans ce cas.

Remarque 1.2. Le problème général traité dans cette thèse est celui du *filtrage* multi-objets qu'il ne faut pas confondre avec le problème de *pistage* multi-objets. Dans le premier cas, nous nous focalisons sur l'estimation du nombre d'objets et de leur état caché, tandis que dans le second cas nous cherchons en plus à différencier l'historique des trajectoires de chaque objet. Ce problème sera brièvement abordé dans l'Annexe D.2 qui expliquera comment il est possible d'obtenir un algorithme de pistage à partir de la propagation du PHD.

1.2.3 Critère de performance

L'évaluation de la performance d'un algorithme de filtrage multi-objets à partir d'une vérité terrain est un problème plus difficile qu'en filtrage mono-objet dans lequel nous calculons le critère (1.1.4) pour un estimateur $\hat{\Theta}_n$ donné. Dans le contexte multi-objets, la performance des algorithmes de filtrage repose sur leur capacité à évaluer le nombre d'objets ainsi qu'à estimer l'état caché de chaque objet. Ces deux critères peuvent être tous deux pris en compte en utilisant une distance spécifique pour ce problème, la distance OSPA (Optimal SubPattern Analysis) [113]. Soient $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ l'ensemble vérité terrain et $\hat{\mathcal{X}} = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{\hat{m}}\}$ un estimateur de cet ensemble. Pour $1 \leq \bar{p} < +\infty$ et $\bar{c} > 0$, nous notons $d^{(\bar{c})}(\mathbf{x}, \hat{\mathbf{x}}) = \min(\bar{c}, \|\mathbf{x} - \hat{\mathbf{x}}\|)$ ($\|\cdot\|$ est la norme euclidienne), Π_m l'ensemble des permutations sur $\{1, 2, \dots, m\}$ et $\pi(i)$ la i -ème permutation de cet ensemble. La distance OSPA entre \mathcal{X} et $\hat{\mathcal{X}}$ est alors définie par

$$\bar{d}_{\bar{p}}^{\bar{c}}(\mathcal{X}, \hat{\mathcal{X}}) \triangleq \left(\frac{1}{\hat{m}} \left(\min_{\pi \in \Pi_{\hat{m}}} \sum_{i=1}^m d^{(\bar{c})}(\mathbf{x}_i, \hat{\mathbf{x}}_{\pi(i)})^{\bar{p}} + \bar{c}^{\bar{p}}(\hat{m} - m) \right) \right)^{1/\bar{p}} \quad (1.2.13)$$

si $m \leq \hat{m}$, et $\bar{d}_{\bar{p}}^{\bar{c}}(\mathcal{X}, \hat{\mathcal{X}}) \triangleq \bar{d}_{\bar{p}}^{\bar{c}}(\hat{\mathcal{X}}, \mathcal{X})$ si $\hat{m} < m$. Il est possible de montrer que $\bar{d}_{\bar{p}}^{\bar{c}}$ définit bien une distance [113]. Le premier terme de la somme représente l'erreur en localisation tandis que le second terme représente

l'erreur en cardinalité. Notons que cette distance fait intervenir deux paramètres \bar{p} et \bar{c} . Le paramètre \bar{c} influe sur le poids de la pénalité $\bar{c}^{\bar{p}}(\hat{m} - m)$ assignée aux erreurs de cardinalité par rapport à $d^{(\bar{c})}(\mathbf{x}_i, \hat{\mathbf{x}}_{\pi(i)})^{\bar{p}}$, terme associé aux erreurs de localisation tandis que pour \bar{c} fixé, le paramètre \bar{p} a tendance à pénaliser les objets de l'ensemble estimé trop éloignés des objets de l'ensemble vérité terrain. Lorsque des analyses de performances seront menées sur des algorithmes de filtrage multi-objets, nous utiliserons donc cette distance OSPA.

1.3 Contributions de cette thèse

Les Paragraphes 1.1 et 1.2 laissent entrevoir une méthodologie commune pour aborder les problèmes de filtrage mono- et multi-objet(s), composée de trois étapes :

1. on commence par se donner un modèle statistique (soit (1.1.1), soit (1.1.8)) vérifié par chaque objet \mathbf{X} et l'observation associée \mathbf{Y} ;
2. pour le modèle choisi, le problème d'estimation du vecteur d'état de chaque objet caché, et éventuellement du nombre d'objets cachés, consiste essentiellement à propager, dans le cas mono-objet, la densité de probabilité de filtrage $p_{n|n}(\mathbf{x})$ (ou un moment suivant cette densité), et dans le cas multi-objets le PHD a posteriori $v_n(\mathbf{x})$. L'équation récursive propageant $p_{n|n}(\mathbf{x})$ ou $v_n(\mathbf{x})$ dépend du modèle considéré au point 1 ;
3. si la densité impliquée dans le problème d'estimation ne peut être calculée exactement, on se donne une méthode d'approximation. En particulier, les méthodes MC basées sur des tirages probabilistes peuvent être appliquées à des modèles très généraux et ont été validées théoriquement par des analyses asymptotiques [32] [24] [26] [64].

Cette démarche soulève par ailleurs les questions suivantes :

1. Dans un premier temps, à modèle choisi et problème donné, on peut se demander, parmi les différents algorithmes de filtrage SMC existant, quelles sont les solutions qu'il est préférable d'utiliser, c'est à dire les solutions pour lesquelles la variance conditionnelle de l'estimateur approché $\hat{\Theta}_n = \sum_{i=1}^N w_n^i f(\mathbf{x}_n^i)$ est la plus petite ;
2. dans le cas où nous nous intéressons en définitive à $\hat{\Theta}_n$ calculé à partir d'une approximation de SMC $\hat{p}_{n|n}$ de $p_{n|n}$ il est naturel de se demander si les techniques de réduction de variance sont applicables, ou non, dans le modèle considéré, et si la réduction de variance peut se faire avec un coût de calcul raisonnable ;
3. enfin, quelle que soit la méthode d'estimation choisie, celle ci ne pourra donner de bons résultats que si les données traitées suivent effectivement le modèle statistique que l'on s'est donné en amont de notre problème. Par conséquent, il est légitime de se demander si le modèle statistique utilisé est pertinent, ou si d'autres modèles ne pourraient pas être mieux adaptés. De plus, est ce que la considération de modèles statistiques différents de ceux utilisés habituellement ne pourrait pas faire émerger des techniques d'estimation alternatives à celles utilisées dans les modèles classiques ?

Notre travail est découpé en trois chapitres distincts qui visent à apporter quelques éléments de réponse aux questions très générales ci-dessus.

Chapitre 2 - Ce chapitre est consacré à l'étude générale des méthodes SMC, dans le cadre du filtrage statistique mono-objet. Nous nous intéressons aux différentes solutions qui permettent de propager une approximation $\hat{p}_{n|n}$ de la loi de filtrage $p_{n|n}$ ou de calculer récursivement un estimateur $\hat{\Theta}_n$ de Θ_n . Parmi ces solutions, nous nous intéressons en particulier aux solutions dites « localement optimales » au sens où les tirages probabilistes utilisés minimisent la variance conditionnelle des poids associés. Nous proposons une analyse statistique de ces solutions dans le but de prévoir leur comportement en fonction du modèle considéré.

Le Chapitre 2 est donc organisé de la manière suivante :

- les Sections 2.1 et 2.2 sont dédiées à la présentation des algorithmes permettant de propager récursivement un estimateur MC $\hat{\Theta}_n$ de Θ_n . Plutôt que d'utiliser directement le filtre bayésien (1.1.2)-(1.1.3) duquel sont déduits les algorithmes SMC courants, nous proposons des chemins bayésiens alternatifs, dont le but est toujours de calculer la loi de filtrage $p_{n|n}$, mais dont la boucle de propagation porte sur la densité de prédiction $p_{n|n-1}$ ou de lissage $p_{n-1|n}$ à un pas ;
- la Section 2.3 est consacrée à la comparaison par simulations des algorithmes proposés, ainsi qu'à leur éventuelle approximation lorsque les fonctions nécessaires à leur implémentation ne sont pas calculables ;
- la Section 2.4 se focalise sur une analyse statistique non asymptotique des estimateurs issus des algorithmes localement optimaux ; cette analyse nous conduit à proposer des estimateurs $\hat{\Theta}_n$ plus performants que ceux utilisés habituellement.

Chapitre 3 - Nous nous intéressons au problème de réduction de variance des estimateurs ponctuels construits à partir des approximations SMC du chapitre précédent. Les solutions proposées reposent sur le théorème de RB qui stipule que pour deux v.a. X_1 et X_2 ,

$$\text{var}(E(f(X_2)|X_1)) = \text{var}(f(X_2)) - E(\text{var}(f(X_2)|X_1)). \quad (1.3.1)$$

Par conséquent, les v.a. $E(f(X_2)|X_1)$ et $f(X_2)$ ont même espérance mais la variance de $E(f(X_2)|X_1)$ est inférieure ou égale à celle de $f(X_2)$. Ce résultat a été utilisé en filtrage mono-objet, afin de palier aux mauvaises performances de $\hat{\Theta}_n$ lorsque la dimension de l'état caché est grande. Plus précisément, une application de (1.3.1) concerne les modèles HMC dans lesquels le vecteur d'état multi-dimensionnel $\mathbf{X}_n = (\mathbf{X}_n^{nl}, \mathbf{X}_n^l)$ peut se décomposer en une composante non-linéaire et une composante linéaire ; le FP n'agit alors que sur la composante non-linéaire \mathbf{X}_n^{nl} tandis qu'un estimateur de la composante linéaire est calculé exactement, conditionnellement à l'approximation particulière. Dans les modèles à sauts (1.1.8), ce résultat a été utilisé pour construire la seconde classe d'algorithmes de FP qui repose sur (1.1.15).

Notre contribution de ce domaine est la suivante :

- d'une part, les techniques de réduction de variance ont été appliquées en filtrage mono-objet, mais ont été peu exploitées en filtrage multi-objets. Nous aborderons donc le problème dans ce contexte multi-objets, problème rendu d'autant plus nécessaire qu'il convient d'estimer l'état de chaque objet, mais aussi le nombre d'objets ;
- jusqu'ici, les techniques de réduction de variance ont été appliquées au cas où il est possible d'effectuer une décomposition spatiale du vecteur d'état (cf. [44] [23] [24] [76] et item ci-dessus). Nous proposons une nouvelle application du théorème de RB, appliqué cette fois à une décomposition temporelle de $\mathbf{X}_{0:n}$. En effet, nous partons du constat que les algorithmes de MC *séquentiels* fournissent par nature une variable conditionnante \mathbf{X}_1 , puisque $p_{0:n|n}$ est construit à partir de $p_{0:n-1|n-1}$. Nous exploitons cette idée dans les contextes mono- et multi-objet(s).

Le Chapitre 3 est donc organisé de la manière suivante.

- Nous rappelons, en introduction du chapitre, les modèles statistiques dans lesquels les techniques de Rao-Blackwellisation ou méthodes MC conditionnelles (*Conditional MC = CMC*) sont classiquement utilisées ;
- Dans la Section 3.2, nous cherchons à améliorer les estimateurs du nombre d’objets et des états cachés de chaque objet en étendant les techniques de FP marginalisé à la propagation du PHD. Ces techniques sont également combinées à des adaptations de méthodes SMC efficaces étudiées au Chapitre 2 ;
- dans la Section 3.3, nous exploitons la nature séquentielle des algorithmes de FP afin d’appliquer le résultat (1.3.1) à une partition temporelle de la séquence $\mathbf{X}_{0:n} = (\mathbf{X}_{0:n-1}, \mathbf{X}_n)$. En d’autres termes, nous cherchons des estimateurs de Θ_n calculés exactement, conditionnellement à une approximation particulière qui n’agit que sur le vecteur $\mathbf{X}_{0:n-1}$. Dans cette section, nous traiterons simultanément les problèmes de filtrages mono- et multi-objet(s) dans les modèles (1.1.1) et (1.1.8).

Chapitre 4 - Dans le dernier chapitre de cette thèse, nous reconsidérons les modèles statistiques sur lesquels reposent les techniques de filtrage abordées aux Chapitres 2 et 3, qu’il s’agisse du problème de filtrage mono- ou multi-objet(s). L’utilisation des modèles HMC (1.1.1) ainsi que les modèles à sauts (1.1.8) implique en effet que les données traitées suivent des propriétés statistiques bien spécifiques. Néanmoins, ces propriétés statistiques peuvent ne pas être vérifiées en pratique. Il est donc légitime de se demander si d’autres modèles statistiques permettent de relâcher certaines propriétés de ces modèles classiques, tout en permettant le calcul d’estimateurs d’intérêt. À cet effet, nous rappelons que les modèles chaînes de Markov couple (*Pairwise Markov Chains = PMC*) et triplet permettent de prendre en compte des propriétés beaucoup plus générales et nous nous posons la question du calcul d’estimateurs d’intérêt dans ces modèles. En particulier, nous nous focalisons sur le problème original du filtrage multi-objets dans le contexte des modèles de Markov couple.

Enfin nous montrons que le filtrage mono-objet exact (au sens où l’on cherche à propager l’espérance conditionnelle $E(\mathbf{X}_n | \mathbf{y}_{0:n})$) est possible dans une certaine classe de modèles triplets, avec une complexité linéaire en nombre d’observations. Nous cherchons alors à appliquer ce résultat pour proposer une alternative aux techniques de filtrage dans des modèles à sauts linéaires et gaussiens (1.1.13)-(1.1.14), dans lesquels le calcul exact de l’estimateur optimal est un problème NP-complet.

Le Chapitre 4 est donc organisé ainsi :

- la Section 4.1 est consacrée à un rappel des modèles de Markov couple (qui généralisent les modèles HMC (1.1.1)) puis triplet (qui généralisent les modèles JMSS (1.1.8)). L’adaptation à ces modèles des techniques existantes de filtrage mono-objet est également rappelé ;
- Dans la Section 4.2 nous nous penchons sur le problème du filtrage multi-objets dans le cas où chaque objet et son observation associée suit un modèle couple ;
- Enfin, dans la Section 4.3 nous revenons au cas mono-objet, et nous exploitons le fait que le filtrage exact (au sens défini ci-dessus) est possible dans certains modèles triplet pour proposer une technique de filtrage originale dans les modèles (1.1.13)-(1.1.14). Notre démarche est la suivante. Partant d’un modèle à sauts (1.1.13)-(1.1.14), nous construisons le modèle triplet le plus proche possible de ce modèle de départ (au sens où il conserve les propriétés physiques du modèle originel, et qui est tel que la Divergence de Kullback-Leibler (DKL) entre les lois $p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ des deux modèles est minimale), et dans lequel il est possible de calculer exactement (sans approximation numérique, ni tirage MC) l’espérance conditionnelle d’intérêt $E(\mathbf{X}_n | \mathbf{y}_{0:n})$. Cette technique de filtrage exact ne se

restreint pas uniquement aux modèles à sauts (1.1.13)-(1.1.14), mais s'applique également à toute une classe de modèles partageant des propriétés physiques identiques.

1.4 Publications de l'auteur

Ce travail a fait l'objet de plusieurs articles dans des revues internationales (publiés, acceptés ou soumis) et dans des conférences internationales. Nous présentons ici la liste des différentes publications ainsi que leur lien avec les différentes parties de ce travail.

1.4.1 Articles de revues internationales avec comité de lecture

Articles soumis

- J.6** Y. Petetin and F. Desbouvries, A class of fast exact Bayesian filters in dynamical models with jumps, *IEEE Transactions on Signal Processing*, submitted for publication, September 2013.
- J.5** Y. Petetin and F. Desbouvries, Bayesian Conditional Monte Carlo Algorithms for Sequential Single and Multi-Object filtering, *Statistics and Computing*, submitted for publication, May 2013, under revision.

Articles publiés ou acceptés pour publication

- J.4** Y. Petetin and F. Desbouvries, Bayesian Multi-object filtering for Pairwise Markov Chains, *IEEE Transactions on Signal Processing*, Volume 61, Issue 18, Pages 4481-4490, September 2013.
- J.3** Y. Petetin, M. Morelande and F. Desbouvries, Marginalized particle PHD filters for multiple object Bayesian filtering, *IEEE Transactions on Aerospace and Electronic Systems*, accepted for publication, June 2013.
- J.2** Y. Petetin and F. Desbouvries, Optimal SIR algorithm vs. fully adapted auxiliary particle filter : a non asymptotical analysis, *Statistics and computing*, accepted for publication, August 2012. *Online first version* : <http://www.springerlink.com/content/w1n70050770h8664/?MUD=MP>.
- J.1** F. Desbouvries, Y. Petetin and B. Ait-el-Fquih, Direct, Prediction- and Smoothing-based Kalman and Particle Filter Algorithms, *Signal Processing*, Volume 91, Number 8, Pages 2064-2077, August 2011.

1.4.2 Articles de conférences internationales avec actes et comité de lecture

- C.10** Y. Petetin and F. Desbouvries, Multi-object Filtering for Pairwise Markov Chains, *Proceedings of the 11th International Conference on Information Sciences, Signal Processing and their Applications (ISSPA2012)*, Montreal, Quebec, Canada, July 3-5, 2012.
- C.9** Y. Petetin and F. Desbouvries, A Mixed GM/SMC Implementation of the Probability Hypothesis Density Filter, *Proceedings of the 11th International Conference on Information Sciences, Signal Processing and their Applications (ISSPA2012)*, Montreal, Quebec, Canada, July 3-5, 2012.
- C.8** Y. Petetin and F. Desbouvries, Marginalized PHD Filters for Multi-Target Filtering, *Proceedings of the 11th International Conference on Information Sciences, Signal Processing and their Applications (ISSPA2012)*, Montreal, Quebec, Canada, July 3-5, 2012.

- C.7** Y. Petetin and F. Desbouvries, Further Rao-Blackwellizing an Already Rao-Blackwellized Algorithm for Jump Markov State Space Systems, *Proceedings of the 11th International Conference on Information Sciences, Signal Processing and their Applications (ISSPA2012)*, Montreal, Quebec, Canada, July 3-5, 2012.
- C.6** Y. Petetin and F. Desbouvries, A Semi-exact Sequential Monte Carlo Filtering Algorithm in Hidden Markov Chains, *Proceedings of the 11th International Conference on Information Sciences, Signal Processing and their Applications (ISSPA2012)*, Montreal, Quebec, Canada, July 3-5, 2012.
- C.5** Y. Petetin and F. Desbouvries, A Particle Smoothing Implementation of the Fully-Adapted Auxiliary Particle Filter : An Alternative to Auxiliary Particle Filters, *Proceedings of the IEEE workshop on Statistical Signal Processing (SSP'11)*, Nice, France, June 28-30, 2011.
- C.4** F. Desbouvries, Y. Petetin and E. Monfrini, A Non Asymptotical Analysis of the Optimal SIR Algorithm vs. the Fully Adapted Auxiliary Particle Filter, *Proceedings of the IEEE workshop on Statistical Signal Processing (SSP'11)*, Nice, France, June 28-30, 2011.
- C.3** F. Desbouvries, Y. Petetin and E. Monfrini, Optimal SIR algorithm vs. fully-adapted auxiliary particle filter : a matter of conditional independence, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'11)*, Prague, Czech Republic, May 22-27, 2011.
- C.2** Y. Petetin, D. Clark, B. Ristic, and D. Maltese, An efficient CPHD filter based approach for multiple target tracking, *The International Society for Optical Engineering (SPIE)*, Orlando, USA, April, 2011.
- C.1** Y. Petetin and F. Desbouvries, Direct vs. indirect sequential Monte-Carlo filters, *Proceedings of the international conference on Stochastic Modeling Techniques and Data Analysis (SMTDA '10)*, Chania, Greece, June 8-11, 2010.

Section	Titre	Article(s) correspondant(s)
2.2	Algorithmes de filtrage particulaire direct, par prédiction et par lissage	J.1 - C.1
2.3	Implémentation pratique des algorithmes de filtrage particulaire	C.5
2.4	Comparaison théorique d'algorithmes de filtrage localement optimaux	J.2 - C.3 - C.4
3.2	Filtres particuliers PHD marginalisés pour le filtrage multi-objets bayésien	J.3 - C.8
3.3	Méthodes CMC temporelles	J.5 - C.6 - C.7 - C.9
4.2	Filtrage multi-objets pour chaînes de Markov couple	J.4 - C.10
4.3	Une classe d'algorithmes de filtrage exact rapide pour les modèles à sauts	J.6
D.2	Un pisteur robuste basé sur une approche CPHD	C.2

TABLE 1.1 – Correspondance entre les différentes publications et les parties de cette thèse.

Méthodes de Monte Carlo séquentielles

Sommaire

2.1	Le filtrage particulaire et le filtrage particulaire auxiliaire	34
2.2	Algorithmes de filtrage particulaire direct, par prédiction et par lissage	39
2.3	Implémentation pratique des algorithmes de filtrage particulaire	45
2.4	Comparaison théorique d'algorithmes de filtrage localement optimaux	53
2.5	Conclusion	75

Ce chapitre est dédié au problème de la propagation de la loi $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = p_{0:n|n}$ (ou plus particulièrement de la loi de filtrage $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = p_{n|n}$) ainsi qu'à l'estimation d'un moment d'une fonction d'intérêt $f(\cdot)$ par rapport à l'une de ces lois de probabilité dans le modèle HMC (1.1.1),

$$\Theta_n = \int f(\mathbf{x}_{0:n})p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})d\mathbf{x}_{0:n}. \quad (2.0.1)$$

Excepté dans le cas linéaire et gaussien caractérisé par les équations (1.1.6)-(1.1.7), ces lois ne sont pas calculables en général. Nous nous intéressons donc ici à l'utilisation des méthodes SMC pour ces problèmes d'estimation, qui permettent de propager une approximation discrète des lois d'intérêts. Le problème peut alors être ramené à la formulation suivante ; à l'instant $n - 1$, nous supposons que la loi $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$ est approximée par un jeu de points pondérés $\{\mathbf{x}_{0:n-1}^i, w_{n-1}^i\}_{i=1}^N$, c'est à dire $\hat{p}(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ et nous cherchons à déduire un nouvel ensemble $\{\mathbf{x}_{0:n}^i, w_n^i\}_{i=1}^N$ à partir duquel sont déduites les approximations respectives de $\hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$, $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:n})$ et Θ_n :

$$\hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_{0:n}^i}, \quad (2.0.2)$$

$$\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_n^i}, \quad (2.0.3)$$

$$\hat{\Theta}_n = \sum_{i=1}^N w_n^i f(\mathbf{x}_{0:n}^i). \quad (2.0.4)$$

L'objet de la première section du chapitre est de rappeler le principe du filtrage particulaire, permettant le passage de $\{\mathbf{x}_{0:n-1}^i, w_{n-1}^i\}_{i=1}^N$ à $\{\mathbf{x}_{0:n}^i, w_n^i\}_{i=1}^N$. Dans un premier temps, nous rappelons que ce problème peut être résolu par l'application séquentielle des techniques d'échantillonnage d'importance, menant à la classe des algorithmes d'échantillonnage d'importance séquentiel (*Sequential Importance Sampling* = SIS), des algorithmes d'échantillonnage d'importance séquentiel avec rééchantillonnage (*Sampling Importance Resampling* = SIR) et des algorithmes de filtrage particulaire auxiliaire (*Auxiliary Particle Filter* = APF).

Ensuite, dans la Section 2.2, nous nous focalisons sur la propagation théorique de la loi de filtrage $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. Les équations classiques (1.1.2)-(1.1.3) indiquent que la loi de filtrage à l'instant n est obtenue à partir de la loi de prédiction $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$. (1.1.2)-(1.1.3) reposent donc sur le chemin bayésien récursif $p_{n-1|n-1} \rightarrow p_{n|n-1} \rightarrow p_{n|n}$. L'objectif de la Section 2.2 est de proposer des chemins bayésiens alternatifs menant à la loi de filtrage $p_{n|n}$. En particulier, nous illustrons le rôle des lois de prédiction à un pas $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$ et de lissage à un pas $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$. Nous obtenons alors plusieurs « chemins » bayésiens faisant intervenir la loi d'intérêt $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. Pour chaque chemin, nous proposons une implémentation MC qui repose sur l'utilisation de trois opérations élémentaires (propagation, pondération et sélection, appelée également rééchantillonnage) dont les mécanismes seront également rappelés. Les algorithmes obtenus nous permettront alors de revisiter et de réinterpréter des algorithmes de filtrage particulière bien connus tel l'algorithme Bootstrap [55] ou l'algorithme de filtrage particulière auxiliaire pleinement adapté (*Fully Adapted Auxiliary Particle Filter* = FA-APF) [97], qui sera appelé plus simplement algorithme FA .

Dans la Section 2.3, nous proposons des approximations de ces algorithmes dans le cas où il ne peuvent être implémentés de façon exacte. C'est par exemple le cas de l'algorithme FA qui n'est implémentable que dans des classes de modèles particulières. Nous proposons une alternative à l'APF, qui peut être vu comme une implémentation par échantillonnage d'importance en dimension augmentée du FA, pour les cas où les performances de l'implémentation auxiliaire ne sont pas satisfaisantes. Cette alternative s'appuie en partie sur l'interprétation en termes de chemin bayésien que nous aurons faite dans la Section 2.2.

Nous serons enfin amenés à comparer certains des algorithmes de filtrage obtenus avec des cas particuliers des algorithmes SIS et SIR classiquement utilisés. Cette comparaison, qui est faite dans la Section 2.4, est motivée par le fait que certains algorithmes SMC de la Section 2.2 ne diffèrent que par l'ordre des opérations élémentaires qui les composent. Par conséquent, nous nous interrogeons sur l'impact statistique que cela peut avoir sur l'estimateur $\hat{\Theta}_n$. Partant d'un jeu de points $\{\mathbf{x}_{n-1}^i, w_{n-1}^i\}_{i=1}^N$, nous étudions les propriétés statistiques de $\hat{\Theta}_n$ en fonction de l'algorithme choisi dans le but de proposer un algorithme SMC minimisant à chaque instant la variance conditionnelle de $\hat{\Theta}_n$.

2.1 Le filtrage particulière et le filtrage particulière auxiliaire

2.1.1 Principe de l'échantillonnage d'importance

Rappelons dans un premier temps que le filtrage particulière est basé sur des techniques d'échantillonnage d'importance, appelé aussi échantillonnage préférentiel [57]. Supposons que l'on souhaite calculer le moment

$$\Theta = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}, \quad (2.1.1)$$

où $p(\mathbf{x})$ est une densité de probabilité connue à une constante près,

$$p(\mathbf{x}) \propto p'(\mathbf{x}), \quad \int p(\mathbf{x})d\mathbf{x} = 1. \quad (2.1.2)$$

Introduisons alors une distribution d'importance $q(\cdot)$ selon laquelle il est possible de faire des tirages et qui est choisie de façon à ce que le support de $p(\cdot)$ soit inclus dans celui de $q(\cdot)$. Θ peut être réécrit sous la forme

$$\Theta = \frac{1}{\int \frac{p'(\mathbf{x})}{q(\mathbf{x})}d\mathbf{x}} \int f(\mathbf{x}) \frac{p'(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x})d\mathbf{x}, \quad (2.1.3)$$

ce qui permet d'obtenir une approximation MC $\hat{\Theta}$ de Θ à partir de N échantillons $\mathbf{x}^{1:N} = \{\mathbf{x}^i\}_{i=1}^N$ i.i.d. suivant $q(\mathbf{x})$:

$$\hat{\Theta} = \sum_{i=1}^N w^i f(\mathbf{x}^i), \quad (2.1.4)$$

$$w^i = \frac{\frac{p'(\mathbf{x}^i)}{q(\mathbf{x}^i)}}{\sum_{j=1}^N \frac{p'(\mathbf{x}^j)}{q(\mathbf{x}^j)}}. \quad (2.1.5)$$

Les propriétés de l'estimateur obtenu dépendent bien sûr de la loi d'importance $q(\cdot)$ et ont été discutées dans plusieurs ouvrages et contributions [52] [104] [19]. En particulier, le choix $q(\mathbf{x}) \propto |f(\mathbf{x}) - \Theta|p(\mathbf{x})$ minimise la variance de l'estimateur $\hat{\Theta}$ [52], mais cette fonction n'est pas calculable puisqu'elle requiert justement la quantité Θ que l'on cherche à estimer.

L'intérêt pratique de cette technique se trouve dans la possibilité d'approcher Θ même si la loi cible $p(\mathbf{x})$ n'est connue qu'à une constante près. C'est justement le problème auquel nous sommes confrontés lorsque nous souhaitons calculer le moment Θ_n défini par (2.0.1), dans un modèle HMC. En effet, la loi $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ n'est connue qu'à une constante près, à partir de $p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ donnée dans (1.1.1). L'objectif est donc de construire un algorithme séquentiel basé sur l'échantillonnage d'importance, permettant de passer de $\hat{\Theta}_{n-1}$ à $\hat{\Theta}_n$. Supposons qu'on dispose de trajectoires $\{\mathbf{x}_{0:n-1}^i, w_{n-1}^i\}_{i=1}^N$ obtenues à partir de l'échantillonnage d'importance faisant intervenir la loi d'importance $q_{n-1}(\mathbf{x}_{0:n-1})$; cette loi $q_{n-1}(\mathbf{x}_{0:n-1})$ peut dépendre des observations $\mathbf{y}_{0:n-1}$, mais cette dépendance ne sera pas explicitée dans la notation. Nous cherchons à mettre à jour ces nouvelles trajectoires, ainsi que leur poids, à partir de la nouvelle observation \mathbf{y}_n . Dans le modèle (1.1.1), la relation entre $p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ et $p(\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1})$ s'écrit

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}) p(\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1}). \quad (2.1.6)$$

Choisissons une distribution d'importance à l'instant n déduite de celle à l'instant $n - 1$, et qui se factorise sous la forme

$$q_n(\mathbf{x}_{0:n}) = q_{n-1}(\mathbf{x}_{0:n-1}) q(\mathbf{x}_n | \mathbf{x}_{0:n-1}); \quad (2.1.7)$$

ainsi, comme nous le rappellerons au Paragraphe 2.2.1, il suffit de tirer des points $\mathbf{x}_n^i \sim q(\mathbf{x}_n | \mathbf{x}_{0:n-1}^i)$, pour $1 \leq i \leq N$ afin d'obtenir N points tirés suivant $q_n(\mathbf{x}_{0:n})$. D'après (2.1.6), l'expression du poids d'importance associé à la i -ème trajectoire s'écrit

$$w_n^i \propto \frac{p(\mathbf{x}_{0:n-1}^i, \mathbf{y}_{0:n-1}) p(\mathbf{x}_n^i, \mathbf{y}_n | \mathbf{x}_{n-1}^i)}{q_{n-1}(\mathbf{x}_{0:n-1}^i) q(\mathbf{x}_n^i | \mathbf{x}_{0:n-1}^i)}, \quad \sum_{i=1}^N w_n^i = 1, \quad (2.1.8)$$

$$\propto w_{n-1}^i \frac{p(\mathbf{x}_n^i, \mathbf{y}_n | \mathbf{x}_{n-1}^i)}{q(\mathbf{x}_n^i | \mathbf{x}_{0:n-1}^i)}, \quad \sum_{i=1}^N w_n^i = 1; \quad (2.1.9)$$

une approximation de $p_{0:n|n}$ et de Θ_n sont données respectivement par les équations (2.0.2) et (2.0.4) dans lesquelles les poids d'importance w_n^i sont donnés par (2.1.8). Il est enfin possible d'obtenir N' trajectoires tirées approximativement suivant $p_{0:n|n}$, de poids égaux à $1/N'$, à partir du mécanisme de Rubin [107] [49] [114], en tirant N' points selon (2.0.2). Cette étape optionnelle est généralement appelée étape de sélection ou étape de rééchantillonnage et son utilité sera discutée ci-dessous. Nous obtenons finalement l'Algorithme 2.1 [43] [42] [8] composé de trois opérations élémentaires (Propagation = **P**, Pondération = *Weighting* = **W**, Rééchantillonnage = **R**); lorsque l'algorithme est exécuté *sans* l'étape de rééchantillonnage, nous parlerons d'Algorithme SIS, et dans le cas contraire, d'Algorithme SIR.

Algorithme 2.1: Algorithme SIS/SIR

Soit $\{\mathbf{x}_{0:n-1}^i, w_{n-1}^i\}_{i=1}^N$ une approximation de $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$.

pour $1 \leq i \leq N$ **faire**

P. Tirer $\mathbf{x}_n^i \sim q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^i)$;

W. Calculer $w_n^i \propto w_{n-1}^i p(\mathbf{x}_n^i, \mathbf{y}_n|\mathbf{x}_{0:n-1}^i)/q(\mathbf{x}_n^i|\mathbf{x}_{0:n-1}^i)$, $\sum_{i=1}^N w_n^i = 1$;

fin

Alors $\hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_{0:n}^i}$ approxime $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ et $\hat{\Theta}_n = \sum_{i=1}^N w_n^i f(\mathbf{x}_{0:n}^i)$ approxime Θ_n .

si Rééchantillonnage alors

pour $1 \leq i \leq N'$ **faire**

R. $\tilde{\mathbf{x}}_{0:n}^i \sim \sum_{j=1}^N w_n^j \delta_{\mathbf{x}_{0:n}^j}$, $\tilde{w}_n^i = 1/N'$;

fin

 Poser $\{\mathbf{x}_{0:n}^i, w_n^i\}_{i=1}^{N'} = \{\tilde{\mathbf{x}}_{0:n}^i, \tilde{w}_n^i\}_{i=1}^{N'}$; $\hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \frac{1}{N'} \sum_{i=1}^{N'} \delta_{\mathbf{x}_{0:n}^i}$ approxime $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$.

fin

2.1.2 Considérations pratiques

Le rééchantillonnage

L'aspect pratique du rééchantillonnage réside dans la possibilité d'obtenir un jeu de N' points à partir d'un jeu de N points. Mais, et c'est l'aspect le plus important, l'algorithme décrit ci-dessus, sans étape de rééchantillonnage, ne peut pas être utilisé en pratique. En effet, la plupart des poids d'importance $\{w_n^i\}_{i=1}^N$ ont tendance à tendre vers 0 au bout de quelques itérations de l'algorithme. Par conséquent, très peu de trajectoires $\{\mathbf{x}_{0:n}^i\}_{i=1}^N$ participent à l'approximation $\hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ de $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$, qui devient très rapidement mauvaise. Ce phénomène de dégénérescence a été étudié dans certaines contributions [19] [34]. L'idée du rééchantillonnage est alors d'éliminer les particules de poids très faible et de dupliquer celles qui ont un poids important, de façon à recréer de la diversité lors de l'étape de propagation de l'itération suivante. Pour cela, nous effectuons N' tirages de telle manière que la trajectoire $\mathbf{x}_{0:n}^i$ est sélectionnée avec la probabilité w_n^i . **Excepté dans la Section 2.4, nous prendrons $N' = N$.** Plusieurs points sont à prendre en compte lors de cette étape :

- $\hat{\Theta}_n$ doit être calculé *avant* l'étape de rééchantillonnage. En effet, il est bien connu (et nous le rappellerons dans la Section 2.4) que le rééchantillonnage dégrade localement la qualité de l'approximation obtenue. Les bienfaits éventuels du rééchantillonnage n'interviennent que dans les itérations suivantes ;
- le rééchantillonnage est optionnel et peut être nocif pour les itérations suivantes. Il ne doit donc pas intervenir à chaque itération de l'algorithme. Quelques règles ont été établies dans de précédentes contributions à partir de la taille effective de l'échantillon $\{\mathbf{x}_{0:n}^i\}_{i=1}^N$ obtenu après les étapes de propagation et de pondération [79]. D'autres techniques d'adaptation, basées sur le filtrage particulaire auxiliaire ont été mises au point plus récemment [31]. Une des contributions de la Section 2.4 sera de mesurer l'impact du rééchantillonnage en fonction des paramètres du modèle (1.1.1) ;
- le rééchantillonnage présenté dans l'Algorithme 2.1 est le rééchantillonnage multinomial. Il est possible d'utiliser des procédures alternatives, éventuellement moins complexes, et qui réduisent la nocivité locale introduite par rééchantillonnage multinomial. En particulier, l'impact des techniques de rééchantillonnages stratifié [71], résiduel [126] [79], et systématique [22] [126] a été étudié théorique-

ment dans plusieurs contributions [40] [60]. Dans ce travail, nous ne discuterons pas de ces différentes méthodes de rééchantillonnage et nous choisirons le rééchantillonnage multinomial.

Choix de la loi d'importance $q(\mathbf{x}_n|\mathbf{x}_{0:n-1})$

Les performances de l'algorithme dépendent également du choix de la loi d'importance conditionnelle $q(\mathbf{x}_n|\mathbf{x}_{0:n-1})$. Rappelons qu'il doit être possible de faire des tirages suivant cette loi, et que son expression doit être connue pour le calcul des poids. Les deux choix suivants présentent chacun un intérêt particulier :

- La loi $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}) = f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$ [55] est fréquemment utilisée car l'expression cette loi est généralement connue, à partir de (1.1.1), et il est possible d'en obtenir des tirages. Dans ce cas particulier, l'expression des poids d'importance dans (2.1.8) est réduite à $w_n^i \propto w_{n-1}^i g_n(\mathbf{y}_n|\mathbf{x}_n^i)$, $\sum_{i=1}^N w_n^i = 1$, elle aussi aisément calculable ;
- Bien que très pratique, le choix précédent peut donner de mauvais résultats car l'observation \mathbf{y}_n disponible n'est pas prise en compte pour tirer de nouveaux points. Plus précisément, il n'est pas rare de se trouver dans une situation où le terme $g_n(\mathbf{y}_n|\mathbf{x}_n^i)$ est faible pour la plupart des particules. Après normalisation, la masse totale n'est alors répartie que sur très peu de particules. Pour résoudre ce problème, il est préférable de tirer les particules à partir de la loi

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) = \frac{f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})g_n(\mathbf{y}_n|\mathbf{x}_n)}{\int f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})g_n(\mathbf{y}_n|\mathbf{x}_n)d\mathbf{x}_n}. \quad (2.1.10)$$

Cette loi d'importance conditionnelle $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ prend en compte la nouvelle observation \mathbf{y}_n et est dite « localement optimale » au sens où la variance conditionnelle des poids w_n^i est nulle. En effet, en injectant $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ dans (2.1.8), nous obtenons

$$w_n^i \propto w_{n-1}^i p(\mathbf{y}_n|\mathbf{x}_{n-1}^i), \quad \sum_{i=1}^N w_n^i = 1$$

avec

$$p(\mathbf{y}_n|\mathbf{x}_{n-1}) = \int f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})g_n(\mathbf{y}_n|\mathbf{x}_n)d\mathbf{x}_n. \quad (2.1.11)$$

Nous observons alors que les nouveaux poids w_n^i ne dépendent plus des particules $\{\mathbf{x}_{n-1}^i\}_{i=1}^N$ qui viennent d'être tirées [127] [79] [43] d'où $\text{var}(w_n^i|\{\mathbf{x}_{0:n-1}^i\}_{i=1}^N, \mathbf{y}_{0:n}) = 0$. Ce choix n'est malheureusement pas toujours possible, car le calcul de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ et $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ n'est pas toujours réalisable. Des méthodes permettant d'approcher cette loi optimale ont été proposées et sont basées sur des techniques de linéarisation [110], l'utilisation de l'EKF [43] ou de l'UKF [119][66].

2.1.3 Le filtrage particulaire auxiliaire

Le filtrage particulaire auxiliaire utilise également l'échantillonnage d'importance mais pour une loi cible différente. En effet, repartons de la formule (2.1.6), dans laquelle nous injectons une approximation $\hat{p}(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ de $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$. Nous obtenons alors l'approximation $\pi_{n|n}(\mathbf{x}_n)$ de $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ donnée par

$$\pi_{n|n}(\mathbf{x}_n) = \sum_{i=1}^N \frac{w_{n-1}^i f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}^i)g_n(\mathbf{y}_n|\mathbf{x}_n)}{\sum_{j=1}^N w_{n-1}^j \int f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}^j)g_n(\mathbf{y}_n|\mathbf{x}_n)d\mathbf{x}_n}, \quad (2.1.12)$$

$$= \sum_{i=1}^N \frac{w_{n-1}^i p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)}{\sum_{j=1}^N w_{n-1}^j p(\mathbf{y}_n|\mathbf{x}_{n-1}^j)} p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n). \quad (2.1.13)$$

Pour obtenir une approximation de $p(\mathbf{x}_n | \mathbf{y}_{0:n})$, nous pouvons viser le mélange $\pi_{n|n}(\cdot)$, par l'intermédiaire d'un mélange d'importance $q(\cdot)$, en utilisant l'échantillonnage d'importance. L'application directe de cette méthode peut s'avérer complexe pour calculer les poids d'importance associés car elle impliquerait d'évaluer le terme $\sum_{i=1}^N w_{n-1}^i f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}^i) g_n(\mathbf{y}_n | \mathbf{x}_n)$ en chaque point \mathbf{x}_n^j tiré. Pour contourner cette difficulté, le mélange $\pi_{n|n}$ peut être interprété comme la marginale d'une loi $p(k, \mathbf{x}_n)$, avec $\Pr(k = i) \propto w_{n-1}^i p(\mathbf{y}_n | \mathbf{x}_{n-1}^i)$ et $p(\mathbf{x}_n | k = i) = p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n)$. Il est alors possible de viser cette loi, connue à une constante près, par échantillonnage d'importance en dimension augmentée en utilisant une loi d'importance $q(k, \mathbf{x}_n)$ telle que $q(k) \propto w_{n-1}^k \tau(\mathbf{x}_{0:n-1}^k)$ (la fonction $\tau(\cdot)$ est appelée poids de première espèce) et $q(\mathbf{x}_n | k) = q(\mathbf{x}_n | \mathbf{x}_{0:n-1}^k)$. Nous obtenons alors l'algorithme APF, résumé ci-dessous, qui consiste à tirer d'abord une composante k , suivant $q(k)$, puis une nouvelle particule suivant $q(\mathbf{x}_n | \mathbf{x}_{0:n-1}^k)$.

Algorithme 2.2: Algorithme APF

Soit $\hat{p}(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ une approximation MC de $p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1})$.

pour $1 \leq i \leq N$ **faire**

W.1 Calculer $\tilde{w}_{n-1}^i \propto w_{n-1}^i \tau(\mathbf{x}_{0:n-1}^i)$, $\sum_{i=1}^N \tilde{w}_{n-1}^i = 1$;

fin

pour $1 \leq i \leq N'$ **faire**

R. Tirer $\tilde{\mathbf{x}}_{0:n-1}^i \sim \sum_{j=1}^N \tilde{w}_{n-1}^j \delta_{\mathbf{x}_{0:n-1}^j}$;

P. Tirer $\mathbf{x}_n^i \sim q(\mathbf{x}_n | \tilde{\mathbf{x}}_{0:n-1}^i)$;

W.2 Calculer $w_n^i \propto \frac{f_{n|n-1}(\mathbf{x}_n^i | \tilde{\mathbf{x}}_{0:n-1}^i) g_n(\mathbf{y}_n | \mathbf{x}_n^i)}{\tau(\tilde{\mathbf{x}}_{0:n-1}^i) q(\mathbf{x}_n | \tilde{\mathbf{x}}_{0:n-1}^i)}$, $\sum_{i=1}^N w_n^i = 1$.

fin

Poser $\{\mathbf{x}_{0:n}^i\}_{i=1}^{N'} = \{\tilde{\mathbf{x}}_{0:n-1}^i, \mathbf{x}_n^i\}_{i=1}^{N'} : \hat{p}(\mathbf{x}_{0:n} | \mathbf{y}_{0:n}) = \sum_{i=1}^{N'} w_n^i \delta_{\mathbf{x}_{0:n}^i}$ approxime $p(\mathbf{x}_{0:n} | \mathbf{y}_{0:n})$.

Les performances de l'APF dépendent du choix des poids de première espèce $\tau(\mathbf{x}_{0:n-1})$ et de la loi d'importance $q(\mathbf{x}_n | \mathbf{x}_{0:n-1})$. Ces choix ont été discutés dans de nombreuses contributions [61] [31] [41] [63] [124] [30]. En particulier, le choix $\tau(\mathbf{x}_{0:n-1}) = 1$ donne l'Algorithme 2.1 avec rééchantillonnage, à l'ordre des opérations près : la première étape de rééchantillonnage de l'algorithme APF correspond, dans ce cas, à la dernière étape de rééchantillonnage de l'algorithme SIS/SIR. Le choix $\tau(\mathbf{x}_{0:n-1}) = p(\mathbf{y}_n | \mathbf{x}_{n-1})$ et $q(\mathbf{x}_n | \mathbf{x}_{0:n-1}) = p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$ donne, dans la terminologie de Pitt et Shephard [97], le FA. Dans ce cas, les points \mathbf{x}_n^i sont en effet tirés directement suivant le mélange $\pi_{n|n}$ et les poids sont égaux à $1/N'$. C'est pour cette raison qu'une approximation de ces deux lois est habituellement suggérée [124]. Notons alors que comme dans l'algorithme SIS/SIR dit « optimal », les lois $p(\mathbf{y}_n | \mathbf{x}_{n-1})$ et $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$ jouent un rôle central.

Remarque 2.1. Bien que localement optimaux (au sens où ils minimisent la variance conditionnelle des poids d'importance), les algorithmes SIS/SIR avec densité optimale conditionnelle et l'algorithme FA ne garantissent pas les meilleures performances lorsqu'il s'agit de calculer un estimateur de $\Theta_n = \int f(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{y}_{0:n}) d\mathbf{x}_n$, pour une fonction $f(\cdot)$ donnée [63]. Par exemple, pour ce problème, un choix optimal des poids de première espèce $\tau(\mathbf{x}_{0:n-1})$, basé sur une analyse asymptotique et différent de $p(\mathbf{y}_n | \mathbf{x}_{n-1})$, a été proposé dans [41]. Ce choix est cependant difficilement utilisable en pratique car ces poids optimaux dépendent de la quantité Θ_n que l'on cherche à calculer, même si des approximations, utilisant un « pré filtrage particulière » ont été proposées. En pratique, ce réglage optimal donne des résultats assez proche de ceux du FA (voir [41]). De plus, cette stratégie est d'intérêt pratique dès lors que nous avons besoin d'estimer un unique moment Θ_n ; dans le cas contraire le jeu de particules produit par cette stratégie peut être optimale pour un estimateur Θ_n^1 mais donner de mauvais résultats pour un estimateur Θ_n^2 . Comme l'ont souligné Whiteley et Johansen

[124], le jeu de particules produit par un filtre particulaire sert à la fois à approximer Θ_n et $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. C'est pour cette raison que nous nous concentrerons dans ce travail sur ces algorithmes localement optimaux.

2.2 Algorithmes de filtrage particulaire direct, par prédiction et par lissage

Les algorithmes SIS et SIR permettent d'obtenir une approximation de $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ et donc une approximation de $p(\mathbf{x}_j|\mathbf{y}_{0:n})$, pour $0 \leq j \leq n$. Cependant, l'étape (indispensable) de rééchantillonnage a tendance à appauvrir la diversité du jeu de points $\{\mathbf{x}_j^i\}_{i=1}^N$, en particulier lorsque $j \ll n$. Par conséquent, l'algorithme d'échantillonnage d'importance séquentiel est d'intérêt pratique et est généralement utilisé lorsque $f(\mathbf{x}_{0:n}) = f(\mathbf{x}_n)$ dans (2.0.1). Pour les autres cas, des algorithmes de lissage visant à résoudre ce problème ont été développés [43] [48] [90]. A présent, dans cette section, nous cherchons à propager directement des approximations de $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. Pour cela, nous commençons par rappeler des mécanismes de simulations probabilistes qui seront communs aux algorithmes proposés. Ensuite, nous proposons plusieurs algorithmes de filtrage, qu'il est possible de regrouper en trois classes, dans lesquelles la loi de filtrage $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ est obtenue de manière différente : la classe des algorithmes de filtrage direct, où nous cherchons à propager $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ en utilisant différentes lois intermédiaires ; la classe des algorithmes de filtrage par prédiction, dans laquelle nous propageons $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$ en utilisant différentes lois intermédiaires, $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ étant déduit de $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$; et enfin, la classe des algorithmes de filtrage par lissage, dans laquelle $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$ est propagée en utilisant différentes lois intermédiaires, et de laquelle on déduit $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. Les lois intermédiaires intervenant dans la propagation de la loi $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ sont présentées dans la Figure 2.1. Quelle que soit la classe considérée, la propagation des lois d'intérêts est composé d'une étape de propagation P dans laquelle est introduite le nouvel état caché et une étape de mise à jour (*update*) U dans laquelle est introduite la nouvelle observation. Pour chaque chemin bayésien obtenu, une implémentation SMC est proposée et s'appuie sur des techniques de simulation rappelées dans le paragraphe suivant.

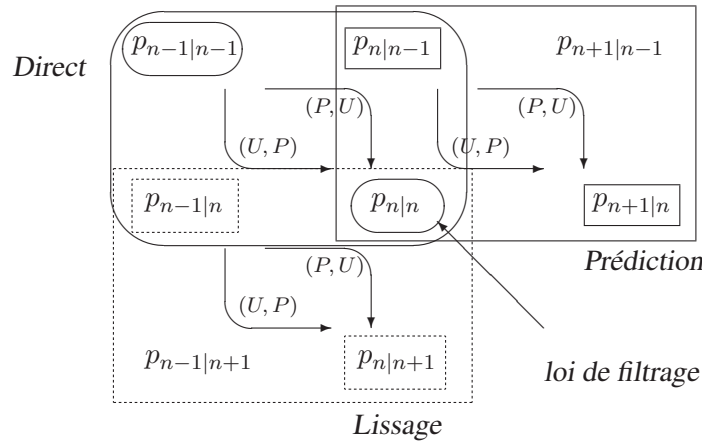


FIGURE 2.1 – Les chemins direct, par prédiction et par lissage pour propager une approximation de $p_{n|n}$. La loi de filtrage $p_{n|n}$, au centre, est déduite de $p_{n|n-1}$ ou $p_{n-1|n}$. La propagation de la loi $p_{n|n-1}$ (resp. $p_{n-1|n}$) se fait par le chemin $p_{n|n-1} \rightarrow p_n|n \rightarrow p_{n+1|n}$ (resp. $p_{n-1|n} \rightarrow p_n|n \rightarrow p_{n|n+1}$) ou $p_{n|n-1} \rightarrow p_{n+1|n-1} \rightarrow p_{n+1|n}$ (resp. $p_{n-1|n} \rightarrow p_{n-1|n+1} \rightarrow p_{n|n+1}$).

2.2.1 Quelques mécanismes de simulations

L'objet de ce paragraphe est de rappeler brièvement les mécanismes de simulations permettant le passage d'une approximation de $p(\mathbf{x}_1)$ à une approximation de $p(\mathbf{x}_2)$ liées par

$$p(\mathbf{x}_2) = \int p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1)d\mathbf{x}_1, \quad (2.2.1)$$

puis permettant le passage d'une approximation de $p(\mathbf{x})$ à une approximation de $p(\mathbf{x}|\mathbf{y})$ liées par

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}}. \quad (2.2.2)$$

Excepté dans le cas linéaire et Gaussien, $p(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1; \mathbf{m}_1; \mathbf{P}_1)$ (resp. $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{m}; \mathbf{P})$) et $p(\mathbf{x}_2|\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_2; \mathbf{F}\mathbf{x}_1; \mathbf{Q})$ (resp. $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{H}\mathbf{x}; \mathbf{R})$) où $p(\mathbf{x}_2)$ (resp. $p(\mathbf{x}|\mathbf{y})$) sont des Gaussiennes dont les paramètres sont calculables en utilisant les Lemmes A.1 (resp. A.2) en Annexe A, $p(\mathbf{x}_2)$ et $p(\mathbf{x}|\mathbf{y})$ ne peuvent pas être calculées exactement. Les méthodes MC reposent sur des tirages probabilistes et permettent d'approcher $p(\mathbf{x}_2)$ et $p(\mathbf{x}|\mathbf{y})$ à partir des techniques suivantes :

1. Échantillonnage hiérarchique : Soit $\mathbf{x}_1^i \sim p(\mathbf{x}_1)$ et $\mathbf{x}_2^i \sim p(\mathbf{x}_2|\mathbf{x}_1^i)$. Alors $(\mathbf{x}_1^i, \mathbf{x}_2^i)$ est tiré suivant $p(\mathbf{x}_1, \mathbf{x}_2)$;
2. Tirage suivant un mélange discret [99] : soit $p(\mathbf{x}_2) = \sum_{i=1}^N \alpha_i p_i(\mathbf{x}_2)$ un mélange. En tirant une composante $j \sim \sum_{i=1}^N \alpha_i \delta_i$ puis un point $\mathbf{x}_2^j \sim p_j(\mathbf{x}_2)$, alors \mathbf{x}_2^j est tiré suivant $p(\mathbf{x}_2)$;
3. Mécanisme de rééchantillonnage de Rubin [107] [55] : soient $\{\mathbf{x}^i\}_{i=1}^N$ N points tirés suivant $p(\mathbf{x})$, et conditionnellement à $\{\mathbf{x}^i\}_{i=1}^N$, $\{\tilde{\mathbf{x}}^i\}_{i=1}^J$ J points i.i.d suivant $\sum_{i=1}^N \frac{l(\mathbf{x}^i)}{\sum_{i=1}^N l(\mathbf{x}^i)} \delta_{\mathbf{x}^i}$. Alors $\{\tilde{\mathbf{x}}^i\}_{i=1}^J$ deviennent des points tirés i.i.d. suivant $l(\mathbf{x})p(\mathbf{x}) / \int l(\mathbf{x})p(\mathbf{x})d\mathbf{x}$ lorsque $N \rightarrow \infty$.

Ces techniques de simulations permettent alors d'implémenter (2.2.1) puis (2.2.2) à partir d'un des deux mécanismes suivants.

Mécanisme non pondéré

Dans le cas où nous disposons de tirages $\{\mathbf{x}_1^i\}_{i=1}^N$ i.i.d. suivant $p(\mathbf{x}_1)$, une approximation de $p(\mathbf{x}_2)$ est donnée par les points $\{\mathbf{x}_2^i\}_{i=1}^N$ qui sont tirés respectivement suivant $p(\mathbf{x}_2|\mathbf{x}_1^i)$. Enfin, si l'on dispose de points \mathbf{x}^i tirés suivant $p(\mathbf{x})$, une approximation de $p(\mathbf{x}|\mathbf{y})$ est obtenue en tirant les points $\tilde{\mathbf{x}}^i$ suivant la loi discrète $\sum_{k=1}^N \frac{p(\mathbf{y}|\mathbf{x}^k)}{\sum_{j=1}^N p(\mathbf{y}|\mathbf{x}^j)} \delta_{\mathbf{x}^k}$.

Mécanisme pondéré

Partons d'une approximation pondérée $\hat{p}(\mathbf{x}_1) = \sum_{i=1}^N w_1^i \delta_{\mathbf{x}_1^i}$ de $p(\mathbf{x}_1)$. En injectant cette approximation dans l'expression de $p(\mathbf{x}_2)$, nous obtenons une approximation $\tilde{p}(\mathbf{x}_2)$ donnée par le mélange de lois $\sum_{i=1}^N w_1^i p(\mathbf{x}_2|\mathbf{x}_1^i)$. Une approximation de $p(\mathbf{x}_2)$ est alors obtenue en tirant des points \mathbf{x}_2^i suivant ce mélange. Enfin, partant de N points $\{\mathbf{x}^i\}_{i=1}^N$ tirés suivant $p(\mathbf{x})$ (donc de poids égaux à $1/N$), une approximation pondérée de $p(\mathbf{x}|\mathbf{y})$ est donnée par les points \mathbf{x}^i associés aux poids $w^i \propto p(\mathbf{y}|\mathbf{x}^i)$, $\sum_{i=1}^N w^i = 1$.

Remarque 2.2. Les deux mécanismes sont équivalents lorsque les étapes (2.2.1)-(2.2.2) sont exécutées de façon séquentielle. Dans le premier cas, nous commençons par une étape de tirage, puis de pondération et enfin de rééchantillonnage. Dans le second cas nous commençons par une étape de rééchantillonnage, puis par une étape de tirage et enfin par une étape de pondération. À un décalage près, les mécanismes sont donc identiques. De ce fait, nous partirons toujours d'approximations discrètes pondérées.

Nous appliquons ces techniques dans les paragraphes suivants, afin d'implémenter un chemin bayésien faisant intervenir la loi de filtrage $p(\mathbf{x}_n|\mathbf{y}_{0:n})$.

2.2.2 Filtrage direct

Dans ce paragraphe, nous présentons deux algorithmes de filtrage direct pour approcher $p(\mathbf{x}_n|\mathbf{y}_{0:n})$. Ainsi, nous verrons que les algorithmes Bootstrap [55] et FA [97] correspondent à une implémentation MC de deux chemins bayésiens directs.

Filtrage exact

Partons de (2.1.6). Dans le modèle (1.1.1), $p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{n-1})$ peut se factoriser de deux manières,

$$p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{n-1}) = f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})g_n(\mathbf{y}_n|\mathbf{x}_n), \quad (2.2.3)$$

$$= p(\mathbf{y}_n|\mathbf{x}_{n-1})p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n), \quad (2.2.4)$$

où $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ et $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ sont respectivement données dans (2.1.10) et dans (2.1.11).

- Injectons (2.2.3) dans (2.1.6) puis marginalisons par rapport à $\mathbf{x}_{0:n-1}$; nous obtenons alors les équations classiques

$$p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) = \int f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1}, \quad (2.2.5)$$

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \frac{g_n(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}{\int g_n(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1})d\mathbf{x}_n}. \quad (2.2.6)$$

qui traduisent le passage de $p_{n-1|n-1}$ à $p_{n|n}$, par l'intermédiaire $p_{n|n-1}$. Étant donné que le chemin $p_{n-1|n-1} \rightarrow p_{n|n-1} \rightarrow p_{n|n}$ implique la loi de prédiction à un pas $p_{n|n-1}$, ce chemin est noté 1-P ;

- À l'opposé, si nous injectons (2.2.4) dans (2.1.6), nous obtenons, après marginalisation,

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) \underbrace{\frac{p(\mathbf{y}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})}{\int p(\mathbf{y}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1}}}_{p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})} d\mathbf{x}_{n-1}, \quad (2.2.7)$$

qui correspond au chemin $p_{n-1|n-1} \rightarrow p_{n-1|n} \rightarrow p_{n|n}$. Contrairement au chemin précédent, c'est ici la loi de lissage (*smoothing*) à un pas, $p_{n-1|n}$, qui est impliquée ; ce chemin sera donc appelé 1-S.

Excepté dans le cas linéaire et gaussien (1.1.6)-(1.1.7) où il est possible de calculer exactement $p_{n|n}$ ainsi que les distributions intermédiaires $p_{n|n-1}$ ou $p_{n-1|n}$ en utilisant les Lemmes A.1 et A.2 en Annexe A, les équations précédentes ne sont pas implémentables en pratique. Nous proposons donc les deux algorithmes génériques SMC correspondant.

Implémentation MC de 1-P

Supposons qu'à l'instant $n - 1$ nous disposons de N points $\{\mathbf{x}_{n-1}^i, w_{n-1}^i\}_{i=1}^N$ tirés approximativement suivant $p_{n-1|n-1}$. En utilisant les résultats du Paragraphe 2.2.1, nous obtenons alors N points tirés suivant $p_{n|n-1}$ en tirant $\mathbf{x}_n^i \sim \sum_{i=1}^N w_{n-1}^i f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$, puis N points tirés approximativement suivant $p_{n|n}$ en pondérant chaque \mathbf{x}_n^i par $w_n^i \propto g_n(\mathbf{y}_n|\mathbf{x}_n^i)$, $\sum_{i=1}^N w_n^i = 1$. Nous obtenons alors l'Algorithme 2.3 qui correspond, à l'ordre des opérations **P**, **W** et **R**, près, à l'algorithme Bootstrap de Gordon, Salmond et Smith [55].

Algorithme 2.3: Algorithme 1- P = Algorithme Bootstrap

Soit $\hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{n-1}^i}$ une approximation MC de $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$.

pour $1 \leq i \leq N$ **faire**

- R.** Tirer $\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{j=1}^N w_{n-1}^j \delta_{\mathbf{x}_{n-1}^j}$
- P.** Tirer $\mathbf{x}_n^i \sim f_{n|n-1}(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i)$
- W.** Calculer $w_n^i \propto p(\mathbf{y}_n|\mathbf{x}_n^i)$, $\sum_{i=1}^N w_n^i = 1$

fin

Finalement, $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_n^i}$ approxime $p(\mathbf{x}_n|\mathbf{y}_{0:n})$.

Implémentation MC de 1- S

Toujours à partir d'une approximation $\{\mathbf{x}_{n-1}^i, w_{n-1}^i\}_{i=1}^N$ de $p_{n-1|n-1}$, nous cherchons une approximation de $p_{n|n}$ déduite d'une approximation de $p_{n-1|n}$. Cette dernière approximation est obtenue en gardant les mêmes particules mais en réactualisant les poids w_{n-1}^i par $\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$, $\sum_{i=1}^N \tilde{w}_{n-1}^i = 1$. Enfin, une approximation MC de $p_{n|n}$ est obtenue en injectant l'approximation de $p_{n-1|n}$ dans (2.2.7), qui donne $\tilde{p}(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$. En tirant des points suivant ce mélange, nous obtenons alors l'Algorithme 2.4, qui correspond à l'algorithme FA de Pitt et Shephard [97].

Algorithme 2.4: Algorithme 1- S = Algorithme FA

Soit $\hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{n-1}^i}$ une approximation MC de $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$.

pour $1 \leq i \leq N$ **faire**

- W.** Calculer $\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$, $\sum_{i=1}^N \tilde{w}_{n-1}^i = 1$;

fin

pour $1 \leq i \leq N$ **faire**

- R.** Tirer $\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{j=1}^N \tilde{w}_{n-1}^j \delta_{\mathbf{x}_{n-1}^j}$;
- P.** Tirer $\mathbf{x}_n^i \sim p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$.

fin

Finalement, $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N \frac{1}{N} \delta_{\mathbf{x}_n^i}$ approxime $p(\mathbf{x}_n|\mathbf{y}_{0:n})$.

Remarque 2.3. Notons que l'Algorithme 2.4 est très proche de l'Algorithme 2.1 lorsque la distribution d'importance choisie est $q(\mathbf{x}_n|\mathbf{x}_{0:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ (voir la discussion sur le choix de $q(\mathbf{x}_n|\mathbf{x}_{0:n-1})$ dans les algorithmes SIS/SIR au Paragraphe 2.1.2). Dans ce cas, l'étape de Propagation ainsi que l'étape de Pondération sont identiques pour les deux algorithmes. Les deux algorithmes ne diffèrent que par l'étape de rééchantillonnage : dans l'algorithme 1- S , c'est l'ensemble $\{\mathbf{x}_{n-1}^i\}_{i=1}^N$ qui est rééchantillonné à partir des poids w_{n-1}^i , avant l'étape de propagation ; dans l'algorithme SIR, c'est l'ensemble $\{\mathbf{x}_n^i\}_{i=1}^N$ qui est rééchantillonné à partir des poids w_n^i , après l'étape de propagation. L'ensemble $\{\mathbf{x}_n^i\}_{i=1}^N$ de l'algorithme SIR avec densité d'importance conditionnelle optimale, n'est donc pas déduit de l'approximation de $p_{n-1|n}$, contrairement à l'algorithme FA. Cette différence, ainsi que son impact sur l'estimateur de Θ_n résultant, seront étudiés précisément à la Section 2.4.

2.2.3 Filtrage prédictif

Dans ce paragraphe, nous cherchons toujours à approcher la loi $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ mais nous propageons (approximativement) la loi de prédiction à un pas $p(\mathbf{x}_n|\mathbf{y}_{0:n-1})$; nous déduisons alors de (1.1.3) une approxi-

mation de la loi de filtrage $p(\mathbf{x}_n|\mathbf{y}_{0:n})$.

Prediction exacte

Dans le modèle (1.1.1), nous avons

$$p(\mathbf{x}_{0:n+1}|\mathbf{y}_{0:n}) = \frac{p(\mathbf{x}_{n+1}, \mathbf{y}_n|\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n-1}). \quad (2.2.8)$$

Le rôle joué par la transition $p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{n-1})$ dans les algorithmes de filtrage direct est maintenant joué par $p(\mathbf{x}_{n+1}, \mathbf{y}_n|\mathbf{x}_n)$, dont les deux factorisations de Bayes coïncident :

$$p(\mathbf{x}_{n+1}, \mathbf{y}_n|\mathbf{x}_n) = \overbrace{f_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n)}^{p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n)} g_n(\mathbf{y}_n|\mathbf{x}_n), \quad (2.2.9)$$

$$= \underbrace{g_n(\mathbf{y}_n|\mathbf{x}_n)}_{p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{x}_{n+1})} f_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n). \quad (2.2.10)$$

Injectons alors (2.2.9) dans (2.2.8) et intégrons par rapport à $\mathbf{x}_{0:n}$: nous obtenons alors (2.2.5)-(2.2.6), qui comme nous l'avons déjà vu, correspond à 1- P , à un décalage près ; maintenant, injectons (2.2.10) dans (2.2.8) et marginalisons par rapport à $\mathbf{x}_{0:n}$: nous obtenons

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) = \int \frac{g_n(\mathbf{y}_n|\mathbf{x}_n) \overbrace{f_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}^{p(\mathbf{x}_n, \mathbf{x}_{n+1}|\mathbf{y}_{0:n-1})}}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})} d\mathbf{x}_n; \quad (2.2.11)$$

par ailleurs, la transition $p_{n|n-1} \rightarrow p_{n+1|n}$ peut également s'écrire

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:n-1}) \int \overbrace{f_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}^{p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1})} d\mathbf{x}_n}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1})}. \quad (2.2.12)$$

Les équations (2.2.5)-(2.2.6) et (2.2.12) décrivent deux façons différentes de passer de $p_{n|n-1}$ à $p_{n+1|n}$. La première est le chemin $p_{n|n-1} \rightarrow p_{n|n} \rightarrow p_{n+1|n}$, déjà étudié dans le Paragraphe 2.2.2 ; tandis que (2.2.12) décrit le chemin $p_{n|n-1} \rightarrow p_{n+1|n-1} \rightarrow p_{n+1|n}$. Enfin, (2.2.11) est un chemin alternatif, $p_{n|n-1} \rightarrow p_{n,n+1|n-1} \rightarrow p_{n+1|n}$ qui s'avère utile pour l'implémentation MC, et qui peut être vu comme une implémentation trajectorielle de $p_{n|n-1}$, au sens où la propagation de $p_{n|n-1}$ repose sur la loi conditionnelle de la trajectoire $(\mathbf{X}_{n-1}, \mathbf{X}_n)$. L'implémentation MC du chemin $p_{n|n-1} \rightarrow p_{n|n} \rightarrow p_{n+1|n}$ étant donnée par l'Algorithme 2.3, nous nous focalisons sur l'implémentation du second chemin, qui fait intervenir la distribution de prédiction à deux pas et que nous appelons donc 2- P .

Implémentation MC de 2- P

Remarquons qu'une implémentation du chemin (2.2.12) requiert la connaissance de $p(\mathbf{y}_n|\mathbf{x}_{n+1}, \mathbf{y}_{0:n-1})$, qui n'est pas calculable en général. On préférera donc se concentrer sur l'implémentation de (2.2.11). L'Al-

gorithme 2.5 présente l'implémentation MC de (2.2.11) déduite des résultats du Paragraphe 2.2.1.

Algorithme 2.5: Algorithme 2- P

Soit $\hat{p}(\mathbf{x}_n | \mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_{n-1}^i}$.

pour $1 \leq i \leq N$ **faire**

P. Tirer $\tilde{\mathbf{x}}_{n+1}^i \sim f_{n+1}(\mathbf{x}_{n+1} | \mathbf{x}_n^i)$;

W. Calculer $w_{n+1}^i \propto w_n^i p(\mathbf{y}_n | \mathbf{x}_n^i)$, $\sum_{i=1}^N w_{n+1}^i = 1$;

fin

pour $1 \leq i \leq N$ **faire**

R. Tirer $\mathbf{x}_{n+1}^i \sim \sum_{j=1}^N w_{n+1}^j \delta_{\tilde{\mathbf{x}}_{n+1}^j}$;

fin

Alors $\hat{p}(\mathbf{x}_{n+1} | \mathbf{y}_{0:n}) = \sum_{i=1}^N \frac{1}{N} \delta_{\mathbf{x}_{n+1}^i}$ et $\hat{p}(\mathbf{x}_n | \mathbf{y}_{0:n}) = \sum_{i=1}^N w_{n+1}^i \delta_{\mathbf{x}_n^i}$.

Remarque 2.4. Dans l'Algorithme 2.5 les nouveaux points $\{\tilde{\mathbf{x}}_{n+1}^i\}_{i=1}^N$ sont pondérés selon un terme proportionnel à $w_n^i p(\mathbf{y}_n | \mathbf{x}_n^i)$ (indépendant de $\tilde{\mathbf{x}}_{n+1}^i$), puis rééchantillonnés selon les nouveaux poids ; dans l'Algorithme 2.3, les pondérations sont identiques mais c'est l'ancien jeu de points $\{\mathbf{x}_n^i\}_{i=1}^N$ qui est rééchantillonné, avant l'étape de propagation. Par conséquent, l'Algorithme 2- P est une version réorganisée de l'Algorithme 1- P et la comparaison de ces deux algorithmes pour le problème de prédiction à un pas, est identique à celle de l'algorithme SIR, avec densité d'importance optimale, avec l'algorithme 1- S pour le problème de filtrage, aux lois de tirage et poids près.

2.2.4 Filtrage lissé

Enfin, nous nous focalisons sur la propagation de la loi de lissage à un pas $p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n})$. La loi de filtrage $p(\mathbf{x}_n | \mathbf{y}_{0:n})$ est déduite à partir de (2.2.7).

Lissage exact

Dans le modèle (1.1.1), les lois de lissage à un pas $p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n})$ et $p(\mathbf{x}_n | \mathbf{y}_{0:n+1})$ sont liées par

$$p(\mathbf{x}_n | \mathbf{y}_{0:n+1}) = \int \frac{p(\mathbf{x}_n, \mathbf{y}_{n+1} | \mathbf{x}_{0:n-1}, \mathbf{y}_{0:n})}{p(\mathbf{y}_{n+1} | \mathbf{y}_{0:n})} p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n}) d\mathbf{x}_{n-1}. \quad (2.2.13)$$

De même, la loi $p(\mathbf{x}_n, \mathbf{y}_{n+1} | \mathbf{x}_{0:n-1}, \mathbf{y}_{0:n})$ est factorisable de deux manières différentes :

$$p(\mathbf{x}_n, \mathbf{y}_{n+1} | \mathbf{x}_{0:n-1}, \mathbf{y}_{0:n}) = p(\mathbf{y}_{n+1} | \mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n), \quad (2.2.14)$$

$$= p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n, \mathbf{y}_{n+1}) p(\mathbf{y}_{n+1} | \mathbf{x}_{n-1}, \mathbf{y}_n). \quad (2.2.15)$$

Si nous injectons (2.2.14) dans (2.2.13), nous retombons alors sur (2.2.7), à un décalage près, qui décrit le chemin $p_{n-1|n} \rightarrow p_{n|n} \rightarrow p_{n|n+1}$. L'implémentation de ce chemin correspond à l'Algorithme 2.4. A présent, injectons (2.2.15) dans (2.2.13). Nous obtenons

$$p(\mathbf{x}_n | \mathbf{y}_{0:n+1}) = \int p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n, \mathbf{y}_{n+1}) \underbrace{\frac{p(\mathbf{y}_{n+1} | \mathbf{x}_{n-1}, \mathbf{y}_n) p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n})}{p(\mathbf{y}_{n+1} | \mathbf{y}_{0:n})}}_{p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n+1})} d\mathbf{x}_{n-1}, \quad (2.2.16)$$

2.3 Implémentation pratique des algorithmes de filtrage particulière

qui décrit le chemin $p_{n-1|n} \rightarrow p_{n-1|n+1} \rightarrow p_{n|n+1}$. Ce chemin, appelé 2- S car il implique la loi de lissage à deux pas $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n+1})$, fait intervenir les lois

$$p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{y}_n) = \int p(\mathbf{y}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)d\mathbf{x}_n, \quad (2.2.17)$$

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n, \mathbf{y}_{n+1}) \propto p(\mathbf{y}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) \quad (2.2.18)$$

Implémentation MC de 2- S

L'algorithme ci-dessous décrit une implémentation MC du chemin 2- S . Partant d'un jeu de points $\{\mathbf{x}_{n-1}^i, w_{n-1}^i\}_{i=1}^N$ approximant $p_{n-1|n}$, l'algorithme est composé d'une étape de pondération, de rééchantillonnage et de propagation.

Algorithme 2.6: Algorithme 2- S

Soit $\hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{n-1}^i}$ une approximation MC de $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$.

pour $1 \leq i \leq N$ **faire**

 | **W.** Calculer $\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$, $\sum_{i=1}^N \tilde{w}_{n-1}^i = 1$;

fin

pour $1 \leq i \leq N$ **faire**

 | **R.** Tirer $\tilde{\mathbf{x}}_{n-1}^i \sim \sum_{j=1}^N \tilde{w}_{n-1}^j \delta_{\mathbf{x}_{n-1}^j}$

 | **P.** Tirer $\mathbf{x}_n^i \sim p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n, \mathbf{y}_{n+1})$;

fin

Alors $p(\mathbf{x}_n|\mathbf{y}_{0:n+1}) = \sum_{i=1}^N \frac{1}{N} \delta_{\mathbf{x}_n^i}$ approxime $p(\mathbf{x}_n|\mathbf{y}_{0:n+1})$.

pour $1 \leq i \leq N$ **faire**

 | Tirer $\bar{\mathbf{x}}_n^i \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$;

fin

Alors $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N \frac{1}{N} \delta_{\bar{\mathbf{x}}_n^i}$ approxime $p(\mathbf{x}_n|\mathbf{y}_{0:n})$.

Remarque 2.5. Comme dans les cas précédents, il est possible d'obtenir une version trajectorielle de l'algorithme 2- S , appelée 2'- S , qui propagerait alors $p_{n-1|n} \rightarrow p_{n-1,n|n+1} \rightarrow p_{n|n+1}$. On obtiendrait alors l'Algorithme 2.6 dans lequel les étapes de rééchantillonnage et de tirage seraient inversées. La comparaison entre cette version trajectorielle et l'algorithme 2- S est donc équivalente à celle de l'algorithme SIR, avec densité d'importance conditionnelle optimale, avec l'algorithme FA. Le but de la Section 2.4 sera de comparer les estimateurs de Θ_n dans (2.0.1) issus des algorithmes SIR/SIS « optimaux » et de l'algorithme FA, lorsque $f(\mathbf{x}_{0:n}) = f(\mathbf{x}_n)$. L'analyse que nous ferons peut être adaptée au cas où nous souhaitons comparer les estimateurs d'un moment d'intérêt suivant la loi de prédiction (resp. lissage) $p_{n|n-1}$ (resp. $p_{n|n+1}$) issus des Algorithmes 1- P (resp. 2- S) et 2- P (resp. 2'- S).

2.3 Implémentation pratique des algorithmes de filtrage particulière

Le but de ce paragraphe est de s'intéresser à l'implémentation pratique des algorithmes considérés précédemment. Plus précisément, pour les algorithmes basés sur une étape de prédiction (1- P et 2- P), nous avons besoin des lois $f_{n|n-1}$ et g_n dont nous supposons la disponibilité dès lors que nous nous donnons le modèle (1.1.1). Pour les algorithmes de lissage, nous avons besoin de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ et $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ pour 1- S et $p(\mathbf{y}_{n+1}|\mathbf{y}_n, \mathbf{x}_{n-1})$ et $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n, \mathbf{y}_{n+1})$ pour 2- S . En fonction du modèle considéré, nous discutons de l'implémentation exacte ou approchée de ces algorithmes. Lorsque des simulations informatiques

sont impliquées, nous générons $K = 100$ trajectoires, d'une longueur $T = 50$ et nous exécutons les algorithmes $P = 30$ fois, pour chaque trajectoire. Pour une trajectoire donnée k , nous calculons à chaque instant n $\text{EQM}_k(n) = \frac{1}{P} \sum_{p=1}^P (\hat{x}_{n|n}^{k,p} - x_n^k)^2$, où $\hat{x}_{n|n}^{k,p}$ et x_n^k sont respectivement les états estimé et vrai à l'instant n , de la p -ième réalisation, pour la k -ième trajectoire. Les EQM sont moyennés sur les diverses trajectoires, $\text{EQM}_{\text{moy}}(n) = \frac{1}{K} \sum_{k=1}^K \text{EQM}_k(n)$ et nous calculons finalement la moyenne temporelle de la racine de l'EQM, $\mathcal{J} = \frac{1}{T} \sum_{n=1}^T [\text{EQM}_{\text{moy}}(n)]^{\frac{1}{2}}$.

2.3.1 Simulations dans un modèle linéaire et gaussien

Dans ce paragraphe, nous comparons les algorithmes de filtrage proposés sur le modèle linéaire et Gaussien suivant :

$$\begin{cases} X_n &= 0.2X_{n-1} + U_n \\ Y_n &= 5X_n + V_n \end{cases}, \quad (2.3.1)$$

où U_n et V_n sont i.i.d, mutuellement indépendants de X_0 , avec $X_0 \sim \mathcal{N}(\cdot; 0.5; 0.5)$, $U_n \sim \mathcal{N}(\cdot; 0; Q)$ et $V_n \sim \mathcal{N}(\cdot; 0; R)$. Dans ce modèle, le FK est optimal et servira de solution de référence pour la comparaison des algorithmes SMC. Pour les cinq algorithmes SMC considérés (SIR avec densité d'importance conditionnelle optimale, 1- P , 2- P , 1- S et 2- S), nous avons $f_{n+1|n}(x_{n+1}|x_n) = \mathcal{N}(x_{n+1}; 0.2x_n; Q)$, $g_n(y_n|x_n) = \mathcal{N}(y_n; 5x_n; R)$; $p(x_n|x_{n-1}, y_n) = \mathcal{N}(x_n; m; P)$ et $p(y_n|x_{n-1}) = \mathcal{N}(y_n; x_{n-1}; \Sigma)$ avec $m = \frac{0.2R}{R+25Q}x_{n-1} + \frac{5Q}{R+25Q}y_n$, $P = \frac{RQ}{R+25Q}$, et $\Sigma = R + 25Q$; et enfin $p(x_n|x_{n-1}, y_n, y_{n+1}) = \mathcal{N}(x_n; \frac{\Sigma}{\Sigma+P}m + \frac{P}{\Sigma+P}y_{n+1}; \frac{\Sigma P}{\Sigma+P})$ et $p(y_{n+1}|x_{n-1}, y_n) = \mathcal{N}(y_{n+1}; m; \Sigma + P^2)$. Ces résultats découlent des Lemmes A.1 et A.2 en Annexe A.

Nous prenons $N = 100$ et $R = 2$. La Table 2.1 présente les résultats pour différentes valeurs de Q . Les solutions basées sur les algorithmes de lissage donnent de meilleurs résultats que ceux basés sur les algorithmes de prédiction. Pour une classe donnée (prédiction ou lissage), nous obtenons de meilleurs résultats lorsque l'étape de mise à jour, qui prend en compte l'arrivée d'une nouvelle observation, précède l'étape de propagation. L'algorithme 1- S améliore de façon très légère l'algorithme SIR. Pour $Q = 0.1$, tous les algorithmes se comportent de manière identique mais les algorithmes prédictifs se dégradent rapidement lorsque Q augmente. Dans ce cas, les fortes variations de x_n sont mieux prises en compte lorsque y_n (pour SIR et 1- S) ou même y_n et y_{n+1} (pour 2- S) sont pris en compte.

	2- P	1- P	SIR	1- S	2- S	FK
$Q = 0.1$	0.2183713	0.2155558	0.2147512	0.2134734	0.2129922	0.2126259
$Q = 1$	0.3489346	0.2844732	0.2754586	0.2739999	0.2731135	0.2726688
$Q = 3$	0.6020060	0.3092687	0.2820246	0.2809878	0.2809739	0.2794307
$Q = 5$	0.8511697	0.3287174	0.2829190	0.2819416	0.2815135	0.2801607
$Q = 10$	1.3505633	0.3723547	0.2843347	0.2833163	0.2830501	0.2817664

TABLE 2.1 – Critère \mathcal{J} - Modèle linéaire et gaussien (2.3.1).

2.3.2 Le PS-APF

Les algorithmes localement optimaux requièrent la connaissance de $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ et de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$. Néanmoins, les expressions de ces lois reposent sur le calcul d'une intégrale (voir (2.1.11) et (2.1.10)). Les

algorithmes localement optimaux ne peuvent donc pas être implémentés si le calcul de ces intégrales n'est pas possible, ce que nous supposons dans tout ce paragraphe.

Les algorithmes SIS/SIR (cf. Algorithme 2.1) peuvent toujours être utilisés car la densité d'importance $q(\mathbf{x}_n|\mathbf{x}_{0:n-1})$ peut être choisie arbitrairement, même si la densité conditionnelle optimale n'est pas disponible. Une autre alternative consiste à utiliser l'APF, qui peut être vu comme une implémentation générale du FA, basée sur l'échantillonnage d'importance en dimension augmentée, lorsqu'il n'est pas possible de faire des tirages suivant le mélange $\pi_{n|n}$ dans (2.1.13). Les performances de l'APF sont cependant sensibles au choix des poids de première espèce $\tau(\mathbf{x}_{0:n-1})$ et de la loi d'importance $q(\mathbf{x}_n|\mathbf{x}_{0:n-1})$, qui doivent idéalement approcher $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ et $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ [63] [124] (cf. Remarque 2.1). Néanmoins, l'approximation analytique de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ est un problème difficile qui peut nécessiter un coût de calcul important ; un mauvais choix des poids de première espèce peut alors produire des estimateurs moins performants que ceux déduits des algorithmes SIS/SIR [124] [41].

Dans l'esprit du FA et de l'APF, la solution proposée ici consiste à rééchantillonner d'abord l'ancien jeu de particules à l'instant $n - 1$, en prenant en compte la nouvelle observation \mathbf{y}_n , avant de tirer de nouvelles particules. Plus précisément, l'objet de ce paragraphe est d'utiliser l'interprétation du Paragraphe 2.2.2 de l'algorithme FA, qui implémente le chemin $p_{n-1|n-1} \rightarrow p_{n-1|n} \rightarrow p_{n|n}$, pour construire une implémentation alternative de l'algorithme FA. La différence entre l'algorithme proposé ici et l'APF est que nous n'utilisons plus la fonction $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ et n'avons donc plus besoin de l'approximer ; ainsi notre algorithme s'avère être une alternative à l'APF dans des situations où les approximations sur lesquelles il repose ne sont pas fiables.

Dans le cas où $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ n'est pas connu, partant d'une approximation MC de $p_{n-1|n-1}$, il est tout de même possible de déduire une approximation de $p_{n-1|n}$. En effet, si $\{\mathbf{x}_{n-1}^i, w_{n-1}^i\}_{i=1}^N$ approxime $p_{n-1|n-1}$, l'Algorithme 2.1 permet d'obtenir une approximation de $p_{n-1,n|n-1}$ et donc de $p_{n-1|n}$; pour $1 \leq i \leq N$,

1. **P.** Tirer $\mathbf{x}_n^{\text{PS},i} \sim q^{\text{PS}}(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$;
2. **W.** Calculer $w_n^{\text{PS},i} \propto w_{n-1}^i \frac{f_{n|n-1}(\mathbf{x}_n^{\text{PS},i}|\mathbf{x}_{n-1}^i)g_n(\mathbf{y}_n|\mathbf{x}_n^{\text{PS},i})}{q^{\text{PS}}(\mathbf{x}_n^{\text{PS},i}|\mathbf{x}_{n-1}^i)}$;
3. **R.** Tirer $\mathbf{x}_{n-1}^{\text{PS},i} \sim \sum_{i=1}^N w_n^{\text{PS},i} \delta_{\mathbf{x}_{n-1}^i}$.

Par construction, les points $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$ sont donc approximativement tirés suivant $p_{n-1|n}$. Plutôt que d'utiliser le jeu de points $\{\mathbf{x}_n^{\text{PS},i}\}_{i=1}^N$ pour déduire une approximation de $p_{n|n}$, comme le fait l'algorithme SIS/SIR, nous allons tirer un nouveau jeu de points $\{\mathbf{x}_n^i\}_{i=1}^N$, déduit de $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$, en utilisant la formule reliant $p_{n|n}$ et $p_{n-1|n}$, $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n}) p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) d\mathbf{x}_{n-1}$. Cette étape est triviale dès lors qu'il est possible de faire des tirages suivant la loi $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$; dans les autres cas, nous pourrions envisager d'utiliser de nouveau l'échantillonnage d'importance, basé sur une loi $q(\mathbf{x}_n|\mathbf{x}_{n-1})$. Les poids w_n^i associés au tirage de $\mathbf{x}_n^i \sim q(\mathbf{x}_n|\mathbf{x}_{n-1}^{\text{PS},i})$ sont alors donnés par

$$w_n^i = \frac{1}{N} \frac{p(\mathbf{x}_n^i|\mathbf{x}_{n-1}^{\text{PS},i}, \mathbf{y}_n)}{q(\mathbf{x}_n^i|\mathbf{x}_{n-1}^{\text{PS},i})} = \frac{1}{N} \frac{f_{n|n-1}(\mathbf{x}_n^i|\mathbf{x}_{n-1}^{\text{PS},i})g_n(\mathbf{y}_n|\mathbf{x}_n^i)}{p(\mathbf{y}_n|\mathbf{x}_{n-1}^{\text{PS},i})q(\mathbf{x}_n^i|\mathbf{x}_{n-1}^{\text{PS},i})}, \quad (2.3.2)$$

mais ne sont calculables que si l'on dispose de $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$, et donc de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$. Une solution consiste alors à partir d'une approximation $\hat{p}(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ suivant laquelle nous tirons les particules \mathbf{x}_n^i , puis de la combiner avec une méthode MCMC (par exemple, un algorithme de Metropolis-Hasting [58]) pour obtenir des points tirés suivant $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$, qui sont donc associés à un poids $w_n^i = 1/N$. Notons qu'une méthode similaire a été proposée dans [53] : une étape MCMC est introduite après l'étape de rééchantillonnage de l'algorithme SIR (c.f. Algorithme 2.1) afin de diversifier le support des nouveaux points. Ici, les nouveaux

points $\{\mathbf{x}_n^i\}_{i=1}^N$ sont directement construits à partir des points $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$ et sont éventuellement remplacés par ceux de la méthode MCMC, ce qui garantit que tous les points $\{\mathbf{x}_n^i\}_{i=1}^N$ sont différents les uns des autres, comme dans l'algorithme FA. Notre méthodologie peut être résumée par l'algorithme suivant.

Algorithme 2.7: Algorithme PS-APF

Soit $\hat{p}(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{n-1}^i}$ une approximation MC de $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$.

Lissage : obtenir des points $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$ approximativement tirés suivant $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$ en utilisant une itération de l'Algorithme 2.1, avec une distribution d'importance $q^{\text{PS}}(\mathbf{x}_n|\mathbf{x}_{n-1})$;

Filtrage : pour $1 \leq i \leq N$ faire

 | Tirer $\mathbf{x}_n^i \sim q(\mathbf{x}_n|\mathbf{x}_{n-1}^{\text{PS},i})$ et ajuster les tirages à partir d'une méthode MCMC ; poser $w_n^i = \frac{1}{N}$.

fin

Alors $\hat{p}(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_n^i}$ est une approximation MC de $p(\mathbf{x}_n|\mathbf{y}_{0:n})$.

2.3.3 Discussion

Nous pouvons noter que le réglage particulier $q^{\text{PS}}(\mathbf{x}_n|\mathbf{x}_{n-1}) = q(\mathbf{x}_n|\mathbf{x}_{n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ coïncide avec le FA, puisque dans ce cas, $w_n^{\text{PS},i} \propto w_{n-1}^i p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$, et les nouveaux points \mathbf{x}_n^i sont directement tirés suivant $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$; le jeu de points intermédiaire $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$ est par ailleurs inutile puisque les poids intermédiaires $\{w_n^{\text{PS},i}\}$ n'en dépendent pas. En pratique, $q^{\text{PS}}(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $q(\mathbf{x}_n|\mathbf{x}_{n-1})$ doivent idéalement être proches de $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$; mais rappelons que dans ce contexte, ces approximations ne sont utilisées que pour tirer des particules, contrairement à l'APF où $\tau(\mathbf{x}_{0:n-1})$ est utilisé pour approcher analytiquement la fonction $p(\mathbf{y}_n|\mathbf{x}_{n-1})$.

La principale différence entre les algorithmes APF et PS-APF provient du jeu de points intermédiaire : $\{\tilde{\mathbf{x}}_{n-1}^i\}_{i=1}^N$ pour l'APF, et $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$ pour le PS-APF. Dans l'APF, le jeu de points $\{\tilde{\mathbf{x}}_{n-1}^i\}_{i=1}^N$ cible une loi proportionnelle à $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})\tau(\mathbf{x}_{0:n-1})$, qui peut être éloignée de $p_{n-1|n}$ si $\tau(\mathbf{x}_{0:n-1})$ n'est pas choisi correctement ; dans un tel cas, il est possible que des particules $\{\tilde{\mathbf{x}}_{n-1}^i\}_{i=1}^N$ ayant une faible vraisemblance avec la nouvelle observation \mathbf{y}_n soient sélectionnées, ce qui risque de produire par la suite une mauvaise approximation de $p_{n|n}$, malgré la pondération finale visant à corriger cette approximation. À l'opposé, le jeu intermédiaire $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$ de l'algorithme PS-APF cible directement la loi $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n})$ en utilisant l'ensemble $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$ qui explore le nouvel espace d'état à l'instant n afin d'identifier les trajectoires prometteuses. Dans l'algorithme PS-APF, nous concentrons donc nos efforts sur l'étape de sélection des particules prometteuses, et comme nous allons le voir dans les deux prochains paragraphes, cet algorithme s'avère d'intérêt pratique lorsque les approximations classiques utilisées dans l'algorithme APF ne permettent pas d'effectuer cette étape efficacement.

Nous comparons à présent l'algorithme PS-APF avec l'Algorithme 2.2 (APF) ainsi que l'algorithme SIS/SIR, qui correspond à l'Algorithme 2.1.

Remarque 2.6. L'implémentation que nous avons proposée est facilement transposable au cas où nous voudrions implémenter l'Algorithme 2-S (cf. Algorithme 2.6) dans le cas où $p(\mathbf{y}_{n+1}|\mathbf{x}_n)$ serait calculable, mais pas $p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{y}_n) = \int p(\mathbf{y}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)d\mathbf{x}_n$ (voir (2.2.17)). Dans ce cas, nous chercherions à obtenir une approximation de $p_{n-1|n+1}$ à partir d'une étape de lissage, basée sur une itération d'un algorithme d'échantillonnage d'importance, avant de déduire une approximation de $p_{n|n+1}$.

2.3.4 Simulations dans le modèle de Kitagawa

Considérons le modèle

$$\begin{cases} X_n &= h(X_{n-1}) + U_n \\ Y_n &= \frac{X_n^2}{20} + V_n \end{cases} \quad (2.3.3)$$

avec $h(x_n) = 0.5x_n + 25\frac{x_n}{1+x_n^2} + 8 \cos(1.2(n+1))$, où $U_n \sim \mathcal{N}(\cdot; 0; Q)$ et $V_n \sim \mathcal{N}(\cdot; 0; R)$ sont mutuellement indépendants et indépendants de $X_0 \sim \mathcal{N}(\cdot; 0; 1)$.

Paramètres des différents algorithmes

Pour les poids de première espèce de l'APF, Pitt et Shephard [97] ont proposé l'heuristique $\tau(x_{n-1}) = g_n(y_n | \mu_{n|n-1}(x_{n-1}))$ où $\mu_{n|n-1}(x_{n-1})$ est le mode ou la moyenne de la loi $f_{n|n-1}(x_n | x_{n-1})$. Ce choix peut conduire à des estimateurs de mauvaise qualité [6] [30] [41]. En effet, la loi proportionnelle à $p_{n-1|n-1}\tau(\cdot)$ n'a pas de raison d'être proche de $p_{n-1|n}$. Cependant, dans le modèle (2.3.3), les deux premiers moments de la loi $p(x_n, y_n | x_{n-1})$ sont exactement calculables. Il est alors possible de trouver une meilleure approximation de $p(y_n | x_{n-1})$ par la méthode des Moments Exactes (*Exact Matching Moment* = EMM) [110]. Considérons donc l'approximation $\hat{p}(x_n, y_n | x_{n-1})$ de $p(x_n, y_n | x_{n-1})$ donnée par

$$\hat{p}(x_n, y_n | x_{n-1}) = \mathcal{N}\left(x_n, y_n; \left(\frac{h(x_{n-1})}{\frac{h^2(x_{n-1})}{20} + \frac{Q}{20}}\right); \left(\frac{Q}{\frac{h(x_{n-1})Q}{10} + \frac{h^2(x_{n-1})Q}{100} + \frac{Q^2}{200} + R}\right)\right). \quad (2.3.4)$$

De (2.3.4), nous déduisons $\hat{p}(x_n | x_{n-1}, y_n)$, pour tirer de nouvelles particules, et $\phi(x_{n-1}) = \hat{p}(y_n | x_{n-1})$, pour approcher les poids de première espèce. Nous étudions alors les effets de l'approximation (2.3.4) sur les poids de première espèce utilisés dans l'APF. En revanche, il n'est pas évident de comparer directement $\hat{p}(y_n | x_{n-1})$ à $p(y_n | x_{n-1})$ car (2.1.11) n'est pas calculable. Mais $\hat{p}(y_n | x_{n-1})$ peut se réécrire

$$\hat{p}(y_n | x_{n-1}) = \int \hat{p}(y_n | x_n, x_{n-1}) f_{n|n-1}(x_n | x_{n-1}) dx_n \quad (2.3.5)$$

avec $\hat{p}(y_n | x_n, x_{n-1}) = \mathcal{N}(y_n; \tilde{m}_{n|n-1}; \tilde{\sigma}^2)$, $\tilde{m}_{n|n-1} = \frac{h^2(x_{n-1})}{20} + \frac{Q}{20} + \frac{h(x_{n-1})}{10}(x_n - h(x_{n-1}))$ et $\tilde{\sigma}^2 = R + \frac{Q^2}{200}$. Comparons alors le premier facteur dans l'intégrale (2.3.5) à celui dans l'intégrale (2.1.11), c'est à dire $\hat{p}(y_n | x_n, x_{n-1})$ à $g_n(y_n | x_n) = \mathcal{N}(y_n; x_n^2/20; R)$. La vraie moyenne $x_n^2/20$ de $g_n(y_n | x_n)$ est ici approximée par $\tilde{m}_{n|n-1}$, ce qui est correct tant que $x_n \approx h(x_{n-1})$ et $Q/20$ est négligeable. D'autre part, si Q est petit, le second facteur $f_{n|n-1}(x_n | x_{n-1}) = \mathcal{N}(x_n; h(x_{n-1}); Q)$ est à peu près non nul lorsque x_n est dans le voisinage de $h(x_{n-1})$, c'est à dire lorsque $\hat{p}(y_n | x_n, x_{n-1})$ est une bonne approximation de $g_n(y_n | x_n)$. En résumé, $\hat{p}(y_n | x_{n-1})$ est proche de $p(y_n | x_{n-1})$ lorsque Q est petit.

Nous comparons deux variantes de l'APF et trois variantes du PS-APF. Les paramètres des algorithmes APF (poids de première espèce et loi d'importance) et ceux du PS-APF (première et seconde loi d'importance) sont décrits dans la Table 2.2. Pour le PS-APF, nous spécifions également le nombre d'itérations de l'étape MCMC, qui est un algorithme Métropolis-Indépendant [58]. Ces algorithmes sont également comparés avec l'algorithme SIS/SIR du Paragraphe 2.1, dans lequel $q^{\text{SIR}}(x_n | \mathbf{x}_{0:n-1}) = \hat{p}(x_n | x_{n-1}, y_n)$ (l'étape d'estimation est effectuée avant l'étape de rééchantillonnage).

Performance en fonction de Q

Posons $N = 200$ et $R = 1$, puis étudions les performances des algorithmes en fonction de Q . Les résultats sont présentés dans la Table 2.3. Lorsque Q est petit, l'APF et le PS-APF présentent des performances

similaires et sont plus performants que l'algorithme SIR, qui rappelons le, ne sélectionne pas d'anciennes particules prometteuses avant de tirer les nouvelles. Cependant, lorsque Q augmente, les performances des deux algorithmes APF se dégradent et sont dépassées par celles de l'algorithme PS-APF ainsi que celles du SIR ; dans ce cas, le mécanisme de sélection de l'algorithme APF n'est plus efficace, comparé à celui du PS-APF, car l'approximation $\hat{p}(y_n|x_{n-1})$ n'est plus fiable, comme nous l'avons prévu ci-dessus. Notons également que les deux algorithmes PS-APF qui utilisent une approximation de la loi d'importance optimale sont meilleurs que celui qui utilise la loi $f_{n|n-1}(x_n|x_{n-1})$.

Performance en fonction de N

Choisissons maintenant $Q = 10$ et $R = 0.1$ et étudions les performances des algorithmes en fonction de N (voir Figure 2.2). Ces paramètres représentent une difficulté pour les algorithmes SMC classiques. Par exemple, pour l'algorithme SIR, lorsque la distribution optimale d'importance est utilisée, les poids associés aux particules sont réduits à $w_n^{\text{SIR},i} \propto p(y_n|x_{n-1}^i)$. Mais, dans l'algorithme SIR, les particules $\{x_{n-1}^i\}_{i=1}^N$ n'ont aucune raison d'avoir une vraisemblance élevée avec y_n , surtout lorsque R est petit. C'est pour cela que les méthodes auxiliaires sont intéressantes, car elles tentent de sélectionner les anciennes trajectoires en adéquation avec la nouvelle observation. La Figure 2.2 montre que l'algorithme PS-APF présente de meilleures performances que les algorithmes APF et SIR, et semble être efficace même pour un nombre faible de particules. Par exemple, le PS-APF_{EMM-1} avec $N = 200$ particules est meilleur que l'algorithme SIR avec $N = 1000$ particules. L'ordre global entre les algorithmes PS-APF est $\mathcal{J}(\text{PS-APF}_{EMM-1}) \leq \mathcal{J}(\text{PS-APF}_{EMM-0}) \leq \mathcal{J}(\text{PS-APF}_{boot-1})$. L'avantage de PS-APF_{EMM-0} et PS-APF_{EMM-1} par rapport à PS-APF_{boot-1} est qu'ils utilisent une approximation de la loi optimale pour sélectionner et propager les particules ; mais dans le PS-APF_{boot-1}, nous n'avons pas besoin de calculer les deux premiers moments de $p(x_n, y_n|x_{n-1})$, pour chaque particule x_{n-1}^i . Notre prochaine simulation illustre l'intérêt de ce dernier algorithme lorsque le modèle d'espace d'état est plus complexe.

Algorithmes APF	$\tau(x_{n-1})$	$q^{\text{APF}}(x_n x_{n-1})$
APF _{class}	$g_n(y_n \mu_{n n-1}(x_{n-1}))$	$f_{n n-1}(x_n x_{n-1})$
APF _{EMM}	$\hat{p}(y_n x_{n-1})$	$\hat{p}(x_n x_{n-1}, y_n)$
Algorithmes PS-APF	$q^{\text{PS}}(x_n x_{n-1})$	$q(x_n x_{n-1})$
PS-APF _{boot-1}	$f_{n n-1}(x_n x_{n-1})$	$f_{n n-1}(x_n x_{n-1})$
PS-APF _{EMM-0}	$\hat{p}(x_n x_{n-1}, y_n)$	$\hat{p}(x_n x_{n-1}, y_n)$
PS-APF _{EMM-1}	$\hat{p}(x_n x_{n-1}, y_n)$	$\hat{p}(x_n x_{n-1}, y_n)$

TABLE 2.2 – Paramètres des algorithmes APF et PS-APF utilisés pour les simulations dans le modèle (2.3.3). L'indice 0 ou 1 indique le nombre d'itérations MCMC utilisées dans le PS-APF.

2.3.5 Scénario de poursuite angle-distance

Maintenant, nous estimons les positions et vitesses d'un objet ($\mathbf{x}_n = [p_{x,n}, \dot{p}_{x,n}, p_{y,n}, \dot{p}_{y,n}]^T$) à partir d'observations polaires angle-distance :

$$\begin{cases} \mathbf{X}_n &= \mathbf{F}\mathbf{X}_{n-1} + \mathbf{U}_n \\ \mathbf{Y}_n &= [\tan^{-1}(p_y/p_x), (p_x^2 + p_y^2)^{1/2}]^T + \mathbf{V}_n \end{cases} \quad (2.3.6)$$

	$Q = 0.1$	$Q = 1$	$Q = 5$	$Q = 10$
APF_{class}	1.1251	2.4692	4.0857	5.3368
APF_{EMM}	1.1472	2.4848	3.6533	4.4001
$PS-APF_{boot-1}$	1.1493	2.4742	3.6659	4.5044
$PS-APF_{EMM-0}$	1.1227	2.4065	3.6388	4.2872
$PS-APF_{EMM-1}$	1.1309	2.4138	3.6115	4.1834
SIR	1.3653	2.5782	3.6914	4.3388

TABLE 2.3 – Critère \mathcal{J} en fonction de Q - Modèle (2.3.3) - $N = 200$, $R = 1$ - Le nombre d'itérations MCMC de l'algorithme PS-APF est indiqué en indice.

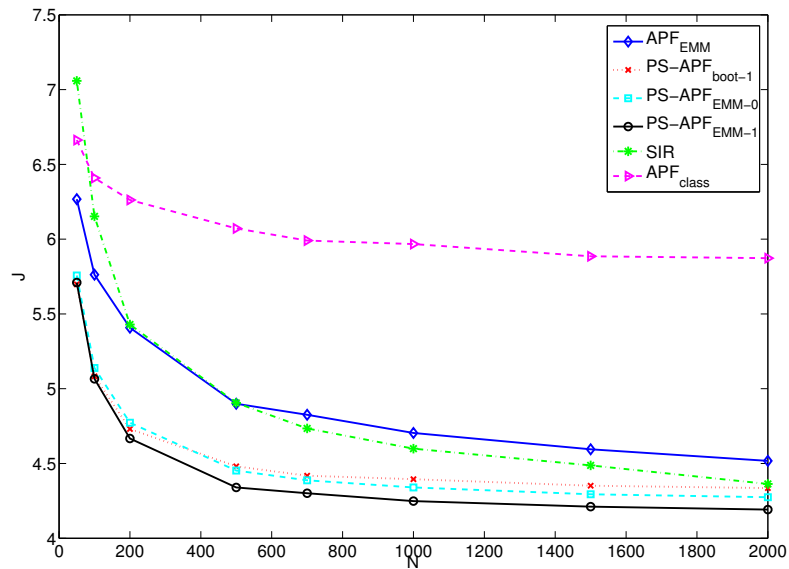


FIGURE 2.2 – Critère \mathcal{J} en fonction de N - Modèle (2.3.3) - $Q = 10$, $R = 0.1$ - Le nombre d'itérations MCMC de l'algorithme PS-APF est indiqué en indice.

où \mathbf{U}_n et \mathbf{V}_n sont des bruits centrés gaussiens indépendants, indépendants de \mathbf{X}_0 , avec $E(\mathbf{U}_n \mathbf{U}_n^T) = \mathbf{Q}$ et $E(\mathbf{V}_n \mathbf{V}_n^T) = \mathbf{R}$. Nous posons $\mathbf{F} = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix}$, $\mathbf{Q} = \mathbf{I}_2 \otimes \sigma_{\mathbf{Q}}^2 \begin{pmatrix} T_s^3/3 & T_s^2/2 \\ T_s^2/2 & T_s \end{pmatrix}$, $\mathbf{R} = \text{diag}(\sigma_b^2, \sigma_r^2)$, où $T_s = 2$ est la période d'échantillonnage, $\sigma_{\mathbf{Q}} = 4$, $\sigma_b = \pi/180$ l'écart type des mesures angulaires et $\sigma_r = 1$ l'écart type des mesures de distance. Comme précédemment, le critère \mathcal{J} est calculé pour chaque composante $(p_x, \dot{p}_x, p_y, \dot{p}_y)$.

Dans ce modèle, les deux premiers moments de $p(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1})$ ne sont pas calculables. L'UKF [119] pourrait être utilisé pour les approcher, mais la méthode est assez intense en coût de calcul car nous devrions exécuter un UKF pour chaque particule [6]. C'est pour cela que nous comparons trois algorithmes qui n'utilisent ni une approximation de $p(\mathbf{y}_n | \mathbf{x}_{n-1})$, ni une approximation de $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$: un algorithme APF avec $\tau(\mathbf{x}_{n-1}) = g_n(\mathbf{y}_n | \mu_{n|n-1}(\mathbf{x}_{n-1}))$ et $q^{\text{APF}}(\mathbf{x}_n | \mathbf{x}_{n-1}) = f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$; un PS-APF avec $q^{\text{PS}}(\mathbf{x}_n | \mathbf{x}_{n-1}) = q(\mathbf{x}_n | \mathbf{x}_{n-1}) = f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$, et une étape MCMC ; un algorithme SIR avec $q^{\text{SIR}}(\mathbf{x}_n | \mathbf{x}_{0:n-1}) = f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$ (les poids associés sont alors donnés par $w_n^{\text{SIR},i} \propto g_n(\mathbf{y}_n | \mathbf{x}_n^i)$, $\sum_{i=1}^N w_n^{\text{SIR},i} = 1$) dans lequel nous ajoutons, après l'étape de rééchantillonnage, une étape MCMC [53].

Les résultats pour les composantes p_x et \dot{p}_x sont donnés dans la Table 2.4. L'étape de lissage de l'algorithme PS-APF est ici crucial, surtout lorsque le nombre de particules est petit et la dimension de l'espace d'état est grande. Supposons en effet que l'état caché de l'objet à l'instant $n - 1$ évolue subitement (virage, accélération, modélisés par une grande valeur de $\sigma_{\mathbf{Q}}$). Puisque l'algorithme SIR ne prend pas en compte la nouvelle observation \mathbf{y}_n pour tirer les nouvelles particules, nous nous attendons à ce que peu de particules contribuent à l'estimation de l'état caché, et que l'estimation finale soit mauvaise. Le problème est contourné avec le PS-APF, puisque pendant l'étape de lissage, nous cherchons les particules \mathbf{x}_{n-1}^i qui ont généré des échantillons $\mathbf{x}_n^{\text{PS},i}$ en accord avec la nouvelle observation \mathbf{y}_n . C'est précisément le rôle de l'étape de lissage de l'algorithme PS-APF. Les particules $\{\mathbf{x}_{n-1}^{\text{PS},i}\}_{i=1}^N$ sont approximativement tirées suivant $p_{n-1|n}$, puis propagées selon $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$. Notons que, même si le nombre de particules différentes $\mathbf{x}_{n-1}^{\text{PS},i}$ est faible, toutes les nouvelles particules \mathbf{x}_n^i sont différentes et contribuent à l'estimation de l'état caché. Le but de l'étape MCMC est d'éliminer les particules \mathbf{x}_n^i qui ne sont pas tirées suivant $p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n)$. Enfin, notons que l'algorithme APF améliore l'estimation de l'état caché comparé à l'algorithme SIR, même lorsque N est faible, grâce à l'étape de sélection qui essaie de garder les particules prometteuses. Cependant, ce choix n'est pas optimal puisque nous avons approximé $p(\mathbf{y}_n | \mathbf{x}_{n-1})$ par la vraisemblance de \mathbf{y}_n avec la moyenne de $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$ (ce choix a néanmoins l'avantage de ne pas nécessiter une approximation de $p(\mathbf{y}_n | \mathbf{x}_{n-1})$).

	$N = 50$	$N = 100$	$N = 500$	$N = 1000$
\mathcal{J}_{p_x} (APF)	86.8539	29.1766	19.0084	18.4816
\mathcal{J}_{p_x} (PS-APF)	76.3198	19.9037	14.4673	13.6303
\mathcal{J}_{p_x} (SIR)	248.6685	105.3927	16.0130	15.3639
$\mathcal{J}_{\dot{p}_x}$ (APF)	7.5691	6.6040	5.7719	5.7336
$\mathcal{J}_{\dot{p}_x}$ (PS-APF)	7.1072	6.2804	5.6106	5.5852
$\mathcal{J}_{\dot{p}_x}$ (SIR)	11.8139	7.8592	5.6952	5.5907

 TABLE 2.4 – Critère \mathcal{J} en fonction de Q - Modèle (2.3.6)

2.3.6 Modèle semi-linéaire

L'implémentation de 2- S repose sur la connaissance de la loi $p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{y}_n)$ qui n'est connue que dans de rares cas car elle repose sur l'expression de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ (voir (2.2.17)). Dans le cas où l'expression de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ est connue, mais pas celle de $p(\mathbf{y}_{n+1}|\mathbf{x}_{n-1}, \mathbf{y}_n)$, il est possible d'implémenter l'algorithme 2- S en utilisant l'interprétation en terme de chemin bayésien $p_{n-1|n} \rightarrow p_{n-1|n+1} \rightarrow p_{n|n+1}$. Nous nous plaçons donc un modèle semi-linéaire dans lequel l'expression de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ est connue, et nous comparons l'implémentation adaptée du PS-APF pour 2- S avec les autres algorithmes de filtrage :

$$\begin{cases} X_n &= \arctan(X_{n-1}) + U_n \\ Y_n &= X_n + V_n \end{cases}, \quad (2.3.7)$$

où U_n et V_n sont des bruits centrés Gaussiens indépendants, indépendants de X_0 , avec $E(U_n^2) = Q$ et $E(V_n^2) = R$. Dans ce modèle, les lois $p(y_n|x_{n-1})$ et $p(x_n|x_{n-1}, y_n)$ sont calculables mais pas $p(y_{n+1}|x_{n-1}, y_n)$ ni $p(x_n|x_{n-1}, y_n, y_{n+1})$. Par conséquent, nous adaptons notre l'algorithme PS-APF au cas où nous souhaitons implémenter le chemin $p_{n-1|n} \rightarrow p_{n-1|n+1} \rightarrow p_{n|n+1}$ et les lois d'importance utilisées sont basées sur une approximation de $p(x_n|x_{n-1}, y_n, y_{n+1}) \propto p(y_n|x_{n-1})p(x_n|x_{n-1}, y_n)$ qui repose sur la transformation sans parfum (*Unscented Transformation* = UT). De plus, notre implémentation approchée de l'algorithme 2- S utilise une étape d'un algorithme Métropolis-Indépendant.

Nous prenons $N = 500$, $R = 2$ et nous calculons \mathcal{J} en fonction de Q . Les résultats sont présentés dans la Table 2.5. Cette implémentation approchée de 2- S semble donner des résultats satisfaisants puisque elle donne de meilleurs résultats que les autres algorithmes. Néanmoins, le coût de calcul requis est plus important car l'étape permettant d'approcher $p_{n-1|n+1}$ de façon similaire au PS-APF implique le tirage de particules supplémentaires. Les algorithmes SIS/SIR et 1- S sont donc, d'un point de vue calculatoire, plus intéressants ici. Nous pouvons remarquer qu'il est difficile ici encore de déterminer lequel des algorithmes SIS/SIR avec densité d'importance conditionnelle optimale, ou 1- S , produit de meilleurs résultats. L'objet de la prochaine section est justement de comparer statistiquement les estimateurs produits par ces algorithmes afin d'être en mesure de prévoir leur comportement en fonction des paramètres du modèle considéré.

	2- P	1- P	SIR	1- S	2- S_{UT-1}
$Q = 1$	0.868658	0.865020	0.863868	0.864456	0.863815
$Q = 2$	1.042373	1.039208	1.038581	1.038466	1.038209
$Q = 5$	1.216360	1.212510	1.207348	1.207879	1.205878
$Q = 10$	1.316713	1.302639	1.299961	1.298394	1.298043

TABLE 2.5 – Critère \mathcal{J} , Modèle semi-linéaire ($N = 500$, $R = 2$)

2.4 Une comparaison théorique des algorithmes SIS/SIR avec densité d'importance conditionnelle optimale et de l'algorithme FA

Nous nous focalisons à présent sur les algorithmes de filtrage direct ainsi que sur l'estimateur $\hat{\Theta}_n$ construit à partir de ces algorithmes. La section précédente laisse apparaître trois solutions pour le problème de l'estimation de la loi $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ et de Θ_n . L'algorithme 1- P = Bootstrap n'utilise pas la nouvelle observation disponible pour tirer des nouveaux points, contrairement à l'algorithme 1- S = FA ; les

algorithmes SIS/SIR peuvent utiliser la nouvelle observation pour la densité d'importance conditionnelle optimale (2.1.10). Par conséquent, il est naturel de se poser la question suivante : quel algorithme est-il préférable d'utiliser lorsque nous souhaitons calculer un estimateur de Θ_n dans (2.0.1), avec $f(\mathbf{x}_{0:n}) = f(\mathbf{x}_n)$? Il s'agit, à première vue, d'un problème compliqué car l'étape de rééchantillonnage de l'algorithme SIS/SIR n'est qu'optionnelle et souvent aléatoire. Cependant, ce problème a été abordé dans plusieurs contributions [63] [19] ; en particulier, la variance des estimateurs considérés a été étudiée de façon asymptotique, dans le cas où les algorithmes sont utilisés de façon séquentielle. Les résultats de ces études laissent apparaître que les performances des estimateurs dépendent du modèle donné, et qu'il n'est pas possible de les comparer aisément.

Dans cette section, plutôt que de considérer la variance asymptotique des estimateurs de Θ_n , nous partons d'une approximation $\hat{p}_{n-1|n-1} = \sum_{i=1}^N w_{n-1}^i \delta_{x_{n-1}^i}$ de $p_{n-1|n-1}$ puis nous faisons une analyse non asymptotique des estimateurs $\hat{\Theta}_n^{\text{SIS}}$, $\hat{\Theta}_n^{\text{SIR}}$ et $\hat{\Theta}_n^{\text{FA}}$ déduits d'une itération des algorithmes SIS, SIR et FA. Comme nous l'avons rappelé dans le Paragraphe 2.1.2, l'estimateur $\hat{\Theta}_n^{\text{SIS}}$, qui est calculé *avant* l'étape de rééchantillonnage est préférable à l'estimateur $\hat{\Theta}_n^{\text{SIR}}$ [108] [19], qui est calculé *après* l'étape de rééchantillonnage, au sens où $\text{var}(\hat{\Theta}_n^{\text{SIS}}) \leq \text{var}(\hat{\Theta}_n^{\text{SIR}})$. Cependant, une comparaison des deux estimateurs nous permettra d'étudier l'impact du rééchantillonnage dans l'algorithme SIS/SIR, en fonction du critère du nombre effectif de particules [72], ainsi que des paramètres sous-jacents du modèle (1.1.1). D'autre part, l'importance de l'ordre des opérations élémentaires soulevée à la Remarque 2.3 (propagation, pondération et rééchantillonnage) sera étudiée à travers la comparaison des estimateurs $\hat{\Theta}_n^{\text{FA}}$ et $\hat{\Theta}_n^{\text{SIR}}$. Enfin, nous étudierons l'impact de l'étape de rééchantillonnage lorsqu'elle est effectuée *avant* l'étape de propagation en comparant les estimateurs $\hat{\Theta}_n^{\text{FA}}$ et $\hat{\Theta}_n^{\text{SIS}}$. À la fin de notre étude, nous proposerons un algorithme SMC minimisant la variance conditionnelle d'un estimateur de Θ_n , à chaque instant.

Dans toute cette section, pour faciliter la présentation des résultats, on suppose les v.a. à valeurs réelles, c'est à dire $X_n \in \mathbb{R}$ et $Y_n \in \mathbb{R}$; cependant, les démonstrations peuvent être adaptées aux cas multidimensionnels. Cette section est organisée de la façon suivante :

- dans le Paragraphe 2.4.1, nous considérons le problème générique de l'obtention de tirages MC suivant un mélange de lois de probabilité $\pi(x)$, puis nous proposons trois estimateurs de $\int f(x)\pi(x)dx$, issus de trois techniques de simulations différentes. Nous comparons les moments d'ordre 1 et 2 des trois estimateurs obtenus ;
- dans le Paragraphe 2.4.2, nous montrons que les trois algorithmes étudiés dans cette section (SIS, SIR et FA) pour déduire un estimateur de Θ_n correspondent chacun à une des trois techniques du Paragraphe 2.4.1 ;
- les propriétés statistiques des estimateurs $\hat{\Theta}_n^{\text{FA}}$, $\hat{\Theta}_n^{\text{SIS}}$ et $\hat{\Theta}_n^{\text{SIR}}$ sont alors déduites du Paragraphe 2.4.1 et présentées dans le Paragraphe 2.4.2, puis discutées dans les Paragraphes 2.4.3-2.4.5 ;
- le Paragraphe 2.4.6 est consacré à la validation des résultats sur des simulations.

2.4.1 Estimateurs MC et mélanges de lois de probabilité

Considérons le mélange de lois de probabilité

$$\pi(x) = \sum_{i=1}^N \alpha_i p_i(x), \quad (2.4.1)$$

avec $\sum_{i=1}^N \alpha_i = 1$, et $p_i(x)$ une densité de probabilité, pour tout i , $1 \leq i \leq N$. Deux mécanismes de simulations peuvent être considérés pour obtenir N' tirages suivant $\pi(\cdot)$ (**dans cette section, N' n'est plus**

nécessairement égal à N).

Méthodes de composition directe et inverse

La méthode de composition directe (voir par exemple [99]), bien connue, produit des échantillons i.i.d. (nous l'avons appelée au Paragraphe 2.2.1). Elle est composée d'une étape discrète (**D.**), consistant à tirer une composante du mélange, et d'une étape continue (**C.**) consistant à tirer suivant la composante choisie. Une méthode moins intuitive consiste à commencer par l'étape continue, avant l'étape discrète. Les deux méthodes sont résumées par les Algorithmes 2.8 et 2.9.

Algorithme 2.8: Méthode de composition directe

```

pour  $j = 1 \dots N'$  faire
  | D. Tirer i.i.d.  $i_j$  selon  $\sum_{i=1}^N \alpha_i \delta_i$ ;
  | C. Tirer  $x^{1,j}$  suivant  $p_{i_j}(\cdot)$ .
fin

```

Algorithme 2.9: Méthode de composition inverse

```

pour  $j = 1 \dots N'$  faire
  | C. Tirer  $\tilde{x}^{2,j}$  suivant  $p_j(\cdot)$ ;
fin
pour  $j = 1 \dots N'$  faire
  | D. Tirer i.i.d.  $i_j$  suivant  $\sum_{i=1}^N \alpha_i \delta_i$ , et poser  $x^{2,j} = \tilde{x}^{2,i_j}$ .
fin

```

Nous comparons alors les N' échantillons produits par chacune des deux méthodes considérées. Dans la proposition suivante, dont une preuve est donnée en Annexe B.1, nous considérons marginalement les points produits par la méthode de composition inverse.

Proposition 2.1. *Les N' échantillons $\{X^{2,j}\}_{j=1}^{N'}$ obtenus par la méthode de composition inverse sont identiquement distribués suivant $\pi(\cdot)$*

La différence entre les méthodes de composition directe et inversée apparaît lorsque nous considérons la loi $p(\mathbf{x}^{1:N'})$ suivie par l'ensemble $\{X^{2,j}\}_{j=1}^{N'}$. Dans le cas de la méthode de composition directe, nous savons que les tirages obtenus sont indépendants. Pour la méthode de composition inversée, nous avons le résultat suivant, dont une preuve est donnée en Annexe B.2

Proposition 2.2. *Notons $X^1 = x^1$ et X^2 les deux premiers points obtenus par la méthode de composition inverse. Pour n'importe quel Borélien $A \in \mathcal{B}(\mathbb{R})$, posons $\mu(A) = \mathbb{P}(X^2 \in A | X^1 = x^1)$. Alors μ est la mesure de probabilité mixte (c'est à dire discrète et continue)*

$$\mu([x, x + dx]) = \alpha \delta_{x^1}([x, x + dx]) + (1 - \alpha) \tilde{\pi}(x) dx, \quad (2.4.2)$$

avec

$$\alpha = \mathbb{P}(X^2 = x^1 | X^1 = x^1) = \frac{\sum_{i=1}^N (\alpha_i)^2 p_i(x^1)}{\sum_{i=1}^N \alpha_i p_i(x^1)}, \quad (2.4.3)$$

$$\tilde{\pi}(x) = \sum_{i=1}^N \beta_i p_i(x), \quad (2.4.4)$$

$$\beta_i \propto \alpha_i \sum_{\substack{j=1 \\ j \neq i}}^N \alpha_j p_j(x^1), \quad \sum_{i=1}^N \beta_i = 1. \quad (2.4.5)$$

Remarque 2.7. La méthode de composition inverse commence par tirer des points $\{\tilde{x}^{2,i}\}_{i=1}^N$ suivant chacune des composantes du mélange, les pondère et finalement tire N' nouveaux points $\{x^{2,i}\}_{i=1}^N$ de façon indépendante, conditionnellement à $\tilde{\mathbf{X}}^{2,1:N} = \tilde{\mathbf{x}}^{2,1:N}$. Ces N' points, i.i.d. suivant $\sum_{i=1}^N \alpha_i \delta_{\tilde{x}^i}$, ne sont plus indépendants lorsque le conditionnement par rapport à $\tilde{\mathbf{x}}^{2,1:N}$ n'est plus considéré. Dès que la première particule x^1 a été tirée, la seconde X^2 peut être égale à x^1 avec une probabilité α ou peut prendre n'importe quelle valeur selon le mélange $\tilde{\pi}(\cdot)$ défini en (2.4.4)-(2.4.5).

Notons que $\tilde{\pi}(\cdot)$ est une version repondérée de $\pi(\cdot)$ avec les poids β_i . Considérons le cas où les composantes $\{p_i(\cdot)\}_{i=1}^N$ sont bien séparées. Si le point x^1 a été tiré selon $p_{i_0}(\cdot)$, les valeurs de $p_j(x^1)$, pour $j \neq i_0$ sont très petites et donc β_{i_0} sera très petit. Dans ce cas, les poids β_i ont tendance à écarter le voisinage du premier point x^1 . En définitive, X^2 peut être égale à x^1 , ou est éventuellement tirée suivant n'importe quelle composante du mélange $\tilde{\pi}(\cdot)$, sauf $p_{i_0}(\cdot)$.

Estimateurs MC associés

Supposons que nous souhaitons à présent calculer un estimateur de $E_\pi(f(X)) = \int_{\mathbb{R}} f(x)\pi(x)dx$ à partir de tirages selon $\pi(\cdot)$. Les deux techniques évoquées ci-dessus permettent de construire les trois estimateurs suivants (qui utilisent soit les N' points $\mathbf{X}^{1,1:N'}$, soit les N points $\tilde{\mathbf{X}}^{2,1:N}$, soit les N' points $\mathbf{X}^{2,1:N'}$) :

$$\hat{\Theta}^{1,N'}(\mathbf{X}^{1,1:N'}) = \frac{1}{N'} \sum_{i=1}^{N'} f(X^{1,i}), \quad (2.4.6)$$

$$\tilde{\Theta}^2(\tilde{\mathbf{X}}^{2,1:N}) = \sum_{i=1}^N \alpha_i f(\tilde{X}^{2,i}), \quad (2.4.7)$$

$$\hat{\Theta}^{2,N'}(\mathbf{X}^{2,1:N'}) = \frac{1}{N'} \sum_{i=1}^{N'} f(X^{2,i}). \quad (2.4.8)$$

La proposition suivante, dont une preuve est donnée en Annexe B.3, donne la moyenne et la variance respective de chacun de ces trois estimateurs.

Proposition 2.3. Soit $\pi(\cdot)$ le mélange de lois de probabilité (2.4.1) et soient les estimateurs $\hat{\Theta}^{1,N'}$, $\tilde{\Theta}^2$ et $\hat{\Theta}^{2,N'}$ donnés par (2.4.6)-(2.4.8). Alors $E(\hat{\Theta}^{1,N'}) = E(\tilde{\Theta}^2) = E(\hat{\Theta}^{2,N'}) = E_\pi(f(X))$ et

$$\text{var}(\hat{\Theta}^{1,N'}) = \frac{1}{N'} \text{var}_\pi(f(X)), \quad (2.4.9)$$

$$\text{var}(\tilde{\Theta}^2) = \sum_{i=1}^N \alpha_i^2 \text{var}_{p_i}(f(X)), \quad (2.4.10)$$

$$\text{var}(\hat{\Theta}^{2,N'}) = \frac{1}{N'} \text{var}_\pi(f(X)) + \frac{N' - 1}{N'} \sum_{i=1}^N \alpha_i^2 \text{var}_{p_i}(f(X)). \quad (2.4.11)$$

À présent, nous allons voir comment ces résultats s'appliquent lorsque nous souhaitons comparer les estimateurs issus des algorithmes SIS, SIR et FA.

2.4.2 Algorithmes localement optimaux SIS, SIR et FA

Les algorithmes SIS, SIR et FA permettent de construire un estimateur $\hat{\Theta}_n$ de Θ_n dans (2.0.4). Rappelons brièvement leur principe.

- Les algorithmes SIS / SIR ont été présentés dans la Section 2.1.1 (cf. Algorithme 2.1). Dans le cas localement optimal, partant d'une approximation discrète $\{x_{n-1}^i, w_{n-1}^i\}_{i=1}^N$, les particules sont d'abord tirées suivant $p(x_n|x_{n-1}^i, y_n)$ puis pondérées par

$$w_n^i \propto w_{n-1}^i p(y_n|x_{n-1}^i), \quad \sum_{i=1}^N w_n^i = 1. \quad (2.4.12)$$

Cette étape de pondération est éventuellement suivie d'une étape de rééchantillonnage.

- Dans l'algorithme FA, partant d'une approximation discrète $\{x_{n-1}^i, w_{n-1}^i\}_{i=1}^N$, les particules sont tirées suivant le mélange

$$\pi_{n|n}(x_n) = \sum_{i=1}^N w_{n-1}^i p(x_n|x_{n-1}^i, y_n), \quad (2.4.13)$$

$$w_n^i \propto p(y_n|x_{n-1}^i) w_{n-1}^i, \quad \sum_{i=1}^N w_n^i = 1. \quad (2.4.14)$$

Ces trois algorithmes sont, en somme, assez proches les uns des autres car tous font intervenir les lois $p(x_n|x_{n-1}, y_n)$ et $p(y_n|x_{n-1})$. Remarquons que dans les algorithmes SIS/SIR conditionnellement optimaux, l'étape de pondération et l'étape de propagation peuvent être inversées car les poids ne dépendent pas des nouveaux points tirés (cf. la discussion du Paragraphe 2.1.2). On remarque alors que l'algorithme SIR implémente la méthode de composition inverse pour tirer des points suivant $\pi_{n|n}$. Dans l'algorithme FA (cf. Algorithme 2.4), les points sont tirés suivant $\pi_{n|n}$ par la méthode de composition directe. Finalement, les trois algorithmes se résument à un unique algorithme, l'Algorithme 2.10.

Comme nous venons de le voir, les algorithmes SIR et FA exécutent les mêmes opérations mais dans un ordre différent. La principale différence provient de l'étape de rééchantillonnage, effectuée *avant* l'étape de tirage pour l'algorithme FA, et *après* l'étape de tirage pour l'algorithme SIR. Intéressons nous donc aux effets de l'ordre du rééchantillonnage.

Observons dans un premier temps la taille du support des points obtenus après une itération d'un des deux algorithmes (FA ou SIR). Comme nous l'avons rappelé dans le Paragraphe 2.1.2, le but de l'étape de rééchantillonnage est de lutter contre le phénomène de dégénérescence des poids. Cependant, comme dans l'algorithme SIR un point peut être retiré plusieurs fois, $\{x_n^{\text{SIR},i}\}_{i=1}^{N'}$ est un sous ensemble de $\{x_n^{\text{SIS},i}\}_{i=1}^N$. Le support de la distribution diminue donc localement, ce qui est une forme de dégénérescence. Cette diminution du support peut être mesurée par la taille moyenne du support après l'étape de rééchantillonnage $E(\nu')$, définie comme l'espérance du nombre de points qui ont été sélectionnés au moins une fois. On a

$$E(\nu') = N - \sum_{i=1}^N (1 - w_n^i)^{N'}, \quad (2.4.15)$$

puisque le nombre de tirages des particules $\{x_n^{\text{SIS},i}\}_{i=1}^N$ suit une loi multinomiale de paramètres $\{w_n^i\}_{i=1}^N$. Cette dégénérescence du support ne se produit pas dans l'algorithme FA ; bien que l'étape de rééchantillonnage introduise une diminution du support ($\{\tilde{x}_{n-1}^i\}_{i=1}^{N'}$ est un sous-ensemble de $\{x_{n-1}^i\}_{i=1}^N$), l'étape de tirage suivante assure que les points $\{x_n^{\text{FA},i}\}$ sont différents, même si deux points \tilde{x}_{n-1}^i et \tilde{x}_{n-1}^j sont identiques.

Algorithme 2.10: Algorithme SIS/SIR optimaux et algorithme FA.

 Soit $\hat{p}_{n-1|n-1} = \sum_{i=1}^N w_{n-1}^i \delta_{x_{n-1}^i}$ une approximation MC de $p_{n-1|n-1}$.

pour $1 \leq i \leq N$ **faire**

 | **W.** Calculer $w_n^i \propto w_{n-1}^i p(y_n | x_{n-1}^i)$, $\sum_{i=1}^N w_n^i = 1$
fin
si Algorithme SIS/SIR avec densité conditionnelle optimale **alors**

 | **pour** $1 \leq i \leq N$ **faire**

 | | **P.** Tirer $x_n^{\text{SIS},i} \sim p(x_n | x_{n-1}^i, y_n)$;

 | **fin**

 | **pour** $1 \leq i \leq N'$ **faire**

 | | **R.** Tirer $x_n^{\text{SIR},i} \sim \sum_{j=1}^N w_n^j \delta_{x_n^{\text{SIS},j}}$.

 | **fin**
fin
si Algorithme FA **alors**

 | **pour** $1 \leq i \leq N'$ **faire**

 | | **R.** Tirer $\tilde{x}_{n-1}^i \sim \sum_{j=1}^N w_n^j \delta_{x_{n-1}^j}$;

 | | **P.** Tirer $x_n^{\text{FA},i} \sim p(x_n | \tilde{x}_{n-1}^i, y_n)$.

 | **fin**
fin

 Alors $\sum_{i=1}^N w_n^i \delta_{x_n^{\text{SIS},i}}$, $\sum_{i=1}^{N'} \frac{1}{N'} \delta_{x_n^{\text{SIR},i}}$ et $\sum_{i=1}^{N'} \frac{1}{N'} \delta_{x_n^{\text{FA},i}}$ sont des approximations de p_n .

Intéressons nous maintenant aux considérations statistiques. En effet, les trois algorithmes SIS, SIR et FA permettent de construire les trois estimateurs ponctuels $\hat{\Theta}_n^{\text{SIS}}$, $\hat{\Theta}_n^{\text{SIR}}$ et $\hat{\Theta}_n^{\text{FA}}$ de Θ_n :

$$\hat{\Theta}_n^{\text{SIS}} = \sum_{i=1}^N w_n^i f(X_n^{\text{SIS},i}), \quad (2.4.16)$$

$$\hat{\Theta}_n^{\text{SIR},N'} = \frac{1}{N'} \sum_{i=1}^{N'} f(X_n^{\text{SIR},i}), \quad (2.4.17)$$

$$\hat{\Theta}_n^{\text{FA},N'} = \frac{1}{N'} \sum_{i=1}^{N'} f(X_n^{\text{FA},i}). \quad (2.4.18)$$

À présent, nous cherchons à établir les propriétés statistiques de ces estimateurs, puis à les comparer deux à deux. D'après la Proposition 2.3, nous avons le résultat suivant :

Théorème 2.1. Soient $\hat{\Theta}_n^{\text{SIS}}$, $\hat{\Theta}_n^{\text{SIR},N'}$ et $\hat{\Theta}_n^{\text{FA},N'}$ les estimateurs définis en (2.4.16)-(2.4.18). Notons $\hat{\Theta}_n^{\text{SIR},N} = \hat{\Theta}_n^{\text{SIR}}$, $\hat{\Theta}_n^{\text{FA},N} = \hat{\Theta}_n^{\text{FA}}$ et soit $\pi_{n|n}(\cdot)$ le mélange (2.4.13). Alors $E(\hat{\Theta}_n^{\text{FA},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) = E(\hat{\Theta}_n^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) = E(\hat{\Theta}_n^{\text{SIR},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) = E_{\pi_{n|n}}(f(X))$, et

$$\text{var}(\hat{\Theta}_n^{\text{FA},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) = \frac{1}{N'} \text{var}_{\pi_{n|n}}(f(X)), \quad (2.4.19)$$

$$\text{var}(\hat{\Theta}_n^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) = \sum_{i=1}^N (w_n^i)^2 \text{var}_{p(x_n | x_{n-1}^i, y_n)}(f(X)), \quad (2.4.20)$$

$$\text{var}(\hat{\Theta}_n^{\text{SIR},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) = \frac{1}{N'} \text{var}_{\pi_{n|n}}(f(X)) + \frac{N' - 1}{N'} \sum_{i=1}^N (w_n^i)^2 \text{var}_{p(x_n | x_{n-1}^i, y_n)}(f(X)). \quad (2.4.21)$$

Les trois estimateurs $\hat{\Theta}_n^{\text{SIS}}$, $\hat{\Theta}_n^{\text{SIR},N'}$ et $\hat{\Theta}_n^{\text{FA},N'}$ ayant même moyenne, nous comparons leur variance conditionnelle à $\{x_{n-1}^i, w_{n-1}^i\}_{i=1}^N$ et y_n fixés.

2.4.3 Comparaison de $\hat{\Theta}_n^{\text{SIR},N'}$ et de $\hat{\Theta}_n^{\text{FA},N'}$

D'après (2.4.19) et (2.4.21), nous avons pour tout N et N' ,

$$\text{var}(\hat{\Theta}_n^{\text{SIR},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) \geq \text{var}(\hat{\Theta}_n^{\text{FA},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n). \quad (2.4.22)$$

Partant d'une mesure empirique $\hat{p}_{n-1|n-1}$ et après avoir calculé les poids w_n^i , il est donc préférable de rééchantillonner l'ancien jeu de points, puis de tirer le nouveau, plutôt que de tirer un nouveau jeu de points puis de le rééchantillonner.

Remarque 2.8. Ce résultat est facile à comprendre dès lors que l'on réécrit l'algorithme SIR avec densité d'importance conditionnelle optimale. Puisque l'étape de rééchantillonnage est effectuée à partir de poids qui ne dépendent pas des particules $\{x_n^{\text{SIS},i}\}_{i=1}^N$ tirées, cette étape peut être effectuée avant l'étape de tirage, à condition de ne tirer les nouvelles particules que lorsque c'est nécessaire. On obtient donc une formulation alternative de l'algorithme :

W. Pour $1 \leq j \leq N$, calculer $w_n^j \propto p(y_n | x_{n-1}^j) w_{n-1}^j$, $\sum_{j=1}^N w_n^j = 1$;

R. Pour $1 \leq j \leq N'$, sélectionner un index $i_j \sim \sum_{i=1}^N w_n^i \delta_i$;

P. Pour $1 \leq j \leq N'$,

- Si $i_j \notin \{i_1, \dots, i_{j-1}\}$ alors tirer $x_n^{\text{SIS},j} \sim p(x_n | x_{n-1}^{i_j}, y_n)$, et poser $(x_n^{\text{SIR},j}, w_n^j) = (x_n^{\text{SIS},j}, 1/N')$;
- Si $i_j = i_k$ pour $i_k \in \{i_1, \dots, i_{j-1}\}$ alors poser $(x_n^{\text{SIR},j}, w_n^j) = (x_n^{\text{SIR},k}, 1/N')$.

Cette implémentation est évidemment plus intéressante d'un point de vue calculatoire car les particules $x_n^{\text{SIS},j}$ qui doivent être éliminées à l'étape de rééchantillonnage ne sont plus tirées. Cette réécriture illustre également la différence avec l'algorithme FA : dans l'algorithme SIR, la composante i_j tirée est représentée par une seule particule, tandis que dans l'algorithme FA, cette composante est représentée par N_{i_j} particules, où N_{i_j} est le nombre de fois où cette composante a été tirée.

D'après (2.4.19) et (2.4.21), ce manque de diversité est mesuré par le second terme $\delta(N')$ du membre de droite de (2.4.21), qui dépend des paramètres $\{\text{var}_{p(x_n | x_{n-1}^i, y_n)}(f(X))\}_{i=1}^N$ mais aussi du nombre de particules effectives $N_{\text{eff}}(n)$ (qui est un estimateur de la taille effective du support de l'approximation discrète) après la mise à jour des poids, introduit dans [72] (voir aussi [108], [78] et [31]) et défini par

$$N_{\text{eff}}(n) \stackrel{\text{def}}{=} \frac{1}{\sum_{i=1}^N (w_n^i)^2}. \quad (2.4.23)$$

Sur un plan pratique, si $\text{var}_{p(x_n | x_{n-1}^i, y_n)}(f(X))$ est petit pour tout i , alors les N_{i_j} points $f(x_n^{\text{FA},k})$ associés à l'index i_j sont presque identiques et $\delta(N')$ est petit. Si $\text{var}_{p(x_n | x_{n-1}^i, y_n)}(f(X))$ est quasi constant (égal à γ) alors $\delta(N') \approx \frac{\gamma}{N_{\text{eff}}(n)}$ donc $\delta(N')$ croît lorsque $N_{\text{eff}}(n)$ décroît, et en effet, lorsque $N_{\text{eff}}(n)$ est petit, l'algorithme SIR produit très peu de points différents alors que l'algorithme FA en produit toujours N' . Mais cette différence devient insignifiante lorsque les points $f(x_n^{\text{FA},k})$ ne sont pas séparés, c'est à dire lorsque le numérateur γ est petit.

Remarque 2.9. Nous avons vu que l'estimateur basé sur l'algorithme FA était préférable à l'estimateur basé sur l'algorithme SIR, quels que soient N et N' . Notons cependant que pour produire $N' = N + K$ points suivant $\pi_{n|n}(\cdot)$, l'algorithme SIR effectue $2N + K$ tirages tandis que l'algorithme FA en effectue $2N + 2K$. Comparons alors les deux estimateurs à coût fixe, c'est à dire pour un nombre de tirages donné, $2N + 2K$;

nous comparons ainsi $\widehat{\Theta}_n^{\text{FA},N+K}$ avec $\widehat{\Theta}_n^{\text{SIR},N+2K}$, pour une valeur donnée de N et pour $K > -N/2$. Posons alors

$$\begin{aligned} \delta'(K) &\stackrel{\text{def}}{=} \text{var}(\widehat{\Theta}_n^{\text{SIR},N+2K} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) - \text{var}(\widehat{\Theta}_n^{\text{FA},N+K} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) \\ &= \frac{\text{var}(\widehat{\Theta}_n^{\text{FA}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)}{(N+2K)(N+K)} \times a(K) \end{aligned}$$

avec $a(K) = -NK + (N+K)(N+2K-1)A_n$ et $A_n = \frac{\text{var}(\widehat{\Theta}_n^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)}{\text{var}(\widehat{\Theta}_n^{\text{FA}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)}$. Le signe de $\delta'(K)$ coïncide avec celui du polynôme $a(K)$, qui est positif dès lors que $-\frac{N}{2} < K \leq 0$. Supposons que $K > 0$. Si $A_n \geq 1$, alors $a(K) \geq N^2 + (N+K)(2K-1) \geq 0$. Finalement, pour $N > 1$ et $K > -\frac{N}{2}$, $a(K)$, et donc $\delta'(K)$, est négatif si et seulement si $A_n \leq \alpha_1(N) = \frac{N(3N-1) - 2N(2N(N-1))^{\frac{1}{2}}}{(N+1)^2}$ et $K_1(A_n, N) \leq K \leq K_2(A_n, N)$, avec

$$K_i(A_n, N) = \frac{-A_n(3N-1) + N + \epsilon_i(A_n^2(N+1)^2 - 2A_nN(3N-1) + N^2)^{\frac{1}{2}}}{4A_n},$$

$\epsilon_1 = 1$ et $\epsilon_2 = -1$.

Commentons ce résultat. Notons d'abord que pour de grandes valeurs de N , $\alpha_1(N) \approx 3 - \sqrt{8} \approx 0.1716$. Par conséquent, $\delta'(K)$ peut être négatif seulement si l'estimateur $\widehat{\Theta}_n^{\text{SIS}}$, basé sur le tirage de N points, est beaucoup plus performant que l'estimateur $\widehat{\Theta}_n^{\text{FA}}$ lui aussi basé sur N points. Bien qu'il s'agisse d'une condition forte, nous verrons qu'il existe des cas de figure dans lesquels cette situation peut se produire (voir par exemple la Figure 2.7 dans le Paragraphe 2.4.6). Prenons par exemple $N = 1000$ et $A_n = 0.16$. Nous obtenons $K_1(A, N) \approx 412$ et $K_2(A, N) \approx 1214$, ce qui signifie que l'estimateur $\widehat{\Theta}_n^{\text{SIR},N+2K}$ est préférable à l'estimateur $\widehat{\Theta}_n^{\text{FA},N+K}$ pour des valeurs très particulières de K .

Remarque 2.10. Notons

$$J_n^{N'} = \text{E}((\widehat{\Theta}_n^{N'} - \Theta_n)^2 | \mathbf{y}_{0:n}) \quad (2.4.24)$$

l'EQM (conditionnelle à $\mathbf{y}_{0:n}$) de l'estimateur $\widehat{\Theta}_n^{N'}$, où $\widehat{\Theta}_n^{N'}$ représente $\widehat{\Theta}_n^{\text{FA},N'}$ ou $\widehat{\Theta}_n^{\text{SIR},N'}$. D'après le Théorème 2.1, $\text{E}((\widehat{\Theta}_n^{N'} - \Theta_n)^2 | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, \mathbf{y}_{0:n}) = \text{var}(\widehat{\Theta}_n^{N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, \mathbf{y}_n) + (\text{E}_{\pi_n} (f(X)) - \Theta_n)^2$. En appliquant l'espérance des deux côtés de l'égalité, nous avons

$$J_n^{N'} = \text{E}(\text{var}(\widehat{\Theta}_n^{N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, \mathbf{y}_{0:n}) | \mathbf{y}_{0:n}) + \text{E}([\text{E}_{\pi_n} (f(X)) - \Theta_n]^2 | \mathbf{y}_{0:n}).$$

Le second terme étant constant, d'après (2.4.22), nous obtenons $J_n^{\text{SIR},N'} > J_n^{\text{FA},N'}$.

2.4.4 Comparaison de $\widehat{\Theta}_n^{\text{SIR},N'}$ et $\widehat{\Theta}_n^{\text{SIS}}$

Bien que (2.4.21) décompose $\text{var}(\widehat{\Theta}_n^{\text{SIR},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)$ comme la somme de deux termes positifs, cette égalité ne doit pas être confondue avec l'égalité RB (1.3.1) qui s'écrit ici

$$\text{var}(\widehat{\Theta}_n^{\text{SIR},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) = \text{var}(\widehat{\Theta}_n^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) + \Delta_n, \quad (2.4.25)$$

où

$$\Delta_n = \text{E}((\widehat{\Theta}_n^{\text{SIR},N'} - \widehat{\Theta}_n^{\text{SIS}})^2 | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) \geq 0 \quad (2.4.26)$$

et correspond donc à la variance induite par l'étape de rééchantillonnage. Cette égalité permet de conclure, ainsi qu'il est bien connu [108, Section 4.2.1] [19, p. 213], que

$$\text{var}(\widehat{\Theta}_n^{\text{SIR},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) \geq \text{var}(\widehat{\Theta}_n^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n), \quad (2.4.27)$$

et donc que $J_n^{\text{SIR},N'} > J_n^{\text{SIS}}$. Par conséquent, il est préférable d'utiliser l'estimateur basé sur l'ensemble pondéré $\{x_n^{\text{SIS},i}, w_n^i\}_{i=1}^N$ plutôt que celui basé sur $\{(x_n^{\text{SIR},i}, 1/N')\}_{i=1}^{N'}$. En résumé, le rééchantillonnage introduit un coût de calcul supplémentaire, une diminution du support des points et un terme de variance supplémentaire dans l'expression de l'estimateur ponctuel final.

Les bienfaits éventuels du rééchantillonnage n'interviennent donc qu'aux itérations suivantes. Cependant, notons que le risque de cette étape est d'éliminer de bonnes particules, ce risque étant d'autant plus élevé que l'entropie des poids est grande. Le terme de variance supplémentaire Δ_n introduit par l'étape de rééchantillonnage serait alors d'autant plus important que $N_{eff}(n)$, défini en (2.4.23), est grand. Même si en toute rigueur, ce n'est pas le cas, nous allons voir que Δ_n permet de mesurer l'impact du rééchantillonnage. Nous pouvons invoquer deux arguments.

- Considérons le cas $N' = N$ et étudions le risque d'éliminer des particules de poids importants. Supposons que le support avant rééchantillonnage est réduit à \tilde{N}_{eff} particules de poids non nuls et que la masse totale est distribuée sur ces \tilde{N}_{eff} particules. (2.4.15) devient alors

$$E(\nu') = \tilde{N}_{eff} - \tilde{N}_{eff} \left(1 - \frac{1}{\tilde{N}_{eff}}\right)^N. \quad (2.4.28)$$

Posons $h(k) = k(1 - \frac{1}{k})^N$, qui est une fonction croissante. Le nombre moyen de particules éliminées pendant l'étape de rééchantillonnage augmente donc lorsque \tilde{N}_{eff} croît. Nous avons tracé sur la Figure 2.3 $E(\nu')$ en fonction de \tilde{N}_{eff} :

- si $\tilde{N}_{eff} = 1$, alors la particule est nécessairement retirée et les estimateurs $\widehat{\Theta}_n^{\text{SIR}}$ et $\widehat{\Theta}_n^{\text{SIS}}$ sont identiques ;
- $h'(\cdot)$ a un point d'inflexion en $\tilde{N}_{eff} = \frac{N+1}{3}$: tant que $\tilde{N}_{eff} \leq \frac{N+1}{3}$, le nombre moyen de particules perdues après rééchantillonnage est faible, mais devient plus important après ce point ;
- Enfin, si $\tilde{N}_{eff} = N$, $E(\nu') \approx \frac{2N}{3}$, ce qui signifie qu'un tiers des particules sont perdues.

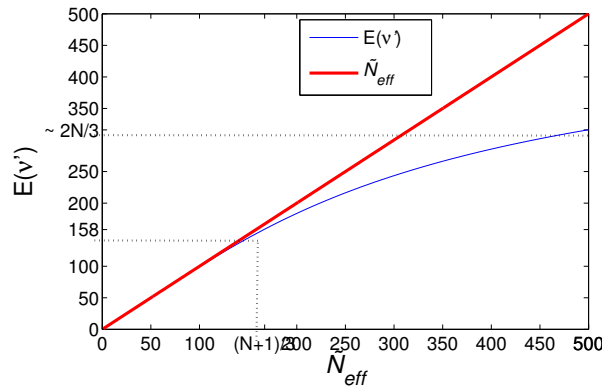


FIGURE 2.3 – Taille du support après rééchantillonnage, en fonction de \tilde{N}_{eff} . Ici, $N = 500$.

- Étudions maintenant l'impact de N_{eff} sur le terme Δ_n . Dans les cas où les particules sont à peu près pondérées de façon égale, la diminution du support est sévère ; on s'attend donc à ce que la différence

de variance Δ_n dans (2.4.25) soit significative. A l'opposé, si l'on exécute plusieurs itérations de l'algorithme SIS, sans rééchantillonnage, les poids dégènèrent et $N_{eff}(n)$ décroît ; nous nous attendons alors à ce que Δ_n diminue également car dans ce cas le rééchantillonnage est bénéfique pour les étapes suivantes. Cependant, Δ_n ne dépend pas seulement de $N_{eff}(n)$ mais aussi du modèle HMC et de la fonction f . Nous pouvons dans un premier temps réécrire

$$\Delta_n = A_n + B_n, \quad (2.4.29)$$

$$A_n = \sum_{i=1}^N \frac{1}{N'} w_n^i (1 - w_n^i) \text{var}_{p(x_n|x_{n-1}^i, y_n)}(f(X)), \quad (2.4.30)$$

$$B_n = \frac{1}{N'} \sum_{1 \leq i < j \leq N} w_n^i w_n^j \left[\mathbb{E}_{p(x_n|x_{n-1}^i, y_n)}(f(X)) - \mathbb{E}_{p(x_n|x_{n-1}^j, y_n)}(f(X)) \right]^2. \quad (2.4.31)$$

Démonstration. D'après (2.4.20) et (2.4.21),

$$\Delta_n = \frac{1}{N'} \text{var}_{\pi_n|n}(f(X)) - \frac{1}{N'} \sum_{i=1}^N (w_n^i)^2 \text{var}_{p(x_n|x_{n-1}^i, y_n)}(f(X)). \quad (2.4.32)$$

D'après (2.4.13),

$$\begin{aligned} \text{var}_{\pi_n|n}(f(X)) &= \sum_{i=1}^N w_n^i \left[\text{var}_{p(x_n|x_{n-1}^i, y_n)}(f(X)) + (\mathbb{E}_{p(x_n|x_{n-1}^i, y_n)}(f(X)))^2 \right] - \\ &\quad \left[\sum_{i=1}^N w_n^i \mathbb{E}_{p(x_n|x_{n-1}^i, y_n)}(f(X)) \right]^2. \end{aligned} \quad (2.4.33)$$

Nous obtenons donc $\Delta_n = A_n + B_n$, avec A_n donné par (2.4.30), et

$$B_n = \frac{1}{N'} \left(\sum_{i=1}^N w_n^i \left[\mathbb{E}_{p(x_n|x_{n-1}^i, y_n)}(f(X)) \right]^2 - \left[\sum_{i=1}^N w_n^i \mathbb{E}_{p(x_n|x_{n-1}^i, y_n)}(f(X)) \right]^2 \right). \quad (2.4.34)$$

En multipliant le premier terme de (2.4.34) par $\sum_{j=1}^N w_n^j$ (qui est égal à 1), puis en réarrangeant les termes de la somme, nous obtenons (2.4.31). \square

La Proposition 2.4 illustre l'impact sur Δ_n des termes de variance dans (2.4.30) et des différences quadratiques dans (2.4.31).

Proposition 2.4. *Soient*

$$p = \min_{i=1 \dots N} \text{var}_{p(x_n|x_{n-1}^i, y_n)}(f(X)), \quad (2.4.35)$$

$$P = \max_{i=1 \dots N} \text{var}_{p(x_n|x_{n-1}^i, y_n)}(f(X)), \quad (2.4.36)$$

$$m = \min_{1 \leq i < j \leq N} \left[\mathbb{E}_{p(x_n|x_{n-1}^i, y_n)}(f(X)) - \mathbb{E}_{p(x_n|x_{n-1}^j, y_n)}(f(X)) \right]^2, \quad (2.4.37)$$

$$M = \max_{1 \leq i < j \leq N} \left[\mathbb{E}_{p(x_n|x_{n-1}^i, y_n)}(f(X)) - \mathbb{E}_{p(x_n|x_{n-1}^j, y_n)}(f(X)) \right]^2. \quad (2.4.38)$$

Alors

$$\frac{1}{N'} \left(1 - \frac{1}{N_{eff}(n)} \right) \left(p + \frac{m}{2} \right) \leq \Delta_n \leq \frac{1}{N'} \left(1 - \frac{1}{N_{eff}(n)} \right) \left(P + \frac{M}{2} \right). \quad (2.4.39)$$

2.4 Comparaison théorique d'algorithmes de filtrage localement optimaux

Démonstration. Soient $S_n = \sum_{i=1}^N \frac{1}{N'} w_n^i (1 - w_n^i)$ et $S'_n = \frac{1}{N'} \sum_{1 \leq i < j \leq N} w_n^i w_n^j$. On rappelle que $\sum_{i=1}^N w_n^i = 1$ et que $1 / \sum_{i=1}^N (w_n^i)^2 = N_{eff}(n)$. Donc $S_n = \frac{1}{N'} (1 - \frac{1}{N_{eff}(n)})$, et

$$S'_n = \frac{1}{2N'} \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N w_n^i w_n^j = \frac{1}{2N'} \left(\left[\sum_{i=1}^N w_n^i \right]^2 - \sum_{i=1}^N (w_n^i)^2 \right) = \frac{1}{2N'} \left(1 - \frac{1}{N_{eff}(n)} \right).$$

En injectant ce résultat dans (2.4.29), et en utilisant (2.4.35)-(2.4.38), nous obtenons (2.4.39). \square

Remarque 2.11. Ce résultat permet d'étudier la variation du terme Δ_n en fonction de $N_{eff}(n)$ lorsque les paramètres p , P , m et M sont fixés, ou d'étudier la variation de Δ_n en fonction de ces paramètres pour une valeur de $N_{eff}(n)$ donnée.

Si $p \approx P$, et M est négligeable comparé à p (et donc m l'est aussi), alors les deux bornes dans (2.4.39) sont presque égales et Δ_n devient une fonction croissante de $N_{eff}(n)$: le rééchantillonnage est alors d'autant plus nuisible que $N_{eff}(n)$ est proche de N . Cependant, observons que pour une valeur donnée de $N_{eff}(n)$, Δ_n croît lorsque $\frac{p}{N'}$ augmente : pour des valeurs importantes de $\frac{p}{N'}$, le rééchantillonnage doit être évité, même pour de petites valeurs de $N_{eff}(n)$.

Si ces conditions ne sont pas satisfaites, Δ_n n'est plus nécessairement une fonction croissante de $N_{eff}(n)$, mais est borné par deux fonctions croissantes de $N_{eff}(n)$; ainsi, Δ_n a tendance à augmenter avec $N_{eff}(n)$. Nous pouvons être plus précis dans deux cas de figure :

- Si $\frac{p + \frac{m}{2}}{N'}$ est grand, alors Δ_n devient grand, même pour des petites valeurs de $N_{eff}(n)$. Un tel cas de figure se produit lorsque pour des valeurs données de N' et m , p est grand (auquel cas nous sommes face à la situation dont nous avons discuté précédemment) ; ou lorsque pour des valeurs données de p et N' , m est grand, c'est à dire lorsque les composantes du mélange $\pi_{n|n}$ sont bien séparées ;
- Si $\frac{P + \frac{M}{2}}{N'}$ est petit (c'est à dire si le mélange est quasiment réduit à une unique composante étroite) alors Δ_n est petit quel que soit $N_{eff}(n)$: le rééchantillonnage est sans conséquence même lorsque $N_{eff}(n)$ est proche de N .

Remarque 2.12. D'après (2.4.20) et (2.4.21), Δ_n tend vers 0 lorsque N' tend vers l'infini, et pour des paramètres donnés, il est possible de choisir N' tel que la borne supérieure dans (2.4.39) soit inférieure à un seuil donné. Les inconvénients du rééchantillonnage mentionnés à la Remarque 2.11 disparaissent bien évidemment lorsque le nombre de particules rééchantillonnées tend vers l'infini. Cependant, notons d'après (2.4.20)-(2.4.21), que cette convergence est lente et logarithmique (rappelons qu'une suite $\{u_n\}_{n \geq 0}$ converge logarithmiquement si $\lim_{N' \rightarrow +\infty} |u_{n+2} - u_{n+1}| / |u_{n+1} - u_n| = 1$) puisque

$$\lim_{N' \rightarrow +\infty} \frac{|\text{var}(\widehat{\Theta}_{n-1}^{\text{SIR}, N'+1} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) - \text{var}(\widehat{\Theta}_{n-1}^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)|}{|\text{var}(\widehat{\Theta}_{n-1}^{\text{SIR}, N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) - \text{var}(\widehat{\Theta}_{n-1}^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)|} = 1,$$

$$\lim_{N' \rightarrow +\infty} \frac{|\text{var}(\widehat{\Theta}_{n-1}^{\text{SIR}, N'+2} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) - \text{var}(\widehat{\Theta}_{n-1}^{\text{SIR}, N'+1} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)|}{|\text{var}(\widehat{\Theta}_{n-1}^{\text{SIR}, N'+1} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) - \text{var}(\widehat{\Theta}_{n-1}^{\text{SIR}, N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)|} = 1.$$

2.4.5 Comparaison de $\widehat{\Theta}_n^{\text{SIS}}$ et $\widehat{\Theta}_n^{\text{FA}, N'}$

Comme nous l'avons vu, il est aisé de comparer $\text{var}(\widehat{\Theta}_n^{\text{SIR}, N'})$ à $\text{var}(\widehat{\Theta}_n^{\text{FA}, N'})$ et $\text{var}(\widehat{\Theta}_n^{\text{SIR}, N'})$ à $\text{var}(\widehat{\Theta}_n^{\text{SIS}})$, conditionnellement à $\{x_{n-1}^i, w_{n-1}^i\}_{i=1}^N$ et y_n . Il nous reste à comparer $\widehat{\Theta}_n^{\text{FA}, N'}$ à $\widehat{\Theta}_n^{\text{SIS}}$.

Les résultats qui comparent ces deux estimateurs [108, Section 4.2.2] [63] laissent à penser qu'aucun d'entre eux n'est systématiquement plus performant que l'autre. Cela peut être d'ailleurs déduit de (2.4.19)-(2.4.21); en supposant que $\frac{N'-1}{N'}$ est à peu près égal à 1, nous avons

$$\text{var}(\widehat{\Theta}_n^{\text{FA},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) \approx \Delta_n;$$

par conséquent $\widehat{\Theta}_n^{\text{FA},N'}$ est préférable à $\widehat{\Theta}_n^{\text{SIS}}$ si dans l'algorithme SIR, le coût supplémentaire de variance Δ_n introduit par le rééchantillonnage est lui-même inférieur à la variance de $\widehat{\Theta}_n^{\text{SIS}}$. En d'autres termes, alors que l'étape de rééchantillonnage de l'algorithme SIR dégrade nécessairement les performances de l'estimateur SIS (cf. Paragraphe 2.4.4), l'étape de rééchantillonnage de l'algorithme FA peut être bénéfique.

Examinons précisément quels sont les paramètres clés dans la comparaison des deux estimateurs. Soit

$$\Delta'_n = \text{var}(\widehat{\Theta}_n^{\text{FA},N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) - \text{var}(\widehat{\Theta}_n^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n). \quad (2.4.40)$$

D'après (2.4.19)-(2.4.21) et (2.4.25), $\Delta'_n = \Delta_n - \frac{N'-1}{N'} \text{var}(\widehat{\Theta}_n^{\text{SIS}})$. En utilisant (2.4.29) nous obtenons

$$\Delta'_n = A'_n + B_n, \quad (2.4.41)$$

avec

$$A'_n = \sum_{i=1}^N w_n^i \left(\frac{1}{N'} - w_n^i \right) \text{var}_{p(x_n | x_{n-1}^i, y_n)}(f(X)), \quad (2.4.42)$$

et B_n défini en (2.4.31). Bien que le signe de Δ'_n dépende de la fonction f et des paramètres du modèle, l'équation (2.4.41) permet de discuter des performances de chacun des deux estimateurs. Commençons avec les deux cas extrêmes suivants :

- Si $N' = N$ et $w_n^i = \frac{1}{N}$ pour tout i , alors $\Delta'_n \geq 0$ et il est préférable de choisir l'estimateur $\widehat{\Theta}_n^{\text{SIS}}$. Cela ne doit pas surprendre puisque dans ce cas particulier, l'étape de rééchantillonnage élimine en moyenne un tiers des particules (voir le Paragraphe 2.4.4) tandis que toutes les particules, pondérées de la même manière, sont prises en compte dans l'algorithme SIS. Observons cependant que Δ'_n , qui est réduit ici à B_n , est fonction des distances $|\mathbb{E}_{p(x_n | x_{n-1}^i, y_n)}(f(X)) - \mathbb{E}_{p(x_n | x_{n-1}^j, y_n)}(f(X))|$ pour $1 \leq i < j \leq N$; Δ'_n décroît donc lorsque ces distances diminuent. Cela est dû au fait que lorsque les distances sont faibles, le rééchantillonnage n'a pas d'influence; après l'étape de tirage, toutes les particules $f(X_n^i)$ ont presque la même moyenne, indépendamment des particules \tilde{x}_{n-1}^i qui ont été sélectionnées (voir également la Remarque 2.13 ci-dessous).
- S'il existe j tel que $w_n^j = 1$ et $w_n^i = 0$, pour tout $i \neq j$, alors $\Delta'_n = \frac{1-N'}{N'} \text{var}_{p(x_n | x_{n-1}^j, y_n)}(f(X)) \leq 0$ et il est préférable de choisir l'estimateur $\widehat{\Theta}_n^{\text{FA},N'}$. Dans ce cas extrême, $\widehat{\Theta}_n^{\text{SIS}}$ ne dépend que de la j -ème particule ($\widehat{\Theta}_n^{\text{SIS}} = f(X_n^{\text{SIS},j})$ avec $X_n^{\text{SIS},j} \sim p(x_n | x_{n-1}^j, y_n)$), alors que $\widehat{\Theta}_n^{\text{FA},N'} = \frac{1}{N'} \sum_{i=1}^{N'} f(X_n^{\text{FA},i})$ avec $X_n^{\text{FA},i} \sim p(x_n | x_{n-1}^i, y_n)$.

Intéressons nous maintenant aux situations intermédiaires dans lesquelles la distribution des poids n'est ni uniforme, ni réduite à une seule composante non nulle. Nous avons le résultat suivant.

Proposition 2.5. Soient p , P , m et M définis respectivement en (2.4.35), (2.4.36), (2.4.37) et (2.4.38), et soit Δ'_n défini en (2.4.40). Alors

$$\frac{p + \frac{m}{2}}{N'} - \left(P + \frac{m}{2N'} \right) \frac{1}{N_{\text{eff}}(n)} \leq \Delta'_n \leq \frac{P + \frac{M}{2}}{N'} - \left(p + \frac{M}{2N'} \right) \frac{1}{N_{\text{eff}}(n)}. \quad (2.4.43)$$

Démonstration. Nous procédons comme dans la preuve de la Proposition 2.4. D'abord $mS'_n \leq B_n \leq MS'_n$. Ensuite $A'_n = \frac{1}{N'} \sum_{i=1}^N w_n^i \text{var}_{p(x_n|x_{n-1}^i, y_n)}(f(X)) - \sum_{i=1}^N (w_n^i)^2 \text{var}_{p(x_n|x_{n-1}^i, y_n)}(f(X))$. Le premier terme appartient à $[\frac{p}{N'}, \frac{P}{N'}]$ et le second à $[-\frac{P}{N_{eff}(n)}, -\frac{p}{N_{eff}(n)}]$, d'où (2.4.43). \square

Corollaire 2.1. Soit $N' = N$. Si $1 \leq N_{eff}(n) \leq \alpha_n = \max(\frac{2pN+M}{2P+M}, 1)$, alors $\hat{\Theta}_n^{\text{FA}}$ est préférable à $\hat{\Theta}_n^{\text{SIS}}$. Si $\beta_n = \min(\frac{2PN+m}{2p+m}, N) \leq N_{eff}(n) \leq N$, alors $\hat{\Theta}_n^{\text{SIS}}$ est préférable à $\hat{\Theta}_n^{\text{FA}}$.

Démonstration. $\hat{\Theta}_n^{\text{FA}}$ (resp. $\hat{\Theta}_n^{\text{SIS}}$) est préférable à $\hat{\Theta}_n^{\text{SIS}}$ (resp. $\hat{\Theta}_n^{\text{FA}}$) si $\Delta'_n \leq 0$ (resp. ≥ 0), ce qui est assuré si la borne supérieure (resp. inférieure) de l'inégalité (2.4.43) est inférieure à 0 (resp. supérieure à 0). Enfin, $0 < \frac{2pN+M}{2P+M} \leq N$, $\frac{2PN+m}{2p+m} \geq 1$, et d'après la discussion précédente $\Delta'_n(N_{eff}(n)) \leq 0$ si $N_{eff}(n) = 1$ et $\Delta'_n(N_{eff}(n)) \geq 0$ si $N_{eff}(n) = N$. \square

Remarque 2.13. Le résultat du Corollaire 2.1 est paramétré par p, P, m, M et N . Commentons certaines situations spécifiques :

- Pour des valeurs données de p, P et N , $\alpha_n \rightarrow \frac{p}{P}N$ si $M \rightarrow 0$, et en particulier si $p \approx P$ alors $\alpha_n \approx N$; dans ce cas, l'algorithme FA est toujours préférable même lorsque $N_{eff}(n)$ est grand et proche de N (cela peut être également vu à partir de (2.4.41), dans laquelle $B_n = 0$ puisque $M \approx 0$ et $A'_n < 0$ puisque $p = P$). M très petit et $p \approx P$ signifie en fait que $E_{p(x_n|x_{n-1}^i, y_n)}(f(X))$ et $\text{var}_{p(x_n|x_{n-1}^i, y_n)}(f(X))$ sont presque indépendants de i ; cette situation se produit lorsque le mélange $\pi_{n|n}$ est réduit à une unique composante. Dans ce cas, le rééchantillonnage est sans effet puisque, quelle que soit la particule \tilde{x}_{n-1}^i sélectionnée à l'étape i , la moyenne et la variance de $f(X_n^i)$, $X_n^i \sim p(x_n|\tilde{x}_{n-1}^i, y_n)$, ne dépendent pas de \tilde{x}_{n-1}^i .
- A l'opposé, pour des valeurs données de p, P et N , $\beta_n \rightarrow 1$ si $m \rightarrow \infty$: dans ce cas, l'algorithme SIS est préférable, même lorsque N_{eff} est très petit. Rappelons que dans l'algorithme FA, $X_n^i \sim p(x_n|\tilde{x}_{n-1}^i, y_n)$, où \tilde{x}_{n-1}^i est la i -ème particule rééchantillonnée. P étant petit comparé à m , les points $f(X_n^i)$ et $f(X_n^j)$ sont bien séparés si $\tilde{x}_{n-1}^i \neq \tilde{x}_{n-1}^j$, ou dans le même voisinage le cas contraire. L'étape de rééchantillonnage crée donc des amas de points, composés des points qui ont été sélectionnés plusieurs fois, et des régions vides.
- En pratique, il est possible de déduire dans certains cas les valeurs de $N_{eff}(n)$ pour lesquelles $\hat{\Theta}_n^{\text{SIS}}$ est préférable à $\hat{\Theta}_n^{\text{FA}}$. Par exemple, considérons le cas où $p \approx P$. Alors si $m = 2P$, $\beta_n = \frac{N+1}{2}$; pour ces valeurs (qui signifient que les points $f(X_n^i)$ sont bien séparés), l'étape de rééchantillonnage est dangereuse, même si le nombre de particules effective est faible ($\approx \frac{N}{2}$).

Remarque 2.14. Dans le Corollaire 2.1 et la Remarque 2.13, nous avons vu que pour une valeur fixée $N' = N$, l'écart de performances entre l'estimateur FA et l'estimateur SIS dépend des paramètres du modèle et de $N_{eff}(n)$. A l'opposé, fixons $N_{eff}(n)$, pour un jeu de points $\mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}$ et y_n et choisissons $N' \neq N$. Puisque $\text{var}(\hat{\Theta}_n^{\text{FA}, N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) \rightarrow 0$ lorsque $N' \rightarrow +\infty$, $\text{var}(\hat{\Theta}_n^{\text{FA}, N'} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n) \leq \text{var}(\hat{\Theta}_n^{\text{SIS}} | \mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)$ si N' est assez grand. Cependant, en fonction des paramètres, N' n'a pas nécessairement besoin d'être très grand. D'après (2.4.43), $\hat{\Theta}_n^{\text{FA}, N'}$ est préférable à $\hat{\Theta}_n^{\text{SIS}}$ si $N' \geq \frac{1}{2p}[(2P + M)N_{eff}(n) - M]$. Cette borne inférieure tend vers $\frac{P}{p}N_{eff}(n)$ lorsque M tend vers 0, c'est à dire lorsque les composantes du mélange sont identiques. Dans un tel cas, $\hat{\Theta}_n^{\text{FA}, N'}$ est préférable à $\hat{\Theta}_n^{\text{SIS}}$ (au sens où la variance de $\hat{\Theta}_n^{\text{FA}, N'}$ est inférieure à celle de $\hat{\Theta}_n^{\text{SIS}}$) avec moins de N points si $\frac{P}{p} \leq \frac{N}{N_{eff}(n)}$.

Un algorithme SIS/FA hybride

Dans le contexte du filtrage bayésien, nous avons besoin de propager une approximation discrète $\hat{p}_{n|n}$ de $p_{n|n}$ ou de construire un estimateur MC $\hat{\Theta}_n$ de Θ_n . Pour des raisons de coût de calcul, cette seconde étape utilise directement la mesure empirique qui vient juste d'être construite. Pour des enjeux de clarté, posons $N' = N$. Comme nous l'avons rappelé, nous ne devons pas utiliser $\hat{\Theta}_n^{\text{SIR}}$. Choisir entre $\hat{\Theta}_n^{\text{SIS}}$ et $\hat{\Theta}_n^{\text{FA}}$ dépend des poids $\{w_n^i\}_{i=1}^N$ et donc des paramètres à l'instant $n - 1$ $\{(x_{n-1}^i, w_{n-1}^i)\}_{i=1}^N$, ainsi que du modèle à travers la loi $p(y_n|x_{n-1})$. Par conséquent, l'analyse précédente permet de construire un algorithme hybride qui alterne automatiquement entre l'algorithme SIS et l'algorithme FA, à condition bien sûr de pouvoir estimer les seuils intervenant dans le Corollaire 2.1. Notre algorithme hybride est présenté ci-dessous.

Algorithme 2.11: Algorithme Hybride SIS/FA

Soit $\hat{p}_{n-1|n-1} = \sum_{i=1}^N w_{n-1}^i \delta_{x_{n-1}^i}$ une approximation MC de $p_{n-1|n-1}$.

pour $1 \leq i \leq N$ **faire**

 | **W.** Calculer $w_n^i \propto p(y_n|x_{n-1}^i)w_{n-1}^i$, $\sum_{i=1}^N w_n^i = 1$

fin

Calculer $N_{eff}(n)$ dans (2.4.23), α_n et β_n^a dans le Corollaire 2.1 ;

si $N_{eff}(n) > \frac{\alpha_n + \beta_n}{2}$ **alors**

pour $1 \leq i \leq N'$ **faire**

 | **P.** Tirer x_n^i selon $p(x_n|x_{n-1}^i, y_n)$;

fin

 Alors $\hat{p}_{n|n} = \sum_{i=1}^N w_n^i \delta_{x_n^i}$ approxime $p_{n|n}$ et $\hat{\Theta}_n^{\text{SIS}} = \sum_{i=1}^N w_n^i f(x_n^i)$ approxime Θ_n .

fin

si $N_{eff}(n) \leq \frac{\alpha_n + \beta_n}{2}$ **alors**

pour $1 \leq i \leq N'$ **faire**

 | **R.** Tirer $\tilde{x}_{n-1}^i \sim \sum_{j=1}^N w_{n-1}^j \delta_{x_{n-1}^j}$;

 | **P.** Tirer x_n^i selon $p(x_n|\tilde{x}_{n-1}^i, y_n)$.

fin

 Alors $\hat{p}_{n|n} = \sum_{i=1}^N \frac{1}{N} \delta_{x_n^i}$ approxime $p_{n|n}$ et $\hat{\Theta}_n^{\text{FA}} = \frac{1}{N} \sum_{i=1}^N f(x_n^i)$ approxime Θ_n .

fin

a. Les bornes α_n et β_n du Corollaire 2.1 sont déduites de conditions suffisantes et nous n'avons pas de résultat si $N_{eff}(n) \in]\alpha_n, \beta_n[$; pour ces raisons, nous choisissons donc comme seuil $\frac{\alpha_n + \beta_n}{2}$. De même, en pratique, du fait de la difficulté de calculer les seuils, α_n et β_n peuvent être remplacés par une valeur fixe qui prend cependant en compte les paramètres du modèle ; ce sera notre choix dans les simulations, voir le Paragraphe 2.4.6.

Remarque 2.15. Notons que cet algorithme hybride résout automatiquement le problème de dégénérescence des poids : lorsque $N_{eff}(n)$ tombe en dessous d'un certain seuil, alors l'algorithme exécute une itération de l'algorithme FA, qui commence effectivement par une étape de rééchantillonnage. Supposons que plusieurs itérations SIS aient été exécutées jusqu'à l'instant $k - 1$, et que nous disposions du jeu de points $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N$. Si l'algorithme exécute une itération de l'algorithme FA à l'instant k , on pourrait penser que la première étape de rééchantillonnage correspond à la dernière étape de la $k - 1$ -ème itération, et donc que l'Algorithme 2.11 n'est autre qu'un algorithme SIR, avec densité d'importance optimale, et rééchantillonnage optionnel. Notons cependant que dans l'algorithme SIR, les points $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N$ sont rééchantillonnés selon $\sum_{j=1}^N w_{k-1}^j \delta_{x_{k-1}^j}$, alors que dans l'algorithme hybride, ces points sont rééchantillonnés selon $\sum_{j=1}^N w_k^j \delta_{x_{k-1}^j}$, avec $w_k^i \propto p(y_k|x_{k-1}^i)w_{k-1}^i$. Dans l'algorithme hybride, le mécanisme de

rééchantillonnage dépend donc d'un seuil qui varie avec le temps et qui à l'instant n prend en compte $\{x_{n-1}^i, w_{n-1}^i\}_{i=1}^N$, mais aussi le modèle et la nouvelle observation y_n .

Remarque 2.16. L'algorithme hybride est construit à partir de considérations locales (c'est à dire d'analyses conditionnelles) puisqu'à chaque étape, nous décidons entre deux alternatives, conditionnellement à $\{x_{n-1}^i, w_{n-1}^i\}_{i=1}^N$. Cependant, les mêmes arguments peuvent être utilisés pour construire un algorithme avec des propriétés inconditionnelles. Comparons par exemple un estimateur déduit de la succession d'itérations de l'algorithme FA, avec un algorithme qui propage les particules en utilisant un algorithme FA mais qui calcule, en fonction de notre analyse précédente, $\hat{\Theta}'_n$ qui correspond à $\hat{\Theta}_n^{\text{SIS}}$ ou à $\hat{\Theta}_n^{\text{FA}}$. D'après l'égalité RB (1.3.1), $\text{var}(\hat{\Theta}_n^{\text{FA}} | \mathbf{y}_{0:n}) > \text{var}(\hat{\Theta}'_n | \mathbf{y}_{0:n})$; $\hat{\Theta}'_n$ est donc préférable (conditionnellement à $\mathbf{y}_{0:n}$ seulement) à $\hat{\Theta}_n^{\text{FA}}$. Cependant, les efforts de calculs sont plus importants; un tel algorithme pourrait calculer $\hat{\Theta}_n^{\text{SIS}}$ mais propager les points en utilisant une itération FA, contrairement à la plupart des algorithmes SMC qui calculent l'estimateur à partir de l'ensemble pondéré de points propagé.

2.4.6 Simulations

À présent, nous validons nos précédentes discussions à partir de simulations informatiques. Les seuils en dessous desquels il est préférable d'implémenter une étape de rééchantillonnage dans l'algorithme Hybride ou dans l'algorithme SIR sont discutés en fonction des paramètres du modèle donné. Excepté pour la Figure 2.7, nous prenons $N' = N$: le nombre de particules dans les algorithmes SMC considérés ne varie donc pas avec le temps.

Nous réalisons des simulations dans des modèles où les algorithmes FA, SIS et SIR avec densité d'importance conditionnelle optimale peuvent être implémentés, c'est à dire des modèles dans lesquels il est possible de faire des tirages suivant $p(x_n | x_{n-1}, y_n)$ et dans lesquels la vraisemblance prédite $p(y_n | x_{n-1})$ est calculable (des techniques d'approximations ont été développées [43] [110] [119] pour les cas où ces quantités ne sont pas disponibles). Ces modèles incluent les modèles ARCH, généralement utilisés pour des applications financières, ou plus généralement les modèles semi-linéaires et gaussiens, avec une équation d'observation linéaire mais une équation d'état non nécessairement linéaire [43]. Enfin, nous adaptons également notre analyse pour les modèles à sauts (1.1.8) conditionnellement linéaires et gaussiens (1.1.13)-(1.1.14) pour lesquels l'algorithme SIS/SIR et l'algorithme FA ont été adaptés [44]; la principale différence de ces modèles est que le filtrage particulaire agit sur la partie discrète du modèle, c'est à dire sur les sauts qui conditionnent le modèle d'état. Nos simulations sont soit « locale », soit « globale »: par « locale », nous entendons une simulation où l'algorithme hybride (excepté à la Figure 2.7 où il s'agira d'un algorithme FA) propage les points, et à un instant donné n , l'ensemble de points $\mathbf{x}_{n-1}^{1:N}$ produit est utilisé pour calculer les trois estimateurs $\hat{\Theta}_n^{\text{FA}}$, $\hat{\Theta}_n^{\text{SIS}}$ et $\hat{\Theta}_n^{\text{SIR}}$; par « globale », nous entendons une simulation les algorithmes FA, SIS, SIR et hybride sont exécutés indépendamment les uns des autres. Toutes les simulations sont moyennées sur $L = 400$ réalisations.

Modèles ARCH

Considérons le modèle ARCH

$$\begin{cases} X_n &= \sqrt{\beta_0 + \beta_1 X_{n-1}^2} \times U_n \\ Y_n &= X_n + V_n \end{cases}, \quad (2.4.44)$$

où $\{U_n\}$ et $\{V_n\}$ sont i.i.d, mutuellement indépendants et indépendants de X_0 , avec $X_0 \sim \mathcal{N}(\cdot; 0; 1)$, $U_n \sim \mathcal{N}(\cdot; 0; 1)$, $V_n \sim \mathcal{N}(\cdot; 0; R)$. Dans ce modèle, $p(x_n | x_{n-1}, y_n) = \mathcal{N}(x_n; \mu(x_{n-1}); \sigma^2(x_{n-1}))$ avec

$\mu(x_{n-1}) = \frac{\beta_0 + \beta_1 x_{n-1}^2}{\beta_0 + \beta_1 x_{n-1}^2 + R} y_n$ et $\sigma^2(x_{n-1}) = \frac{(\beta_0 + \beta_1 x_{n-1}^2)}{\beta_0 + \beta_1 x_{n-1}^2 + R} R$, et $p(y_n | x_{n-1}) = \mathcal{N}(y_n; 0; R + \beta_0 + \beta_1 x_{n-1}^2)$. Soit $N = 1000$ et $f(x) = x$. Nous étudions d'abord comment choisir le seuil T_n de l'algorithme hybride (en dessous duquel nous exécutons une itération de l'algorithme FA), en fonction des paramètres du modèle (2.4.44). Afin de faire face à des scénarios difficiles, nous générons des trajectoires de longueur $T = 60$ avec des paramètres β_0^0, β_1^0 et R^0 inconnus. Les paramètres $(\beta_0^1, \beta_1^1, R^1)$ utilisés dans nos algorithmes sont $(1, 0.1, 9)$ pour $1 \leq n \leq 20$, $(1, 1, 9)$ pour $21 \leq n \leq 40$ et $(9, 5, 1)$ pour $41 \leq n \leq 60$. Ils ne coïncident pas avec les vrais paramètres $(\beta_0^0, \beta_1^0, R^0)$, ce qui aura un impact sur $N_{eff}(n)$.

Notons d'abord que $\mu(x_{n-1})$ varie entre $\frac{\beta_0}{R + \beta_0} y_n$ et y_n et $\sigma^2(x_{n-1})$ entre $\frac{\beta_0}{R + \beta_0} R$ et R . Notons aussi que seule la variance de $p(y_n | x_{n-1})$ dépend de x_{n-1} . Dans le premier jeu de paramètres, β_1^1 est petit, par conséquent si les particules $\mathbf{x}_{n-1}^{1:N}$ sont proches, les moyennes et les variances de $p(x_n | x_{n-1}^i, y_n)$ seront elles aussi proches et le mélange $\pi_{n|n}$ aura tendance à être réduit à une composante unique. L'estimateur FA est alors préférable dès que $N_{eff}(n)$ est en dessous d'un seuil élevé ; nous choisissons $T_n = 0.95N$. Toutefois, dans ce modèle, la dépendance de $p(y_n | x_{n-1})$ en x_{n-1} est faible parce que β_1^1 est petit. $N_{eff}(n)$ sera donc probablement proche de N et les estimateurs FA et SIS devraient présenter des performances similaires. Dans le second jeu de paramètres, $\beta_1^1 = 1$: les composantes du mélange $\pi_{n|n}$ ont tendance à devenir plus éloignées et leur variance sera également différente. Notons cependant que l'écart entre les composantes du mélange est borné puisque $\mu(x_{n-1})$ varie entre $0.1y_n$ et y_n . Dans cette situation intermédiaire, le seuil doit être inférieur au seuil précédent ; nous choisissons $T_n = 0.8N$. Enfin, considérons le cas informatif $\beta_0^1 = 9$, $\beta_1^1 = 5$ et $R^1 = 1$. Les composantes du mélange $\pi_{n|n}$ se rapprochent puisque les moyennes varient ici entre $0.9y_n$ et y_n , et les variances entre $0.9R^1$ et R^1 ; l'algorithme FA est alors préférable dès lors que $N_{eff}(n)$ est en dessous d'un seuil plus élevé que les deux précédents ; nous choisissons $T_n = 0.97N$.

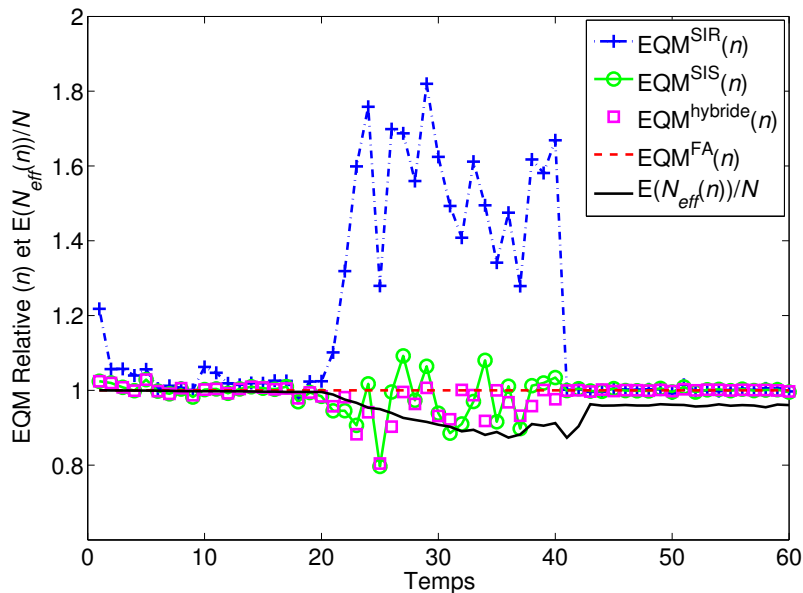


FIGURE 2.4 – EQM relatives et moyenne du nombre de particules effectives - Comparaison locale - Modèle ARCH. Pour $1 \leq n \leq 20$ et $41 \leq n \leq 60$, le mélange $\pi_{n|n}$ a tendance à être réduit à une composante unique, plus ou moins étroite ; pour $21 \leq n \leq 40$, les composantes du mélange $\pi_{n|n}$ sont plus dispersées.

2.4 Comparaison théorique d'algorithmes de filtrage localement optimaux

Nous réalisons une simulation locale et sur la Figure 2.4, nous traçons l'EQM empirique $EQM(n) = \frac{1}{L} \sum_{l=1}^L (\hat{\Theta}_n(l) - \Theta_n)^2$, où $\hat{\Theta}_n$ représente $\hat{\Theta}_n^{FA}$, ou $\hat{\Theta}_n^{SIS}$ ou $\hat{\Theta}_n^{SIR}$, et Θ_n défini en (2.0.1) est la vraie moyenne a posteriori calculée à partir d'un filtre Bootstrap utilisant 5×10^5 particules ainsi que les vrais paramètres $(\beta_0^0, \beta_1^0, R^0)$. Durant les première ($1 \leq n \leq 20$) et dernière ($41 \leq n \leq 60$) parties de la simulation, les algorithmes SIS et FA présentent des performances similaires car $E(N_{eff}(n))$ est proche de N et les composantes du mélange $\pi_{n|n}$ sont très proches. Durant la seconde partie, notons que $\hat{\Theta}_n^{SIS}$ est souvent préférable à $\hat{\Theta}_n^{FA}$, même si $E(N_{eff}(n))$ diminue, et que l'algorithme hybride réussit à s'adapter à la meilleure solution. Notons également que le coût de variance induit par le rééchantillonnage dans l'algorithme SIR dépend du temps. Dans la première partie, les composantes du mélange sont proches, le rééchantillonnage est donc sans effet; tandis que dans la seconde partie, le terme de variance supplémentaire induit est plus important. Dans la dernière partie, le rééchantillonnage n'a aucune incidence.

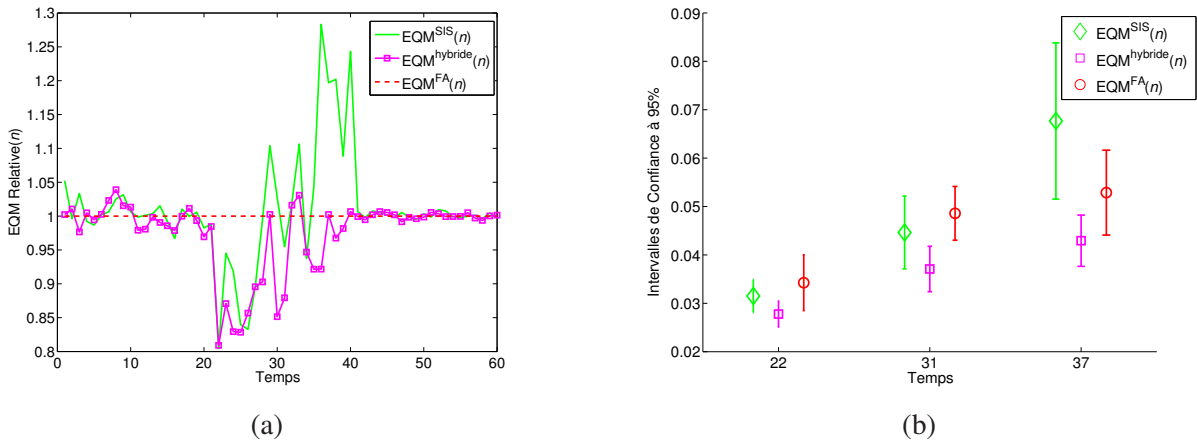


FIGURE 2.5 – Comparaison globale - Modèle ARCH - (a) EQM relatives : durant la seconde partie, le seuil de l'algorithme SIS est trop petit, ce qui entraîne une dégradation de $\hat{\Theta}_n^{SIS}$ au cours du temps. Le seuil de l'algorithme hybride est quant à lui choisi correctement. (b) Intervalles de confiance à 95 % : comportement des $L = 400$ EQM pour $n = 22$, $n = 31$ et $n = 37$; $\hat{\Theta}_n^{SIS}$ et $\hat{\Theta}_n^{FA}$ sont moins fiables que $\hat{\Theta}_n^{hybride}$.

La Figure 2.5 (a), présente les résultats d'une comparaison globale. Nous exécutons indépendamment un algorithme hybride, un algorithme FA et un algorithme SIS où nous rééchantillonnons lorsque $N_{eff}(n) \leq N/3$, une valeur souvent choisie [43]. Durant les première et dernière parties, les composantes du mélange sont proches et $E(N_{eff}(n))$ est proche de N . En effet, au cours la première partie, la dépendance en x_{n-1} de $p(y_n|x_{n-1})$ est faible alors que lors de la troisième, les points $\{x_{n-1}^i\}_{i=1}^N$ ont tendance à être proches puisqu'ils sont tirés suivant la loi $p(x_{n-1}|x_{n-2}^i, y_{n-1})$, quasiment indépendante de i et de variance faible. Lorsque le mélange est réduit à une seule composante et que $N_{eff}(n)$ est important, tous les algorithmes sont équivalents. Durant la seconde partie, les composantes du mélange sont toujours proches mais le modèle est plus sensible à x_{n-1} . Le seuil en dessous duquel nous devons rééchantillonner est important, ce qui n'est pas le cas de l'algorithme SIS. C'est pour cette raison que l'algorithme SIS ne présente pas de bonnes performances comparé à l'algorithme hybride à la fin de la seconde partie.

Sur la Figure 2.5 (b) nous traçons les intervalles de confiance à 95% construits à partir des $L = 400$ EQM de chaque algorithme. Par clarté, ces intervalles sont construits pour trois valeurs différentes du temps, $n = 22$, $n = 31$ et $n = 37$ (dans les première et troisième parties, les algorithmes présentent des performances similaires) et nous avons vérifié que les variances empiriques des 400 EQM présentaient la même forme

caractéristique que celles de la Figure 2.5 (a). Les estimateurs SIS et FA sont moins fiables que celui associé à l’algorithme hybride. Notons que l’estimateur SIS se dégrade en fonction du temps, puisque son seuil de rééchantillonnage n’est pas adapté aux paramètres du modèle. Enfin, nous moyennons également les EQM sur le temps : soit $\mathcal{J} = \frac{1}{T} \sum_{i=1}^T \sqrt{EQM(i)}$; alors nous obtenons $\mathcal{J}^{\text{hybride}} = 0.1173$, $\mathcal{J}^{\text{SIS}} = 0.1429$ et $\mathcal{J}^{\text{FA}} = 0.1305$. Ces résultats mettent en évidence le fait que la minimisation de la variance conditionnelle à chaque instant est une bonne stratégie.

Modèles gaussiens

Considérons maintenant un modèle linéaire et gaussien :

$$\begin{cases} X_n &= 0.9X_{n-1} + U_n \\ Y_n &= X_n + V_n \end{cases}, \quad (2.4.45)$$

où U_n et V_n sont i.i.d., mutuellement indépendants et indépendants de X_0 , avec $X_0 \sim \mathcal{N}(\cdot; 0; 1)$, $U_n \sim \mathcal{N}(\cdot; 0; Q)$ et $V_n \sim \mathcal{N}(\cdot; 0; R)$. Pour ce modèle nous pouvons utiliser le FK mais l’objet de ce paragraphe est d’illustrer les différents points des Paragraphes 2.4.3-2.4.5. Dans ce modèle, $p(x_n|x_{n-1}, y_n) = \mathcal{N}(x_n; \mu(x_{n-1}); \frac{QR}{Q+R})$ avec $\mu(x_{n-1}) = 0.9x_{n-1}(1 - \frac{Q}{Q+R}) + \frac{Q}{Q+R} \times y_n$ et $p(y_n|x_{n-1}) = \mathcal{N}(y_n; 0.9x_{n-1}; Q+R)$. Sur la Figure 2.6, nous réalisons d’abord une simulation globale qui compare l’algorithme FA avec deux algorithmes SIS, dont le rééchantillonnage est effectué lorsque $N_{eff}(n) < 0.8N$ pour le premier, et $0.3N$ pour le second. Nous choisissons $Q = 0.01$, $R = 5$, et $N = 1000$. Pour une telle valeur de Q , les lois $\{p(x_n|x_{n-1}^i, y_n)\}_{i=1}^N$ ont des variances très faibles (≈ 0.01) et sont espacées puisque $1 - \frac{Q}{Q+R} \approx 1$; rappelons alors, d’après la Remarque 2.13, second item, que dans cette situation le rééchantillonnage est localement nuisible ; on s’attend donc à ce que $\hat{\Theta}_n^{\text{SIS}}$ soit préférable à $\hat{\Theta}_n^{\text{FA}}$, même pour de faibles valeurs de $N_{eff}(n)$. Ceci est confirmé par la Figure 2.6, sur laquelle les EQM de l’estimateur FA sont normalisées à 1. Contrairement à la Figure 2.5, l’écart entre les algorithmes SIS et FA est très important. Notons également que l’algorithme SIS utilisant un seuil de $0.3N$ est préférable à celui utilisant un seuil de $0.8N$; dans ce dernier cas, le rééchantillonnage intervient trop rapidement et élimine les composantes essentielles du mélange (rappelons que d’après la Figure 2.3, le nombre de composantes éliminées est plutôt faible lorsque $N_{eff}(n) \approx N/3$).

La Figure 2.7 présente le résultat d’une simulation locale, dans laquelle nous gardons les mêmes paramètres que précédemment, et nous testons l’impact de N' . Nous calculons donc $P(n) \stackrel{\text{def}}{=} \frac{1}{L} \sum_{l=1}^L (\hat{\Theta}_n(l) - E_{\pi_{n|n}}(f(X)))^2$ (un estimateur MC de $\text{var}(\hat{\Theta}_n|\mathbf{x}_{n-1}^{1:N}, \mathbf{w}_{n-1}^{1:N}, y_n)$, moyenné aussi sur $L = 400$ réalisations) pour les estimateurs FA, SIS et SIR et à l’instant $n = 11$, nous comparons, à coût fixe, (voir Remarque 2.9) l’estimateur SIR avec 2000 particules avec l’estimateur FA avec 1500 particules. Les résultats sont normalisés par rapport à $P^{\text{FA}}(n)$, qui vaut ici 1. Comme attendu, si $N' = N$, $\hat{\Theta}_n^{\text{FA}}$ et $\hat{\Theta}_n^{\text{SIS}}$ sont toujours préférables à $\hat{\Theta}_n^{\text{SIR}}$; mais, puisque les composantes du mélange $\pi_{n|n}$ ont des variances faibles et sont séparées, $P^{\text{SIR}}(n)$ est beaucoup plus important que $P^{\text{SIS}}(n)$, ce qui confirme les conclusions de la Remarque 2.11, premier item, alors que $\hat{\Theta}_n^{\text{FA}}$ n’améliore que très légèrement $\hat{\Theta}_n^{\text{SIR}}$, ce qui est en accord avec la fin de la Remarque 2.8. Pour les mêmes raisons, $P^{\text{FA}}(n)$ est aussi plus important que $P^{\text{SIS}}(n)$, ce qui est en accord avec la Remarque 2.13, second item. A l’instant $n = 11$, l’estimateur SIR est préférable à l’estimateur FA. Notons que numériquement, $\frac{P^{\text{SIS}}(11)}{P^{\text{FA}}(11)} < 0.15$ et que la condition sur le terme A_n est donc vérifiée, voir Remarque 2.9.

Discutons à présent du choix des seuils de l’algorithme hybride en fonction des résultats obtenus dans les paragraphes précédents. Dans (2.4.45), Q est un paramètre clé. En effet, pour un jeu de points $\{x_{n-1}^i\}_{i=1}^N$

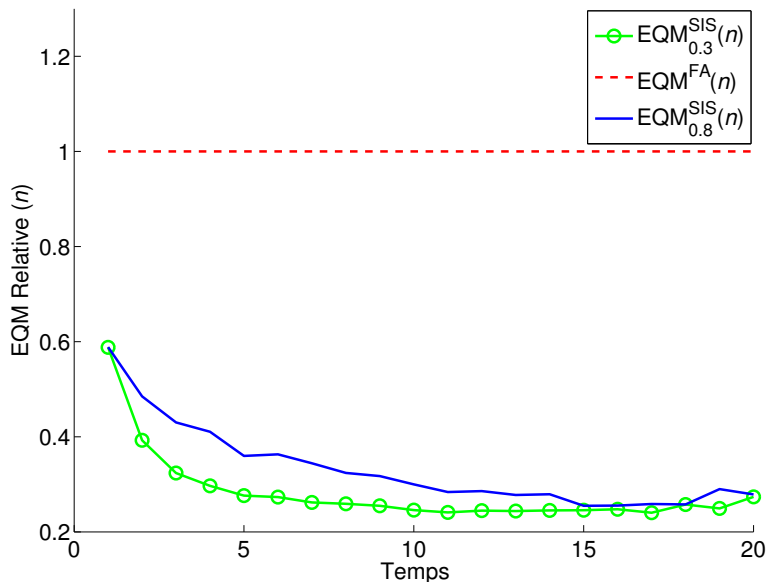


FIGURE 2.6 – EQM relatives - Comparaison globale - Modèle gaussien, $R = 5$, $Q = 0.01$. Les composantes du mélange $\pi_{n|n}$ sont étroites et séparées : le rééchantillonnage est alors nuisible.

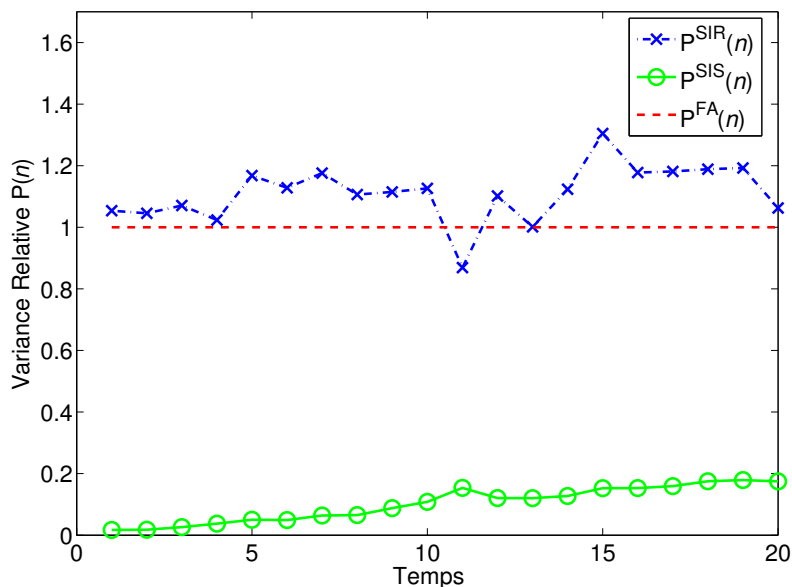


FIGURE 2.7 – Variances relatives $P(n)$ - Modèle gaussien, $R = 5$, $Q = 0.01$. Les composantes du mélange $\pi_{n|n}$ sont étroites et séparées ; le rééchantillonnage est donc nuisible. $N = N' = 1000$, sauf à $n = 11$ où $N'_{SIR} = 2000$ et $N'_{FA} = 1500$; Ici $\hat{\Theta}_{11}^{SIR,2000}$ est préférable $\hat{\Theta}_{11}^{FA,1500}$.

donné, et deux indices j et k , la distance $|m(x_{n-1}^j) - m(x_{n-1}^k)|$ décroît avec Q , alors que la variance commune des lois $\{p(x_n|x_{n-1}^i, y_n)\}_{i=1}^N$ augmente avec Q . Lorsque Q est petit, les lois $\{p(x_n|x_{n-1}^i, y_n)\}_{i=1}^N$ ont donc des variances faibles mais sont bien séparées ; et effectivement, nous avons pu constater sur la Figure 2.6 que $\hat{\Theta}_n^{\text{SIS}}$ est préférable à $\hat{\Theta}_n^{\text{FA}}$ tant que $N_{\text{eff}}(n) > T_n$, $T_n = 0.3N$. Toutefois, lorsque Q augmente, le mélange gaussien a des composantes plus larges et plus proches ; le terme B_n dans (2.4.31) tend alors vers 0 et Δ'_n dans (2.4.40) a tendance à être négatif, quel que soit $N_{\text{eff}}(n)$ (voir Remarque 2.13, premier item). Par conséquent, le rééchantillonnage de l'algorithme FA ne présente pas de risque, même si $N_{\text{eff}}(n)$ est grand. Le seuil T_n de l'algorithme hybride doit être grand dans ce cas. Nous posons $R = 5$ et nous considérons un scénario dans lequel $Q = 0.01$ et $T_n = 0.3N$; pour $1 \leq n \leq 20$; $Q = 2$ et $T_n = 0.65N$ pour $21 \leq n \leq 40$; $Q = 10$ et $T_n = 0.97N$ pour $41 \leq n \leq 60$. Sur la Figure 2.8, nous avons tracé les résultats d'une simulation locale. La première partie ($1 \leq n \leq 20$) est similaire à celle de la Figure 2.7 : puisque les composantes du mélange $\pi_{n|n}$ sont étroites et bien séparées, il est dangereux de rééchantillonner (que ce soit avant ou après l'étape de tirage) et en effet $\hat{\Theta}_n^{\text{SIS}}$ est largement préférable à $\hat{\Theta}_n^{\text{SIR}}$. $\hat{\Theta}_n^{\text{FA}}$ améliore légèrement $\hat{\Theta}_n^{\text{SIR}}$ et $\hat{\Theta}_n^{\text{SIS}}$ est préférable à $\hat{\Theta}_n^{\text{FA}}$ même lorsque $N_{\text{eff}}(n)$ est faible (notons cependant que la distance entre $\hat{\Theta}_n^{\text{SIS}}$ et $\hat{\Theta}_n^{\text{FA}}$ décroît lorsque $E(N_{\text{eff}}(n))$ diminue). Lorsque Q augmente, $\hat{\Theta}_n^{\text{SIS}}$ et $\hat{\Theta}_n^{\text{FA}}$ sont quasiment équivalents pour $\frac{E(N_{\text{eff}}(n))}{N} \approx 0.75$ (bien que $\hat{\Theta}_n^{\text{SIS}}$ reste préférable) mais $\hat{\Theta}_n^{\text{FA}}$ améliore désormais $\hat{\Theta}_n^{\text{SIS}}$ si $\frac{E(N_{\text{eff}}(n))}{N}$ est petit (voir par exemple à l'instant $n = 30$). Dans la troisième partie, ($41 \leq n \leq 60$), $\pi_{n|n}$ est quasiment réduit à une unique composante ce qui influence les performances des algorithmes SIS et FA puisque $\hat{\Theta}_n^{\text{FA}}$ est maintenant préférable à $\hat{\Theta}_n^{\text{SIS}}$ (excepté lorsque $E(N_{\text{eff}}(n))$ est très proche de N). Ce fait illustre la Remarque 2.13, premier item ; dans l'algorithme SIR, l'étape de rééchantillonnage reste dangereuse, comme cela était prévu par les Remarques 2.8 et 2.11, premier item. L'estimateur construit à partir de l'algorithme hybride est toujours préférable aux autres estimateurs, ce qui confirme le choix pertinent des seuils T_n .

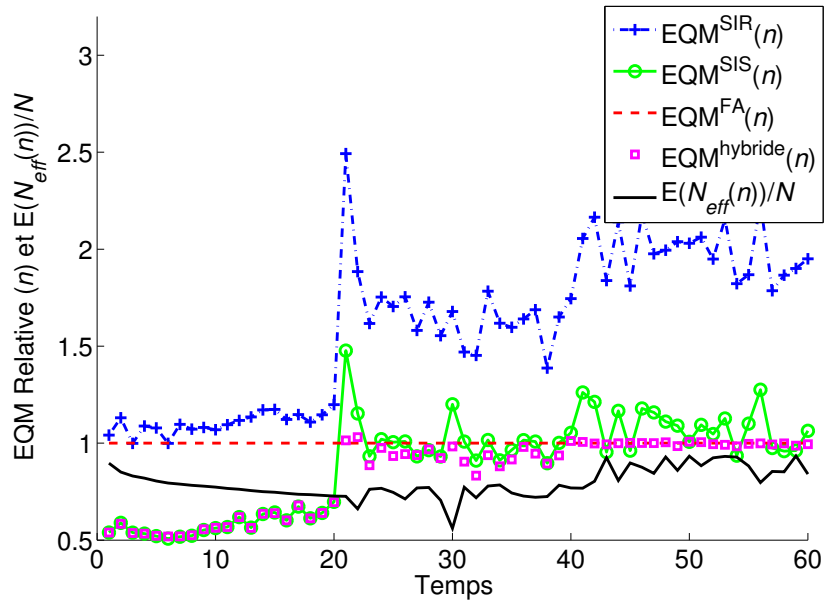


FIGURE 2.8 – EQM relatives - Comparaison locale - Modèle gaussien. Pour $1 \leq n \leq 20$, les composantes du mélange $\pi_{n|n}$ sont séparées et étroites ; pour $21 \leq n \leq 40$, les composantes sont proches ; pour $41 \leq n \leq 60$, le mélange $\pi_{n|n}$ est quasiment réduit à une unique composante large.

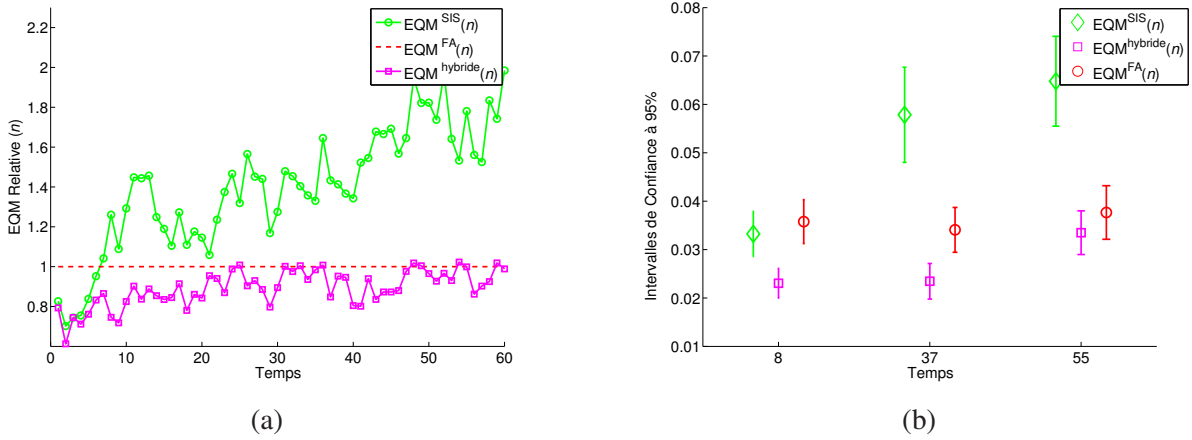


FIGURE 2.9 – Comparaison globale - Modèle gaussien (a) EQM relatives : Au fur et à mesure du temps, le seuil de l'algorithme SIS n'est plus adapté aux paramètres du modèle. (b) Intervalles de confiance à 95 % Comportement des $L = 400$ EQM pour les instants $n = 8$, $n = 37$ et $n = 55$.

Dans une dernière simulation globale, nous générons des trajectoires avec des paramètres inconnus Q^0 et R^0 . Nous réalisons une comparaison globale des algorithmes FA et SIS dans lesquels nous rééchantillons lorsque $N_{eff}(n) \leq N/3$, et hybride, avec un seuil T_n . Les paramètres Q^1 et T_n/N utilisés dans nos algorithmes sont $(1, 0.55)$ pour $1 \leq n \leq 20$, $(2, 0.65)$ pour $21 \leq n \leq 40$ et $(5, 0.85)$ pour $41 \leq n \leq 60$. Les résultats sont donnés sur la Figure 2.9. L'EQM de l'algorithme FA est normalisée à 1, mais est plutôt stable au cours du temps. L'estimateur $\hat{\Theta}_n^{SIS}$ se détériore au cours du temps car lorsque Q^1 augmente, il faut rééchantillonner plus souvent ; le seuil $N_{eff} = N/3$ n'est donc plus adapté aux paramètres du modèle. Observons également que l'algorithme hybride donne systématiquement les meilleurs résultats dans cette comparaison globale, bien que nous l'ayons construit à partir d'une analyse locale. Ceci est aussi confirmé par la Figure 2.9 (b), sur laquelle nous avons tracé les intervalles de confiance à 95% des EQM de chaque algorithme pour des valeurs différentes du temps $n = 6$, $n = 37$ et $n = 55$. Le seuil de l'algorithme SIS n'étant pas adapté, son EQM se dégrade avec le temps tandis que l'estimateur hybride est clairement préférable aux estimateurs FA et SIS. Pendant la troisième partie, les performances des algorithmes hybride et FA sont très proches puisque le seuil de l'algorithme hybride est proche de 1. Enfin, les EQM moyennées sur le temps sont $\mathcal{J}^{hybride} = 0.1781$, $\mathcal{J}^{SIS} = 0.2215$ et $\mathcal{J}^{FA} = 0.1880$, ce qui confirme les bonnes performances de l'algorithme hybride.

Modèles à sauts

Considérons enfin un modèle à sauts conditionnellement linéaire et gaussien,

$$\begin{cases} R_n & \text{est une chaîne de Markov discrète,} \\ \mathbf{X}_n & = \mathbf{F}_n(R_n)\mathbf{X}_{n-1} + \sigma_n(R_n)\mathbf{U}_n, \\ \mathbf{Y}_n & = \mathbf{H}_n(R_n)\mathbf{X}_n + \mathbf{V}_n. \end{cases} \quad (2.4.46)$$

Pour ces modèles, il est possible d'utiliser le FP-RB, dont nous avons rappelé brièvement le principe dans la Section 1.1.2, et qui produit un estimateur $\hat{\Theta}_n$ de Θ_n ,

$$\hat{\Theta}_n = \sum_{i=1}^N w_n^i \mathbb{E}(f(\mathbf{X}_n) | \mathbf{y}_{0:n}, \mathbf{r}_{0:n}^i), \quad (2.4.47)$$

où $E(f(\mathbf{X}_n)|\mathbf{y}_{0:n}, \mathbf{r}_{0:n}^i)$ est calculable à partir du FK. Une approximation MC $\{\mathbf{r}_{0:n}^i, w_n^i\}_{i=1}^N$ de $p(\mathbf{r}_{0:n}|\mathbf{y}_{0:n})$ est propagée par une procédure SIS, SIR ou FA excepté que les poids associés deviennent $w_n^i \propto w_{n-1}^i p(y_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i)$ (le second facteur est calculable en utilisant l'étape de prédiction du FK), et la densité d'importance conditionnelle optimale devient

$$p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n)p(r_n|r_{n-1}^i)}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i)}, \quad (2.4.48)$$

calculable elle aussi par FK. Par conséquent, tous les résultats de cette section peuvent s'appliquer à ces modèles, à une adaptation des notations près : par exemple, $E_{p(x_n|x_{n-1}, y_n)}(f(X))$ doit être remplacé par $E_{p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})}(g^i(R_n))$, où $g^i(R_n) = E(f(\mathbf{X}_n)|\mathbf{r}_{0:n-1}^i, R_n, \mathbf{y}_{0:n})$ qui est calculable à partir du FK. Notons que g^i dépend maintenant de toute la trajectoire $\mathbf{r}_{0:n-1}^i$. Les moyennes $\{E_{p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})}(g^i(R_n))\}_{i=1}^N$ n'ont donc pas de raison d'être proches les unes des autres et le seuil en dessous duquel nous rééchantillons doit être relativement bas, excepté dans certaines situations que nous discuterons.

Nous nous intéressons à un scénario de poursuite dans lequel $R_n \in \{1, 2, 3\}$ représente le comportement d'un objet manoeuvrant (trajectoire rectiligne, virage à gauche, virage à droite). Nous avons $\mathbf{x}_n = [p_{x,n}, \dot{p}_{x,n}, p_{y,n}, \dot{p}_{y,n}]^T$ et nous estimons $p_{x,n}$ (le raisonnement pour les autres composantes de \mathbf{X}_n est identique). Posons

$$\mathbf{F}(R) = \begin{bmatrix} 1 & \frac{\sin(\omega_r T_s)}{\omega_r} & 0 & -\frac{1-\cos(\omega_r T_s)}{\omega_r} \\ 0 & \cos(\omega_r T_s) & 0 & -\sin(\omega_r T_s) \\ 0 & \frac{1-\cos(\omega_r T_s)}{\omega_r} & 1 & \frac{\sin(\omega_r T_s)}{\omega_r} \\ 0 & \sin(\omega_r T_s) & 0 & \cos(\omega_r T_s) \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

avec $T_s = 2s$, $\omega = 0 \text{ rad.s}^{-1}$ lorsque $R = 1$, $\omega_r = 5\pi/180 \text{ rad.s}^{-1}$ lorsque $R = 2$ et $\omega_r = -5\pi/180 \text{ rad.s}^{-1}$ lorsque $R = 3$,

$$\mathbf{U}_n \sim \mathcal{N}\left(., 0; \mathbf{Q} = \begin{bmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} & 0 & 0 \\ \frac{T_s^2}{2} & T_s & 0 & 0 \\ 0 & 0 & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ 0 & 0 & \frac{T_s^2}{2} & T_s \end{bmatrix}\right) \text{ et } \mathbf{V}_n \sim \mathcal{N}\left(., 0; \mathbf{R}_v = \begin{pmatrix} 3^2 & 0 \\ 0 & 3^2 \end{pmatrix}\right).$$

La Figure 2.10 visualise l'effet des probabilités de transition et de $\sigma_n(R)$ sur les EQM des trois estimateurs. Nous discutons par ailleurs du choix du seuil de chaque algorithme.

Nous posons d'abord $p(r_n|r_{n-1}) = 0.8$ si $r_n = r_{n-1}$, $p(r_n|r_{n-1}) = 0.1$ sinon, et $\sigma_n(R_n) = 1$ pour $R_n = 1$, $\sigma_n(R_n) = 3$ si $R_n \neq 1$. Comme nous l'avons fait remarquer plus haut, nous avons un mélange dont les composantes ont tendance à être séparées puisque les fonctions $g^i(R)$ dépendent de trajectoires diversifiées $\{\mathbf{r}_{0:n-1}^i\}_{i=1}^N$. D'un point de vue heuristique, éliminer les bonnes trajectoires $\{\mathbf{r}_{0:n-1}^i\}_{i=1}^N$ pendant l'étape de rééchantillonnage de l'algorithme FA peut être dangereux car l'étape de tirage consiste seulement à tirer $\{r_n^i\}_{i=1}^N$; en d'autres termes, il n'est pas possible de retrouver les trajectoires perdues, ce qui peut avoir de sérieuses conséquences puisque g^i dépend de $\mathbf{r}_{0:n-1}^i$. L'étape de rééchantillonnage ne doit alors être effectuée que lorsque $N_{eff}(n)/N \leq 0.3$. Nous exécutons un algorithme FA, un algorithme SIS dans lequel nous rééchantillons lorsque $N_{eff}(n)/N \leq 0.3$, et notre algorithme hybride. Les EQM relatives (respectivement normalisées par rapport à l'EQM de l'algorithme FA, prises égales à 1) sont tracées sur la Figure 2.10 (a). L'estimateur hybride coïncide avec l'estimateur SIS jusqu'à $n = 7$, et ils sont tous deux préférables à l'estimateur FA. Lorsque $n \geq 7$, l'estimateur hybride est préférable à l'estimateur SIS, ce qui

2.5 Conclusion

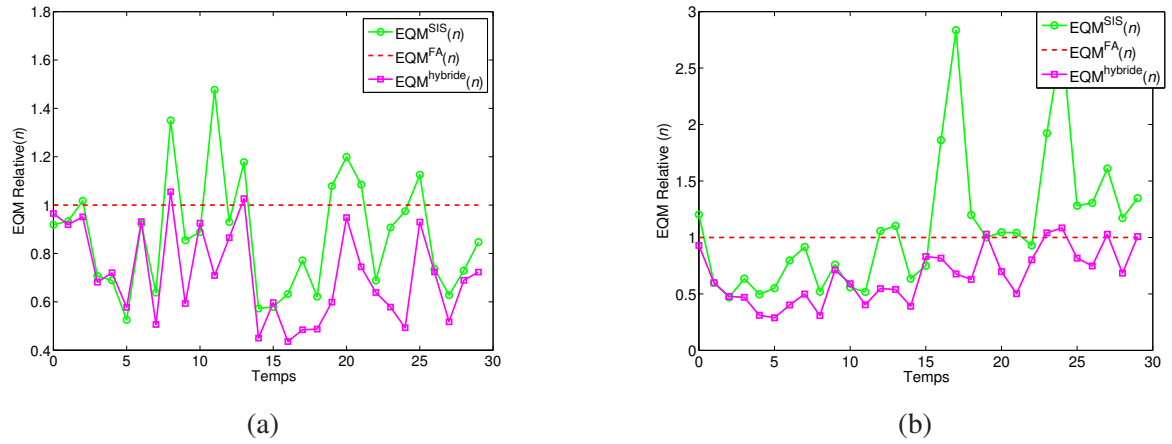


FIGURE 2.10 – EQM relatives - Modèle à sauts - (a) Forte dépendance en r : $p(r_n|r_{n-1}) = 0.8$ si $r_n = r_{n-1}$ et $p(r_n|r_{n-1}) = 0.2$ si $r_n \neq r_{n-1}$; $\sigma_n(1) = 1$, $\sigma_n(2) = \sigma_n(3) = 3$. Le seuil de l'algorithme hybride est $T_n = N/3$. (b) Faible dépendance en r : $p(r_n = 1|r_{n-1}) = 0.9$, $p(r_n|r_{n-1}) = 0.05$ lorsque $r_n \neq 1$; $\sigma_n(1) = \sigma_n(2) = \sigma_n(3) = 1$. Le seuil de l'algorithme hybride est $T_n = 0.75N$.

signifie que le rééchantillonnage employé par l'algorithme hybride (qui correspond en fait à une étape de l'algorithme FA, voir la Remarque 2.15) est plus efficace que celui utilisé dans l'algorithme SIS.

En revanche, sur la Figure 2.10 (b), nous privilégions le mode $R_n = 1$ en posant $p(r_n = 1|r_{n-1}) = 0.9$ et $p(r_n|r_{n-1}) = 0.05$ lorsque $r_n \neq 1$, et nous réduisons également la dépendance du modèle en R_n en posant $\sigma_n(R_n) = 1$ pour tout R_n . La loi optimale (2.4.48) est donc probablement importante lorsque $r_n = 1$ et les fonctions $g^i(R)$ devraient être assez proches les unes des autres. Dans ce cas particulier, le seuil en dessous duquel il faut rééchantillonner doit être relevé aussi prenons nous $T_n = 0.75N$. Nous exécutons les algorithmes FA et SIS (dans lequel nous rééchantillonons lorsque $N_{eff}(n)/N \leq 0.3$) et notre algorithme hybride. Les estimateurs hybride et SIS coïncident jusqu'à $n = 3$, mais le rééchantillonnage intervient visiblement trop tard dans l'algorithme SIS. Il semble également que le seuil de l'algorithme SIS soit trop faible puisqu'après 15 itérations, l'estimateur associé présente des performances beaucoup moins bonnes que les estimateurs construits à partir des algorithmes FA et hybride.

2.5 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux algorithmes de FP permettant de propager une approximation de la loi de filtrage $p(\mathbf{x}_n|y_{0:n})$, et d'en déduire un estimateur d'un moment Θ_n suivant cette loi. Plusieurs algorithmes déduits de différents chemins bayésiens ont alors été proposés. Ces algorithmes ont été comparés avec les algorithmes de FP classiques (algorithmes SIR et APF) par des simulations informatiques, ce qui nous a permis par la même occasion de discuter de leur implémentation pratique. Nous avons observé que les algorithmes utilisant la nouvelle observation \mathbf{y}_n permettent d'anticiper de fortes variations de l'état caché que l'on cherche à estimer et donnent des résultats assez proches.

Au vu de ces résultats, les algorithmes localement optimaux (SIS/SIR avec densité d'importance conditionnelle optimale et FA), qui partagent les mêmes opérations élémentaires, ont fait l'objet d'une analyse statistique non asymptotique locale. Pour un moment Θ_n donné, nous avons discuté localement des performances de chacun de ces algorithmes en fonction des paramètres du modèle probabiliste donné et du nombre

effectif de particules qui caractérise la taille du support de l'approximation discrète. À nombre de particules égal, $\hat{\Theta}_n^{\text{FA}}$ est toujours préférable à $\hat{\Theta}_n^{\text{SIR}}$, ce qui s'interprète, sur un plan pratique, par le fait que l'algorithme FA ne souffre pas du phénomène de dégénérescence du support. De plus, bien que $\hat{\Theta}_n^{\text{SIS}}$ soit toujours préférable à $\hat{\Theta}_n^{\text{SIR}}$, nous avons montré que le rééchantillonnage a tendance à être nuisible dès lors que $N_{\text{eff}}(n)$ est proche de N , même si les effets de cette opération dépendent également du mélange $\pi_{n|n}$ et donc du modèle statistique sous-jacent. Enfin, aucun des estimateurs basés sur l'algorithme FA et SIS n'est systématiquement préférable à l'autre, ce qui est cohérent avec les résultats asymptotiques existants [63]. Nous avons donc proposé un algorithme Hybride qui à chaque instant alterne automatiquement entre une itération de l'algorithme SIS et une itération de l'algorithme FA, en fonction du modèle statistique et du nombre de particules effectif $N_{\text{eff}}(n)$ à cet instant. Des simulations nous ont permis de vérifier nos analyses.

Techniques de réduction de variance Rao-Blackwellisées pour les méthodes SMC

Sommaire

3.1	État de l'art	77
3.2	Filtres particuliers PHD marginalisés pour le filtrage multi-objets bayésien	80
3.3	Méthodes Monte Carlo conditionnelles temporelles	96
3.4	Conclusion	115

3.1 État de l'art

Le problème de réduction de variance dans le contexte des méthodes MC est un problème récurrent qui a fait l'objet de nombreuses recherches (voir par exemple [50] [9]). Une méthode souvent employée repose sur le résultat suivant : soient X_1 et X_2 deux v.a. et $f(\cdot)$ une fonction donnée ; alors (nous supposons implicitement que les moments impliqués existent)

$$E(E(f(X_2)|X_1)) = E(f(X_2)), \tag{3.1.1}$$

$$\text{var}(E(f(X_2)|X_1)) = \text{var}(f(X_2)) - E(\text{var}(f(X_2)|X_1)). \tag{3.1.2}$$

Par conséquent, si nous souhaitons calculer le moment $\Theta = E(f(X_2))$ et que nous avons à notre disposition deux jeux de points aléatoires, $\{X_1^i\}_{i=1}^N \stackrel{\text{i.i.d.}}{\sim} p(x_1)$ et $\{X_2^i\}_{i=1}^N \stackrel{\text{i.i.d.}}{\sim} p(x_2)$, alors les estimateurs

$$\tilde{\Theta} = \frac{1}{N} \sum_{i=1}^N E(f(X_2)|X_1^i), \tag{3.1.3}$$

$$\hat{\Theta} = \frac{1}{N} \sum_{i=1}^N f(X_2^i) \tag{3.1.4}$$

ont la même espérance d'après (3.1.1), mais la variance de l'estimateur $\tilde{\Theta}$ est inférieure à celle de l'estimateur $\hat{\Theta}$ d'après (3.1.2). L'estimateur $\tilde{\Theta}$ est appelé « estimateur CMC » tandis que l'estimateur $\hat{\Theta}$ est appelé « estimateur purement MC (= *crude* MC) ». L'intérêt d'utiliser $\tilde{\Theta}$ plutôt que $\hat{\Theta}$ dépend du choix de X_1 : idéalement, il faut être en mesure d'obtenir aisément des tirages suivant $p(x_1)$; la réduction de variance dans (3.1.2) doit aussi être la plus large possible mais la fonction $g(x_1) = E(f(X_2)|x_1)$ doit être calculable à un coût raisonnable.

Les méthodes CMC ont été adaptées au contexte du filtrage bayésien étudié au Chapitre 2 afin de réduire la variance des estimateurs d'intérêt ; en particulier, ces méthodes connues sous le nom de « Filtres particuliers marginalisés » ou « Filtres particuliers Rao-Blackwellisés (FP-RB) » ont fait l'objet de nombreuses

contributions [23] [43] [44] [76] [112]. L'idée est la suivante. Ainsi que nous l'avons rappelé dans le Chapitre 1, le problème d'estimation consiste à calculer ou approcher le moment Θ_n défini par

$$\Theta_n = \int f(\mathbf{x}_n) p(\mathbf{x}_{0:n} | \mathbf{y}_{0:n}) d\mathbf{x}_{0:n}. \quad (3.1.5)$$

L'application des techniques de marginalisation est très importante dans le contexte du filtrage bayésien car les FP que nous avons présentés tout au long du second chapitre produisent de mauvais estimateurs dès lors que la dimension du vecteur d'état \mathbf{X}_n devient importante. Si la v.a. $\mathbf{X}_n = (\mathbf{X}_n^1, \mathbf{X}_n^2)$ est multi-dimensionnelle, nous pouvons réécrire $\Theta = E(f(X_1, X_2))$ avec $\mathbf{X}_{0:n} = (\mathbf{X}_{0:n}^1, \mathbf{X}_{0:n}^2) = (X_1, X_2)$. Comme nous l'avons vu précédemment, il n'est généralement pas possible d'obtenir des tirages suivant $p(x_1, x_2)$, et $p(x_1, x_2) \propto p'(x_1, x_2)$ n'est connu qu'à une constante près, d'où l'utilisation des techniques d'échantillonnage d'importance normalisé, rappelées au paragraphe 2.1 [51]. Considérons alors les deux estimateurs

$$\hat{\Theta}(\mathbf{x}_1^{1:N}, \mathbf{x}_2^{1:N}) = \sum_{i=1}^N w_2^i(\mathbf{x}_1^{1:N}, \mathbf{x}_2^{1:N}) f(x_1^i, x_2^i) \text{ avec } (x_1^i, x_2^i) \sim q_2(\cdot), \quad (3.1.6)$$

$$\hat{\Theta}^{\text{RB}}(\mathbf{x}_1^{1:N}) = \sum_{i=1}^N w_1^i(\mathbf{x}_1^{1:N}) E(f(x_1^i, X_2) | x_1^i) \text{ avec } x_1^i \sim q_1(\cdot), \quad (3.1.7)$$

où $\sum_{i=1}^N w_1^i = \sum_{i=1}^N w_2^i = 1$. En particulier, si $w_1^i \propto w_1^{u,i} = p'(x_1^i)/q_1(x_1^i)$, $w_2^i \propto w_2^{u,i} = p'(x_1^i, x_2^i)/q_2(x_1^i, x_2^i)$ et $q_1(x_1) = \int q_2(x_1, x_2) dx_2$, alors la variance du poids non normalisé $w_1^{u,i}$ ne peut être qu'inférieure à celle de $w_2^{u,i}$ d'après l'égalité (3.1.2) [43]. De plus, si les points (x_1^i, x_2^i) sont indépendants, une analyse asymptotique basée sur (3.1.1) et (3.1.2) montre que $\hat{\Theta}^{\text{RB}}$ est en effet meilleur que $\hat{\Theta}$ [44]. Cependant, dans le cas général, comme nous l'avons vu au Chapitre 2, l'hypothèse d'indépendance n'est plus vérifiée dès lors que le rééchantillonnage est utilisé. Si l'étape de rééchantillonnage est effectuée à chaque itération alors l'estimateur $\hat{\Theta}^{\text{RB}}$ a une variance asymptotique plus faible que celle de l'estimateur $\hat{\Theta}$ [24].

En définitive $\hat{\Theta}^{\text{RB}}$ peut être utilisé si nous disposons d'une approximation MC de $p(x_1)$ et s'il est possible de calculer $E(f(X_2) | X_1)$, ce qui a été fait dans les modèles suivants :

Modèle partiellement linéaire et gaussien : le vecteur d'état $\mathbf{X}_{0:n}$ peut être décomposé en deux parties, appelées typiquement composante linéaire, $\mathbf{x}_2 = \mathbf{x}_{0:n}^1$, et composante non linéaire, $\mathbf{x}_1 = \mathbf{x}_{0:n}^{\text{nl}}$. Les classes de modèles dans lesquelles le calcul de $\hat{\Theta}^{\text{RB}}$ est possible incluent par exemple le modèle partiellement linéaire et gaussien, éventuellement partiellement observé [23] [112], décrit par les équations

$$\mathbf{X}_n^{\text{nl}} = \mathbf{f}_n^{\text{nl}}(\mathbf{X}_{n-1}^{\text{nl}}) + \mathbf{F}_n^{\text{nl}}(\mathbf{X}_{n-1}^{\text{nl}}) \mathbf{X}_{n-1}^1 + \mathbf{U}_n^{\text{nl}}, \quad (3.1.8)$$

$$\mathbf{X}_n^1 = \mathbf{f}_n^1(\mathbf{X}_{n-1}^{\text{nl}}) + \mathbf{F}_n^1(\mathbf{X}_{n-1}^{\text{nl}}) \mathbf{X}_{n-1}^1 + \mathbf{U}_n^1, \quad (3.1.9)$$

$$\mathbf{Y}_n = \mathbf{h}_n(\mathbf{X}_n^{\text{nl}}) + \mathbf{H}_n(\mathbf{X}_n^{\text{nl}}) \mathbf{X}_n^1 + \mathbf{V}_n, \quad (3.1.10)$$

où $\mathbf{f}_n^{\text{nl}}(\cdot)$ est une fonction de $\mathbb{R}^{p_{\text{nl}}}$ dans $\mathbb{R}^{p_{\text{nl}}}$, $\mathbf{F}_n^{\text{nl}}(\cdot)$ est une matrice de taille $p_{\text{nl}} \times p_l$ qui dépend des composantes du vecteur \mathbf{X}_n^{nl} et qui agit sur le vecteur \mathbf{X}_n^1 (les fonctions $\mathbf{f}_n^{\text{nl}}(\cdot)$, $\mathbf{h}_n(\cdot)$, $\mathbf{F}_n^{\text{nl}}(\cdot)$ et $\mathbf{H}_n(\cdot)$ sont définies de manière similaire), et

$$\begin{bmatrix} \mathbf{U}_n^{\text{nl}} \\ \mathbf{U}_n^1 \\ \mathbf{V}_n \end{bmatrix} \sim \mathcal{N} \left(\cdot; \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}; \begin{bmatrix} \mathbf{Q}_n^{\text{nl}} & (\mathbf{S}_n^{\text{nl}})^T & 0 \\ (\mathbf{S}_n^{\text{nl}}) & \mathbf{Q}_n^1 & 0 \\ 0 & 0 & \mathbf{R}_n \end{bmatrix} \right), \quad (3.1.11)$$

où $[\mathbf{U}_1^{\text{nl}}, \mathbf{U}_1^{\text{l}}], \dots, [\mathbf{U}_n^{\text{nl}}, \mathbf{U}_n^{\text{l}}]$ et $\mathbf{V}_1, \dots, \mathbf{V}_n$ sont indépendants. Conditionnellement à $\mathbf{X}_{n-1}^{\text{nl}}, \mathbf{X}_n^{\text{l}}$ suit un modèle linéaire (voir (3.1.9)) d'où le terme de composante linéaire. De même, conditionnellement à $\mathbf{X}_{n-1}^{\text{l}}, \mathbf{X}_n^{\text{nl}}$ suit un modèle non-linéaire (voir (3.1.8)) d'où le terme de composante non linéaire. Une représentation probabiliste de ce modèle HMC est donnée par les lois

$$f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N} \left(\mathbf{x}_n; \begin{bmatrix} \mathbf{f}_n^{\text{nl}}(\mathbf{x}_{n-1}^{\text{nl}}) \\ \mathbf{f}_n^{\text{l}}(\mathbf{x}_{n-1}^{\text{l}}) \end{bmatrix} + \begin{bmatrix} \mathbf{F}_n^{\text{nl}}(\mathbf{x}_{n-1}^{\text{nl}}) \\ \mathbf{F}_n^{\text{l}}(\mathbf{x}_{n-1}^{\text{l}}) \end{bmatrix} \mathbf{x}_{n-1}^{\text{l}}; \begin{bmatrix} \mathbf{Q}_n^{\text{nl}} & (\mathbf{S}_n^{\text{nl}})^T \\ (\mathbf{S}_n^{\text{nl}}) & \mathbf{Q}_n^{\text{l}} \end{bmatrix} \right), \quad (3.1.12)$$

$$g_n(\mathbf{y}_n | \mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{h}_n(\mathbf{x}_n^{\text{nl}}) + \mathbf{H}_n(\mathbf{x}_n^{\text{nl}}) \mathbf{x}_n^{\text{l}}; \mathbf{R}_n). \quad (3.1.13)$$

Modèle à sauts linéaire et gaussiens : Ces méthodes ont également été appliquées dans le contexte des modèles à sauts,

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n}) = p(r_0) \prod_{i=1}^n p(r_i | r_{i-1}) p(\mathbf{x}_0 | r_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1}, r_i) \prod_{i=0}^n g_i(\mathbf{y}_i | \mathbf{x}_i, r_i), \quad (3.1.14)$$

vérifiant

$$f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n(r_n) \mathbf{x}_{n-1}; \mathbf{Q}_n(r_n)), \quad (3.1.15)$$

$$g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n(r_n) \mathbf{x}_n; \mathbf{R}_n(r_n)), \quad (3.1.16)$$

dans lesquels nous réécrivons

$$\Theta_n = \sum_{\mathbf{r}_{0:n}} \int f(\mathbf{x}_{0:n}) p(\mathbf{x}_{0:n}, \mathbf{r}_{0:n} | \mathbf{y}_{0:n}) d\mathbf{x}_{0:n}, \quad (3.1.17)$$

et dans lesquels nous identifions $\mathbf{x}_2 = \mathbf{r}_{0:n}$ et $\mathbf{x}_1 = \mathbf{x}_{0:n}$. En effet, dans ces modèles, l'intégrale $\int f(\mathbf{x}_{0:n}) \times p(\mathbf{x}_{0:n} | \mathbf{y}_{0:n}, \mathbf{r}_{0:n}) d\mathbf{x}_{0:n}$ peut être déduite du FK tandis que $p(\mathbf{r}_{0:n} | \mathbf{y}_{0:n})$ peut être approchée par un FP dont l'expression des poids est déduite du FK [44] [23].

Lorsqu'il n'est plus possible de calculer un moment suivant $p(\mathbf{x}_2 | \mathbf{x}_1)$, il est possible d'envisager des méthodes d'approximations numériques, de type EKF/UKF afin de propager une approximation de $p(\mathbf{x}_2 | \mathbf{x}_1)$. Néanmoins, cette approximation numérique de $p(\mathbf{x}_2 | \mathbf{x}_1)$ est également utilisée dans le calcul des poids associés à l'expression de $p(\mathbf{x}_1)$ et est par conséquent propagée dans le temps. Plus récemment, des méthodes d'approximations du calcul de $\hat{\Theta}^{\text{RB}}$ basées sur l'utilisation de FP locaux ont été proposées [117] [65]. De façon très générale, ces méthodes utilisent un second FP local pour approcher $p(\mathbf{x}_2 | \mathbf{x}_1)$; l'approximation locale obtenue permet également de calculer les poids associés à l'approximation de $p(\mathbf{x}_1)$. Nous détaillerons cette méthode locale dans le Paragraphe 3.3.1.

Après cette brève introduction des techniques de type RB en FP, nous en venons désormais aux contributions de cette thèse. L'objet de ce chapitre est double.

- Dans un premier temps, nous nous intéressons aux méthodes SMC dans le cadre du filtrage multi-objets et plus précisément pour la propagation du PHD, dont nous avons rappelé le principe au Paragraphe 1.2.2. Bien sûr, les solutions basées sur l'utilisation des méthodes SMC héritent des mêmes problèmes que les filtres bayésiens (mono-objet) dont nous avons discuté jusqu'ici. Mais les conséquences sont plus sévères dans ce contexte, car il faut à la fois estimer les états cachés de chaque objet et le nombre d'objets qui est inconnu et supposé aléatoire. Comme dans le cas classique, ce problème peut être atténué dans des situations spécifiques. Une première tentative de l'utilisation des FP marginalisés a été proposée dans un contexte d'association de données [116] : une méthode particulière est

utilisée pour effectuer l'étape d'association de données, et conditionnellement à cette étape, l'estimation des états cachés est faite de façon exacte. Ces techniques ont été également adaptées pour le filtre bayésien multi-objets (1.2.7)-(1.2.8) [120], mais restent néanmoins rarement implémentables à cause de leur complexité lorsque le nombre d'objets est important. La première section de ce chapitre est donc consacrée à l'adaptation des techniques de marginalisation au filtre PHD (1.2.10)-(1.2.11).

- Dans la seconde section de ce chapitre, nous continuons d'appliquer les résultats (3.1.1)-(3.1.2) au problème de filtrage mono- ou multi-objet(s), mais en utilisant une partition différente du vecteur d'état que celle habituellement utilisée et rappelée ci-dessus. Comme nous l'avons montré tout au long du Chapitre 2, une approximation de Θ_n est construite à partir d'une approximation MC $\hat{p}_{0:n|n}$ de $p_{0:n|n}$, elle-même déduite d'une approximation $\hat{p}_{0:n-1|n-1}$ de $p_{0:n-1|n-1}$. La partition temporelle $\mathbf{X}_{0:n} = (\mathbf{X}_1, \mathbf{X}_2)$, avec $\mathbf{X}_1 = \mathbf{X}_{0:n-1}$ et $\mathbf{X}_2 = \mathbf{X}_n$, apparaît donc naturellement dans le problème d'estimation de Θ_n puisque nous disposons à l'instant $n - 1$ d'une approximation MC impliquant le vecteur aléatoire $\mathbf{X}_{0:n-1}$. Nous étudierons donc l'applicabilité des résultats (3.1.1)-(3.1.2) à une décomposition temporelle du vecteur d'état $\mathbf{X}_{0:n}$.

3.2 Filtres particuliers PHD marginalisés pour le filtrage multi-objets bayésien

Dans cette section, nous nous focalisons sur le problème de filtrage multi-objets. Plus précisément, dans le Paragraphe 3.2.2 nous appliquons le principe de marginalisation au PHD pour les modèles HMC vérifiant (3.1.12)-(3.1.13), ce qui donne le filtre PHD marginalisé (*Marginalized PHD* = M-PHD).

Ce filtre M-PHD ne peut pas être calculé analytiquement, aussi l'objet du Paragraphe 3.2.3 est d'en proposer deux implémentations pratiques. La première est une implémentation « bootstrap », c'est à dire qu'elle s'inspire de l'Algorithme 1-*P* (voir Algorithme 2.3) qui tire de nouvelles particules à partir de la loi $f_{n|n-1}$. Cependant, rappelons que les méthodes SMC peuvent donner de mauvais résultats, même lorsque la dimension du vecteur d'état est petite, si la loi d'importance pour tirer de nouvelles particules est mal choisie. Notre seconde implémentation est donc une implémentation auxiliaire, qui s'inspire des Algorithmes 1-*S* (voir Algorithme 2.4) et APF (voir Algorithme 2.2) en sélectionnant parmi les anciennes particules celles qui ont une vraisemblance importante avec les nouvelles observations disponibles. Cette dernière implémentation requiert cependant des hypothèses supplémentaires spécifiques afin d'obtenir des résultats satisfaisants.

Ensuite, dans le Paragraphe 3.2.4, nous considérons la classe des modèles à sauts linéaires et gaussiens, caractérisée par les équations (3.1.15)-(3.1.16). Pour ces modèles, une implémentation par mélange de gaussiennes a été proposée [92] [83]. De par le nombre d'observations et le nombre de modes pris par le processus discret associé à chaque objet, le PHD est un mélange qui croît exponentiellement au cours du temps ; des approximations numériques de type élagage et fusion de gaussiennes sont alors utilisées. Nous adaptons donc notre algorithme M-PHD pour de tels modèles, ce qui permet de limiter la croissance du mélange due aux nombres de modes, tout en gardant la structure de mélange gaussien pour les états cachés. Nous proposons également une version du M-PHD pour des modèles à sauts présentant des structures partiellement linéaires et non linéaires.

Enfin, le Paragraphe 3.2.5 est consacré à la comparaison par simulations des algorithmes proposés.

3.2.1 Quelques rappels sur le filtre PHD

Dans ce paragraphe, nous nous focalisons sur le principe de l'implémentation SMC du filtre PHD [122] [26] et nous analysons ses limites.

Implémentation du filtre PHD

Les équations du filtre PHD que nous rappelons ici

$$v_{n|n-1}(\mathbf{x}) = \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x}|\mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1} + \gamma_n(\mathbf{x}), \quad (3.2.1)$$

$$v_n(\mathbf{x}) = (1 - p_{d,n}(\mathbf{x})) v_{n|n-1}(\mathbf{x}) + \sum_{\mathbf{z} \in \mathcal{Z}_n} \frac{p_{d,n}(\mathbf{x}) g_n(\mathbf{z}|\mathbf{x}) v_{n|n-1}(\mathbf{x})}{\kappa_n(\mathbf{z}) + \int p_{d,n}(\mathbf{x}) g_n(\mathbf{z}|\mathbf{x}) v_{n|n-1}(\mathbf{x}) d\mathbf{x}}. \quad (3.2.2)$$

ne sont pas calculables en pratique. Plusieurs implémentations sous optimales ont ainsi été proposées dans la littérature.

L'implémentation par mélange de gaussiennes [121] repose sur plusieurs hypothèses. Si le modèle est linéaire et gaussien ($f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n)$ et $g_n(\mathbf{y}_n|\mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n)$), si les probabilités de détection $p_{d,n}$ et de survie $p_{s,n}$ sont constantes, et si $v_{n-1}(\mathbf{x})$ et $\gamma_{n-1}(\mathbf{x})$ sont des mélanges gaussiens, alors $v_n(\mathbf{x})$ est un mélange gaussien dont les paramètres sont calculables à chaque instant (en effet, puisque $f_{n|n-1}(\mathbf{x}|\mathbf{x}_{n-1})$ est une gaussienne, dont la moyenne est linéaire en \mathbf{x}_{n-1} , le PHD $v_{n|n-1}$ dans (3.2.1) est encore un mélange gaussien ; en injectant ce nouveau mélange dans (3.2.2), v_n reste un mélange gaussien ; ce résultat est une conséquence des Lemmes A.1 et A.2 en Annexe A). Lorsque le modèle n'est plus linéaire, mais reste gaussien, des alternatives basées sur l'EKF ou l'UKF ont été proposées [29].

L'implémentation SMC du filtre PHD [122] [26] approxime le PHD par un mélange discret

$$\hat{v}_{n-1}(\mathbf{x}) = \sum_{i=1}^{L_{n-1}} w_{n-1}^{(i)} \delta_{\mathbf{x}_{n-1}^{(i)}}. \quad (3.2.3)$$

En injectant (3.2.3) dans (3.2.1), nous obtenons une approximation continue $\tilde{v}_{n|n-1}$ de $v_{n|n-1}$ pour laquelle nous pouvons déduire une nouvelle approximation MC de $v_{n|n-1}$ obtenue par échantillonnage d'importance, $\hat{v}_{n|n-1}(\mathbf{x}) = \sum_{i=1}^{L_n} w_{n|n-1}^{(i)} \delta_{\mathbf{x}_n^{(i)}}$, puis finalement une approximation \hat{v}_n de v_n donnée par

$$\hat{v}_n(\mathbf{x}) = \sum_{i=1}^{L_n} \underbrace{\left[(1 - p_{d,n}(\mathbf{x}_n^{(i)})) w_{n|n-1}^{(i)} + \sum_{\mathbf{z} \in \mathcal{Z}_n} \frac{p_{d,n}(\mathbf{x}_n^{(i)}) g_n(\mathbf{z}|\mathbf{x}_n^{(i)}) w_{n|n-1}^{(i)}}{\kappa_n(\mathbf{z}) + \sum_{i=1}^{L_n} p_{d,n}(\mathbf{x}_n^{(i)}) g_n(\mathbf{z}|\mathbf{x}_n^{(i)}) w_{n|n-1}^{(i)}} \right]}_{w_n^i} \delta_{\mathbf{x}_n^{(i)}}. \quad (3.2.4)$$

Des variantes de l'algorithme basées sur l'APF ont été également proposées [125], ainsi qu'un algorithme particulière gaussien, qui propage le PHD comme un mélange de gaussiennes dont les moyennes et les variances sont calculées à partir d'une méthode MC [28].

Analyse de l'implémentation SMC du filtre PHD

L'implémentation SMC du filtre PHD souffre des mêmes inconvénients que celle des FP présentés au Chapitre 2. Un terme crucial dans l'expression des poids dans (3.2.4) est $p_{d,n}(\mathbf{x}_n^{(i)}) g_n(\mathbf{z}|\mathbf{x}_n^{(i)}) w_{n|n-1}^{(i)}$ puisqu'il influe directement sur un estimateur $\hat{N}_n = \sum_{i=1}^{L_n} w_n^i$ de la masse totale $\int v_n(\mathbf{x}) d\mathbf{x}$ (qui est elle même

un estimateur du nombre d'objets). Ce terme dépend de la vraisemblance $g_n(\mathbf{z}|\mathbf{x}_n^{(i)})$. Par conséquent, si toutes les particules ont une vraisemblance faible avec toutes les mesures disponibles, l'estimateur MC du nombre d'objets \hat{N}_n a tendance à être proche de 0, même si certaines de ces mesures sont dues à la présence d'objets. Un tel scénario se produit dès lors que les particules ne sont pas concentrées dans des régions à forte vraisemblance, cette contrainte étant d'autant plus exigeante que la dimension des tirages est grande. Enfin, il est bien évidemment difficile d'obtenir un estimateur robuste de l'état caché d'un objet si peu de particules contribuent à la masse d'une région S où cet objet est présent. Pour résoudre cette difficulté, sans augmenter le nombre de tirages, certaines pistes sont exploitées dans cette section.

Dans un premier temps, il est possible d'essayer de tirer les particules selon une distribution d'importance qui guiderait les particules vers des régions à forte vraisemblance, comme en filtrage mono-objet (voir la discussion du Paragraphe 2.1.2). Dans le contexte multi-objets, cette idée est difficilement applicable car nous ne savons pas quelles sont les mesures dues à des objets détectés. Cependant, une implémentation auxiliaire peut résoudre ce problème en sélectionnant, de manière aléatoire, des mesures probablement dues à la présence d'objets [125].

Ensuite, comme nous l'avons rappelé dans l'introduction de ce chapitre, l'utilisation de tirages MC en plus petite dimension permet d'éviter la dispersion des particules.

Nous montrons dans ce paragraphe que ces deux solutions ne sont pas incompatibles dans le contexte multi-objets et nous les combinerons. Plus précisément, nous montrons que les équations du filtre PHD peuvent, dans certaines classes de modèles, se réécrire de telle manière (nous obtenons alors le M-PHD) que l'implémentation par mélange de gaussiennes et l'approche SMC peuvent être combinées. Pour ces classes de modèles, il est bien sûr possible d'utiliser l'approche SMC classique, mais en prenant en compte les spécificités du modèle, nous pouvons améliorer à la fois l'estimation du nombre d'objets ainsi que de leur état caché, sans augmenter le nombre de tirages. Nous nous focalisons d'abord sur le problème de réduction de dimension, puis nous proposons une implémentation Bootstrap ainsi qu'une implémentation auxiliaire de notre filtre M-PHD.

3.2.2 Le filtre M-PHD pour les modèles partiellement linéaires et gaussiens

Comme nous venons de le rappeler, l'estimation du PHD repose sur des approximations numériques ou sur des approximations basées sur des tirages probabilistes qui permettent de considérer des classes de modèles plus générales. Dans les modèles partiellement linéaires et gaussiens (3.1.8)-(3.1.10), nous allons proposer une approximation basée sur la décomposition

$$\int v_n(\mathbf{x})d\mathbf{x} = \int \left[\underbrace{\int v_n(\mathbf{x}^{nl}, \mathbf{x}^l)d\mathbf{x}^{nl}}_{\text{implémentation MC}} \right] d\mathbf{x}^l = \underbrace{\int \hat{v}_n(\mathbf{x}^l)d\mathbf{x}^l}_{\text{calculable exactement}}, \quad (3.2.5)$$

qui est cruciale pour l'estimation du nombre d'objets et des états cachés.

Premièrement, il convient d'explicitier les hypothèses sous lesquelles il est possible de proposer une version marginalisée du filtre PHD :

- H.1** les probabilités de survie $p_{s,n}(\mathbf{x}) = p_{s,n}(\mathbf{x}^{nl})$ et de détection $p_{d,n}(\mathbf{x}) = p_{d,n}(\mathbf{x}^{nl})$ dépendent seulement de la composante non linéaire du vecteur d'état d'un objet ;
- H.2** soient $p_{i,n}(\mathbf{x})$ des lois de probabilité données, pour $1 \leq i \leq N_{\gamma_n}$. Nous supposons que $\gamma_n(\mathbf{x})$ est

un mélange de la forme

$$\gamma_n(\mathbf{x}) = \sum_{i=1}^{N\gamma_n} w_{n|n-1}^{2,(i)} p_{i,n}(\mathbf{x}^{\text{nl}}) \mathcal{N}(\mathbf{x}^{\text{l}}; \mathbf{m}_{n|n-1}^{2,l,(i)}(\mathbf{x}^{\text{nl}}); \mathbf{P}_{n|n-1}^{2,l,(i)}) \quad (3.2.6)$$

(l'exposant 2 caractérisera ici les composantes naissantes). Notons que γ_n n'est pas nécessairement un mélange gaussien.

L'idée sur laquelle repose notre approche est de supposer qu'à l'instant n , le PHD est approché par un mélange Normal-Discret (ND),

$$\hat{v}_{n-1}(\mathbf{x}) = \sum_{i=1}^{L_{n-1}} w_{n-1}^{(i)} \mathcal{N}(\mathbf{x}^{\text{l}}; \mathbf{m}_{n-1}^{1,(i)}; \mathbf{P}_{n-1}^{1,(i)}) \delta_{\mathbf{x}_{n-1}^{\text{nl},(i)}}(\mathbf{x}^{\text{nl}}). \quad (3.2.7)$$

En d'autres termes, pour \mathbf{x}^{l} fixé, le PHD est approché par une mesure discrète dont le support est donné par l'ensemble $A_{n-1} = \{\mathbf{x}_{n-1}^{\text{nl},(i)}\}_{i=1}^{L_{n-1}}$ (il est possible d'avoir $\mathbf{x}_{n-1}^{\text{nl},(i)} = \mathbf{x}_{n-1}^{\text{nl},(j)}$ pour $i \neq j$); réciproquement, pour un élément donné de A_{n-1} , nous supposons que le PHD est approximé par un mélange gaussien. De plus, l'approximation (3.2.7) permet d'utiliser la décomposition (3.2.5) pour le calcul d'un estimateur du nombre d'objets et est propageable, en deux temps, dans les modèles (3.1.8)-(3.1.10).

Étape de Prédiction

Soient $\mathbf{F}_n^{1,(i)} \triangleq \mathbf{F}_n^{\text{nl},(i)}$, $\mathbf{F}_n^{\text{nl},(i)} \triangleq \mathbf{F}_n^{\text{nl}}(\mathbf{x}_{n-1}^{\text{nl},(i)})$, $\mathbf{f}_n^{1,(i)} \triangleq \mathbf{f}_n^{\text{nl},(i)}$, $\mathbf{f}_n^{\text{nl},(i)} \triangleq \mathbf{f}_n^{\text{nl}}(\mathbf{x}_{n-1}^{\text{nl},(i)})$ dans (3.1.8)-(3.1.10) et $p_{s,n}^{(i)} \triangleq p_{s,n}(\mathbf{x}_{n-1}^{\text{nl},(i)})$. Soit \hat{v}_{n-1} dans (3.2.7) une approximation ND du PHD v_{n-1} . En injectant (3.2.7) dans (3.2.1), puis en utilisant les Lemmes A.1 et A.2 en Annexe A, nous obtenons l'approximation $\tilde{v}_{n|n-1}$ de $v_{n|n-1}$,

$$\tilde{v}_{n|n-1}(\mathbf{x}) = \tilde{v}_{n|n-1}^1(\mathbf{x}) + \tilde{v}_{n|n-1}^2(\mathbf{x}), \quad (3.2.8)$$

$$\tilde{v}_{n|n-1}^1(\mathbf{x}) = \sum_{i=1}^{L_{n-1}} w_{n|n-1}^{1,(i)} \mathcal{N}(\mathbf{x}^{\text{l}}; \mathbf{m}_{n|n-1}^{1,l,(i)}(\mathbf{x}^{\text{nl}}); \mathbf{P}_{n|n-1}^{1,l,(i)}) \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{1,nl,(i)}; \mathbf{P}_{n|n-1}^{1,nl,(i)}), \quad (3.2.9)$$

$$\tilde{v}_{n|n-1}^2(\mathbf{x}) = \gamma_n(\mathbf{x}), \quad (3.2.10)$$

où

$$w_{n|n-1}^{1,(i)} = p_{s,n}^{(i)} w_{n-1}^{(i)}, \quad (3.2.11)$$

$$\mathbf{m}_{n|n-1}^{1,nl,(i)} = \mathbf{f}_n^{\text{nl},(i)} + \mathbf{F}_n^{\text{nl},(i)} \mathbf{m}_{n-1}^{1,(i)}, \quad (3.2.12)$$

$$\mathbf{P}_{n|n-1}^{1,nl,(i)} = \mathbf{Q}_n^{\text{nl}} + \mathbf{F}_n^{\text{nl},(i)} \mathbf{P}_{n-1}^{1,(i)} \mathbf{F}_n^{\text{nl},(i)T}, \quad (3.2.13)$$

$$\mathbf{m}_{n|n-1}^{1,l,(i)}(\mathbf{x}^{\text{nl}}) = \mathbf{A}_n^{(i)} \tilde{\mathbf{m}}_{n|n-1}^{1,l,(i)}(\mathbf{x}^{\text{nl}}) + \mathbf{f}_n^{1,(i)} + \tilde{\mathbf{Q}}_n^{\text{nl}}(\mathbf{x}^{\text{nl}} - \mathbf{f}_n^{\text{nl},(i)}), \quad (3.2.14)$$

$$\mathbf{P}_{n|n-1}^{1,l,(i)} = \mathbf{Q}_n^{\text{l}} - (\tilde{\mathbf{Q}}_n^{\text{nl}})^{-1} \mathbf{Q}_n^{\text{nl}T} + \mathbf{A}_n^{(i)} \tilde{\mathbf{P}}_{n|n-1}^{1,l,(i)} \mathbf{A}_n^{(i)T}, \quad (3.2.15)$$

où $\widetilde{\mathbf{m}}_{n|n-1}^{1,(i)}(\mathbf{x}^{\text{nl}})$, $\widetilde{\mathbf{P}}_{n|n-1}^{1,(i)}$ et $\mathbf{A}_n^{(i)}$ sont définis par

$$\widetilde{\mathbf{m}}_{n|n-1}^{1,(i)}(\mathbf{x}^{\text{nl}}) = \mathbf{m}_{n-1}^{1,(i)} + \widetilde{\mathbf{K}}_{n|n-1}^{1,(i)} \left[\mathbf{x}^{\text{nl}} - f_n^{\text{nl},(i)} - \mathbf{F}_n^{\text{nl},(i)} \mathbf{m}_{n-1}^{1,(i)} \right], \quad (3.2.16)$$

$$\widetilde{\mathbf{P}}_{n|n-1}^{1,(i)} = \left[\mathbf{I} - \widetilde{\mathbf{K}}_{n|n-1}^{1,(i)} \mathbf{F}_n^{\text{nl},(i)} \right] \mathbf{P}_{n-1}^{1,(i)}, \quad (3.2.17)$$

$$\mathbf{A}_n^{(i)} = \mathbf{F}_n^{1,(i)} - (\widetilde{\mathbf{Q}}_n^{\text{nl}})^{-1} \mathbf{F}_n^{\text{nl},(i)}, \quad (3.2.18)$$

$$\widetilde{\mathbf{K}}_{n|n-1}^{1,(i)} = \mathbf{P}_{n-1}^{1,(i)} (\mathbf{F}_n^{\text{nl},(i)})^T (\mathbf{P}_{n|n-1}^{\text{nl},(i)})^{-1}, \quad (3.2.19)$$

$$\widetilde{\mathbf{Q}}_n^{\text{nl}} = \mathbf{S}_n^{\text{nl}} (\mathbf{Q}_n^{\text{nl}})^{-1}. \quad (3.2.20)$$

Ensuite, nous cherchons une approximation \widetilde{v}_n de v_n .

Étape de mise à jour

Posons $\mathcal{Z}'_n = \{\theta\} \cup \mathcal{Z}_n$, où θ correspond à la mesure fictive de non détection ; cela facilitera la lecture des équations suivantes. En injectant (3.2.8) dans (3.2.2), puis en utilisant le Lemme A.2 en Annexe A, nous obtenons l'approximation suivante de v_n :

$$\widetilde{v}_n(\mathbf{x}) = \widetilde{v}_n^1(\mathbf{x}) + \widetilde{v}_n^2(\mathbf{x}), \quad (3.2.21)$$

où

$$\widetilde{v}_n^1(\mathbf{x}) = \sum_{i=1}^{L_{n-1}} \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{1,\text{nl},(i)}; \mathbf{P}_{n|n-1}^{1,\text{nl},(i)}) \sum_{\mathbf{z} \in \mathcal{Z}'_n} \Phi_n^{1,(i)}(\mathbf{x}^{\text{nl}}, \mathbf{x}^1, \mathbf{z}), \quad (3.2.22)$$

$$\widetilde{v}_n^2(\mathbf{x}) = \sum_{i=1}^{N_{\gamma_n}} p_{i,n}(\mathbf{x}^{\text{nl}}) \sum_{\mathbf{z} \in \mathcal{Z}'_n} \Phi_n^{2,(i)}(\mathbf{x}^{\text{nl}}, \mathbf{x}^1, \mathbf{z}), \quad (3.2.23)$$

$$\Phi_n^{j,(i)}(\mathbf{x}^{\text{nl}}, \mathbf{x}^1, \mathbf{z}) = \mathcal{N}(\mathbf{x}^1; \mathbf{m}_n^{j,1,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}); \mathbf{P}_n^{j,1,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}})) w_n^{j,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) \text{ pour } j = 1, 2, \quad (3.2.24)$$

avec, pour $j = 1, 2$, et $\mathbf{z} \in \mathcal{Z}'_n$,

$$w_n^{j,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) = \begin{cases} (1 - p_{d,n}(\mathbf{x}^{\text{nl}})) w_{n|n-1}^{j,(i)}, & \mathbf{z} = \theta, \\ \frac{p_{d,n}(\mathbf{x}^{\text{nl}}) q_n^{j,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) w_{n|n-1}^{j,(i)}}{\kappa_n(\mathbf{z}) + \widetilde{B}^1(\mathbf{z}) + \widetilde{B}^2(\mathbf{z})}, & \mathbf{z} \in \mathcal{Z}_n, \end{cases} \quad (3.2.25)$$

$$\mathbf{m}_n^{j,1,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) = \begin{cases} \mathbf{m}_{n|n-1}^{j,1,(i)}(\mathbf{x}^{\text{nl}}), & \mathbf{z} = \theta, \\ \mathbf{m}_{n|n-1}^{j,1,(i)}(\mathbf{x}^{\text{nl}}) + \mathbf{K}_n^{j,(i)}(\mathbf{x}^{\text{nl}}) \widetilde{\mathbf{z}}_n^{(i)}(\mathbf{x}^{\text{nl}}), & \mathbf{z} \in \mathcal{Z}_n, \end{cases} \quad (3.2.26)$$

$$\mathbf{P}_n^{j,1,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) = \begin{cases} \mathbf{P}_{n|n-1}^{j,1,(i)}, & \mathbf{z} = \theta, \\ \left[\mathbf{I} - \mathbf{K}_n^{j,(i)}(\mathbf{x}^{\text{nl}}) \mathbf{H}_n(\mathbf{x}^{\text{nl}}) \right] \mathbf{P}_{n|n-1}^{j,1,(i)}, & \mathbf{z} \in \mathcal{Z}_n, \end{cases} \quad (3.2.27)$$

où

$$q_n^{j,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) = \mathcal{N}(\mathbf{z}; h_n(\mathbf{x}^{\text{nl}}) + \mathbf{H}_n(\mathbf{x}^{\text{nl}}) \mathbf{m}_{n|n-1}^{j,1,(i)}(\mathbf{x}^{\text{nl}}); \mathbf{S}_n^{j,(i)}(\mathbf{x}^{\text{nl}})), \quad (3.2.28)$$

$$\mathbf{K}_n^{j,(i)}(\mathbf{x}^{\text{nl}}) = \mathbf{P}_{n|n-1}^{j,1,(i)} \mathbf{H}_n(\mathbf{x}^{\text{nl}})^T (\mathbf{S}_n^{j,(i)}(\mathbf{x}^{\text{nl}}))^{-1}, \quad (3.2.29)$$

$$\widetilde{\mathbf{z}}_n^{j,(i)}(\mathbf{x}^{\text{nl}}) = \mathbf{z} - \mathbf{H}_n(\mathbf{x}^{\text{nl}}) \mathbf{m}_{n|n-1}^{j,1,(i)}(\mathbf{x}^{\text{nl}}) - h_n(\mathbf{x}^{\text{nl}}), \quad (3.2.30)$$

$$\mathbf{S}_n^{j,(i)}(\mathbf{x}^{\text{nl}}) = \mathbf{H}_n(\mathbf{x}^{\text{nl}}) \mathbf{P}_{n|n-1}^{j,1,(i)} (\mathbf{H}_n(\mathbf{x}^{\text{nl}}))^T + \mathbf{R}_n, \quad (3.2.31)$$

$$\tilde{B}^1(\mathbf{z}) = \sum_{i=1}^{L_n} \int w_{n|n-1}^{1,(i)} p_{d,n}(\mathbf{x}^{\text{nl}}) q_n^{1,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{1,n,(i)}; \mathbf{P}_{n|n-1}^{1,n,(i)}) d\mathbf{x}^{\text{nl}}, \quad (3.2.32)$$

$$\tilde{B}^2(\mathbf{z}) = \sum_{i=1}^{N_{\gamma_n}} \int w_{n|n-1}^{2,(i)} p_{d,n}(\mathbf{x}^{\text{nl}}) q_n^{2,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) p_{i,n}(\mathbf{x}^{\text{nl}}) d\mathbf{x}^{\text{nl}}. \quad (3.2.33)$$

Remarque 3.1. Les approximations obtenues sont valables sous l'hypothèse **H.1** ci-dessus. Les résultats sont cependant extensibles dans le cas où $p_{s,n}(\mathbf{x})$ et $p_{d,n}(\mathbf{x})$ sont des mélanges de la forme (voir par exemple [121])

$$p_{s,n}(\mathbf{x}) = w_{s,n}^{(0)} p_{s,0,n}(\mathbf{x}^{\text{nl}}) + \sum_{i=1}^{N_{s,n}} w_{s,n}^{(i)} p_{s,i,n}(\mathbf{x}^{\text{nl}}) \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{s,n}^{1,(i)}(\mathbf{x}^{\text{nl}}); \mathbf{P}_{s,n}^{1,(i)}), \quad (3.2.34)$$

$$p_{d,n}(\mathbf{x}) = w_{d,n}^{(0)} p_{d,0,n}(\mathbf{x}^{\text{nl}}) + \sum_{i=1}^{N_{d,n}} w_{d,n}^{(i)} p_{d,i,n}(\mathbf{x}^{\text{nl}}) \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{d,n}^{1,(i)}(\mathbf{x}^{\text{nl}}); \mathbf{P}_{d,n}^{1,(i)}), \quad (3.2.35)$$

tels que $0 \leq p_{s,n}(\mathbf{x}) \leq 1$ et $0 \leq p_{d,n}(\mathbf{x}) \leq 1$.

3.2.3 Implémentation pratique du filtre M-PHD

À présent, nous nous intéressons au calcul pratique de $\int \tilde{v}_n(\mathbf{x}) d\mathbf{x}$, qui n'est pas calculable directement. Néanmoins, nous pouvons remarquer que $\int \tilde{v}_n(\mathbf{x}) d\mathbf{x}^1$ est calculable puisque $\int \Phi_n^{j,(i)}(\mathbf{x}^{\text{nl}}, \mathbf{x}^1, \mathbf{z}) d\mathbf{x}^1$ est réduit à $w_n^{j,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}})$. Il reste donc à calculer $\int [\int \tilde{v}_n(\mathbf{x}) d\mathbf{x}^1] d\mathbf{x}^{\text{nl}}$; deux implémentations sont proposées : une implémentation Bootstrap, qui ne nécessite pas d'hypothèse supplémentaires, et une implémentation auxiliaire.

Une implémentation Bootstrap du M-PHD

Cette implémentation repose sur le fait qu'il est possible de déduire aisément une approximation MC de $\tilde{v}_n^1(\mathbf{x})$, terme qui est dû aux objets persistants, ainsi que de $\tilde{v}_n^2(\mathbf{x})$, terme qui est dû aux objets naissants, par le mécanisme d'échantillonnage d'importance. Ici, les lois $\mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{1,\text{nl},(i)}; \mathbf{P}_{n|n-1}^{1,\text{nl},(i)})$ et $p_{i,n}$ sont disponibles et peuvent être utilisées pour tirer le nouveau jeu de particules.

Approximation ND de \tilde{v}_n^1 : une approximation ND de \tilde{v}_n^1 à L_n composantes est donnée par

$$\hat{v}_n^1(\mathbf{x}^{\text{nl}}, \mathbf{x}^1) = \sum_{i=1}^{L_n-1} \sum_{\mathbf{z} \in \mathcal{Z}'_n} \Phi_n^{1,(i)}(\mathbf{x}_n^{1,\text{nl},(i)}, \mathbf{x}^1, \mathbf{z}) \delta_{\mathbf{x}_n^{1,\text{nl},(i)}}(\mathbf{x}^{\text{nl}}), \quad (3.2.36)$$

avec $\mathbf{x}_n^{1,\text{nl},(i)} \sim \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{1,\text{nl},(i)}; \mathbf{P}_{n|n-1}^{1,\text{nl},(i)})$ pour $1 \leq i \leq L_n$.

Approximation ND de \tilde{v}_n^2 : ensuite, nous tirons des particules pour les objets naissants. Il peut être dangereux de tirer seulement N_{γ_n} particules si N_{γ_n} est petit. Nous utilisons donc une approximation de \tilde{v}_n^2 basée sur un ensemble de $L_n^2 = M \times N_{\gamma_n}$ particules. Pour $i = 1, \dots, L_n^2$, posons $j^i = \lceil i/M \rceil$ ($\lceil x \rceil$ désigne l'entier le plus proche supérieur ou égal à x). Une approximation ND de \tilde{v}_n^2 est alors donnée par

$$\hat{v}_n^2(\mathbf{x}^{\text{nl}}, \mathbf{x}^1) = \frac{1}{K} \sum_{i=1}^{L_n^2} \sum_{\mathbf{z} \in \mathcal{Z}'_n} \Phi_n^{2,(j^i)}(\mathbf{x}_n^{2,\text{nl},(i)}, \mathbf{x}^1, \mathbf{z}) \delta_{\mathbf{x}_n^{2,\text{nl},(i)}}(\mathbf{x}^{\text{nl}}), \quad (3.2.37)$$

où $\mathbf{x}_n^{2,\text{nl},(i)} \sim p_{j^i,n}(\mathbf{x}^{\text{nl}})$.

Nombre de composantes du mélange ND : notons $L_{n|n-1} = L_{n-1} + L_{n-1}^2$ le nombre total de particules tirées. En réunissant les approximations ND \hat{v}_n^1 et \hat{v}_n^2 , une approximation ND du PHD à l'instant n est donnée par

$$\hat{v}_n(\mathbf{x}^{\text{nl}}, \mathbf{x}^1) = \sum_{i=1}^{L_{n|n-1}} \sum_{\mathbf{z} \in \mathcal{Z}'_n} \Phi_n^{(i)}(\mathbf{x}_n^{\text{nl},(i)}, \mathbf{x}^1, \mathbf{z}) \delta_{\mathbf{x}_n^{\text{nl},(i)}}(\mathbf{x}^{\text{nl}}). \quad (3.2.38)$$

D'après (3.2.24), nous pouvons écrire

$$\Phi_n^{(i)}(\mathbf{x}_n^{\text{nl},(i)}, \mathbf{x}^1, \mathbf{z}) = w_n^{(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)}) \mathcal{N}(\mathbf{x}^1; \mathbf{m}_n^{1,(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)}); \mathbf{P}_n^{1,(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)})),$$

où $w_n^{(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)})$, pour $i = 1, \dots, L_{n|n-1}$ et pour $\mathbf{z} \in \mathcal{Z}'_n$, sont les poids associés aux tirages. Notre approximation finale $\hat{v}_n(\mathbf{x}^{\text{nl}}, \mathbf{x}^1)$ est un mélange ND de $L_{n|n-1}(1 + |\mathcal{Z}_n|)$ composantes. Par conséquent, le nombre de composantes croît exponentiellement au cours de temps. Comme dans [122], une étape de rééchantillonnage selon les poids $w_n^{(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)})$ peut être exécutée afin de réduire le nombre de composantes à un nombre L_n proportionnel à l'estimateur du nombre d'objets. Dans le cas particulier où les mesures ne dépendent pas de la composante linéaire \mathbf{x}^1 du vecteur d'état, l'approximation ND du PHD devient

$$\hat{v}_n(\mathbf{x}^{\text{nl}}, \mathbf{x}^1) = \sum_{i=1}^{L_{n|n-1}} w_n^{(i)}(\mathbf{x}^{\text{nl}}) \mathcal{N}(\mathbf{x}^1; \mathbf{m}_n^{1,(i)}(\mathbf{x}_n^{\text{nl},(i)}); \mathbf{P}_n^{1,(i)}(\mathbf{x}_n^{\text{nl},(i)})) \delta_{\mathbf{x}_n^{\text{nl},(i)}}(\mathbf{x}^{\text{nl}}). \quad (3.2.39)$$

où $\mathbf{m}_n^{1,(i)}(\mathbf{x}_n^{\text{nl},(i)}) = \mathbf{m}_n^{1,(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)})$, $\mathbf{P}_n^{1,(i)}(\mathbf{x}_n^{\text{nl},(i)}) = \mathbf{P}_n^{1,(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)})$ et $w_n^{(i)}(\mathbf{x}_n^{\text{nl},(i)}) = \sum_{\mathbf{z} \in \mathcal{Z}'_n} w_n^{(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)})$. Dans ce cas, le nombre de composantes du mélange ND est donné par $L_{n|n-1}$. Le rééchantillonnage peut être alors exécuté dans le but de supprimer les particules de poids faibles et de produire un jeu de points de taille proportionnelle à l'estimateur du nombre d'objets.

Enfin, nous pouvons noter, d'après (3.2.25), que le calcul des poids $w_n^{(i)}(\mathbf{z}, \mathbf{x}_n^{\text{nl},(i)})$ requiert celui de $\tilde{B}^j(\mathbf{z})$ dans (3.2.32) et (3.2.33). Ces quantités peuvent être approchées en utilisant le jeu de particules qui vient d'être tiré :

$$\hat{B}^1(\mathbf{z}) = \sum_{i=1}^{L_{n-1}} w_{n|n-1}^{1,(i)} p_{d,n}(\mathbf{x}_n^{1,\text{nl},(i)}) q_n^{1,(i)}(\mathbf{z}, \mathbf{x}_n^{1,\text{nl},(i)}), \quad (3.2.40)$$

$$\hat{B}^2(\mathbf{z}) = \frac{1}{K} \sum_{i=1}^{L_n^2} w_{n|n-1}^{2,(i)} p_{d,n}(\mathbf{x}_n^{2,\text{nl},(i)}) q_n^{2,(i)}(\mathbf{z}, \mathbf{x}_n^{2,\text{nl},(i)}). \quad (3.2.41)$$

Une implémentation marginale auxiliaire du PHD

Même si l'implémentation précédente n'utilise des particules que pour la composante non linéaire du vecteur d'état, elle peut être inefficace car les mesures disponibles \mathcal{Z}_n ne sont pas prises en compte, le risque étant d'avoir des particules dans des régions où les objets ne sont pas présents. Pour prendre en compte ce problème, nous proposons une implémentation auxiliaire dont le but est de guider les particules dans des régions prometteuses, où les objets sont probablement présents. Comme nous l'avons vu à la Section 2.3, une distribution d'importance efficace doit être utilisée, sans quoi l'implémentation auxiliaire peut dégrader les performances de l'estimateur associé (voir les Paragraphes 2.3.4 et 2.3.5). Afin de proposer une implémentation auxiliaire efficace, nous supposons à présent que

H'1 la probabilité de détection $p_{d,n}(\mathbf{x}^{\text{nl}}) = p_{d,n}$ est constante ;

H'2 l'intensité γ_n dans (3.2.6) est un mélange gaussien, $p_{i,n}(\mathbf{x}^{\text{nl}}) = \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{2,\text{nl},(i)}; \mathbf{P}_{n|n-1}^{2,\text{nl},(i)})$ et $\mathbf{m}_{n|n-1}^{2,1,(i)}(\mathbf{x}^{\text{nl}})$ linéaire en \mathbf{x}^{nl} dans (3.2.6).

Réécrivons alors $\tilde{v}_n(\mathbf{x})$ sous une forme plus directe dans laquelle nous ne distinguons plus les objets persistants et naissants. En posant, ici, $L_{n|n-1} = L_{n-1} + N_{\gamma_n}$, \tilde{v}_n s'écrit

$$\tilde{v}_n(\mathbf{x}) = \sum_{i=1}^{L_{n|n-1}} \sum_{\mathbf{z} \in \mathcal{Z}'_n} \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)}) \Phi_n^{(i)}(\mathbf{x}^{\text{nl}}, \mathbf{x}^1, \mathbf{z}),$$

où $\mathbf{m}_{n|n-1}^{\text{nl},(i)}$, $\mathbf{P}_{n|n-1}^{\text{nl},(i)}$, $\Phi_n^{(i)}(\mathbf{x}^{\text{nl}}, \mathbf{x}^1, \mathbf{z})$ sont respectivement égaux à $\mathbf{m}_{n|n-1}^{j,\text{nl},(i)}$, $\mathbf{P}_{n|n-1}^{j,\text{nl},(i)}$, $\Phi_n^{j,(i)}(\mathbf{x}^{\text{nl}}, \mathbf{x}^1, \mathbf{z})$ avec $j = 1$ si $1 \leq i \leq L_{n-1}$ et $j = 2$ si $L_{n-1} + 1 \leq i \leq L_{n|n-1}$. En marginalisant par rapport à la partie linéaire \mathbf{x}^1 , nous obtenons

$$\tilde{v}_n(\mathbf{x}^{\text{nl}}) = \sum_{i=1}^{L_{n|n-1}} \sum_{\mathbf{z} \in \mathcal{Z}'_n} \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)}) w_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}});$$

où $w_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) = w_n^{1,(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}})$ pour $i = 1, \dots, L_{n-1}$ et $w_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) = w_n^{2,(i-L_{n-1})}(\mathbf{z}, \mathbf{x}^{\text{nl}})$ pour $i = L_{n-1} + 1, \dots, L_{n|n-1} + N_{\gamma_n}$.

Nous devons alors tirer L_n points selon $\tilde{v}_n(\mathbf{x}^{\text{nl}})$, qui peut être vu comme un mélange de mélanges. Une implémentation auxiliaire repose alors sur deux variables auxiliaires : la première représente une composante due à un objet persistant ou naissant ; la seconde représente une mesure \mathbf{z} . Par conséquent, pour $1 \leq j \leq L_n$, nous choisissons aléatoirement une composante $i^j \in \{1, \dots, L_{n|n-1}\}$, une mesure $\mathbf{z}^j \in \mathcal{Z}'_n$ et enfin un nouveau point $\mathbf{x}_n^{\text{nl},(j)}$. Les variables auxiliaires et les particules obtenues sont tirées indépendamment et identiquement suivant une distribution d'importance $q(\mathbf{z}, i, \mathbf{x}^{\text{nl}})$. Les poids associés à chaque particule sont donc déduits de

$$w(\mathbf{z}, i, \mathbf{x}^{\text{nl}}) = \frac{\mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)}) w_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}})}{q(\mathbf{z}, i, \mathbf{x}^{\text{nl}})}. \quad (3.2.42)$$

Comme nous l'avons vu dans le Paragraphe 2.3, la loi $q(\mathbf{z}, i, \mathbf{x}^{\text{nl}})$ doit être choisie correctement. La loi optimale (qui minimise ici la variance conditionnelle des poids d'importance) est donnée par

$$q^{\text{opt}}(\mathbf{z}, i, \mathbf{x}^{\text{nl}}) \propto \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)}) w_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}). \quad (3.2.43)$$

Cette loi n'est pas calculable car la dépendance de $w_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}})$ en \mathbf{x}^{nl} est due au terme $q_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}})$ dans (3.2.28) ; or, si $\mathbf{h}(\mathbf{x}^{\text{nl}})$ et $\mathbf{H}(\mathbf{x}^{\text{nl}})$ ne sont pas linéaires en \mathbf{x}^{nl} , alors l'intégrale $\int \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)}) \times q_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) d\mathbf{x}^{\text{nl}}$ n'est pas calculable. Il est néanmoins possible d'approcher la loi optimale en utilisant les approximations suivantes :

$$p^{(i)}(\mathbf{z}) \approx \int q_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)}) d\mathbf{x}^{\text{nl}}, \quad (3.2.44)$$

$$p^{(i)}(\mathbf{x}^{\text{nl}}|\mathbf{z}) \tilde{\propto} q_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)}) \quad (3.2.45)$$

où $\tilde{\propto}$ signifie « approximativement proportionnel à ». Comme nous l'avons rappelé dans le Paragraphe 3.2.5, les approximations gaussiennes $p^{(i)}(\mathbf{z})$ et $p^{(i)}(\mathbf{x}^{\text{nl}}|\mathbf{z})$ peuvent être obtenues par linéarisation de $\mathbf{h}(\cdot)$ ou $\mathbf{H}(\cdot)$ ou par la transformation UT [125] [43] [66] ; nous pouvons employer ces approximations à condition que le support de la loi de transition et de la vraisemblance se chevauchent. Dans le cas contraire, la méthode

de la correction progressive permet de trouver les approximations gaussiennes dans (3.2.44) et (3.2.45) [86]. À présent, nous détaillons précisément comment obtenir une particule $(i^j, \mathbf{z}^j, \mathbf{x}_n^{\text{nl},(j)}) \sim q(i) \times q(\mathbf{z}|i) \times q(\mathbf{x}^{\text{nl}}|i, \mathbf{z})$.

Sélection des anciennes composantes : la première étape consiste à sélectionner un index i dans l'ensemble $\{1, \dots, L_{n|n-1}\}$. Si $i \in \{1, \dots, L_{n-1}\}$, alors nous avons sélectionné une composante qui doit être tirée conditionnellement à la i -ème particule $\mathbf{x}_{n-1}^{\text{nl},(i)}$. Si $i \in \{L_{n-1} + 1, \dots, L_{n-1} + N_{\gamma_n}\}$, alors la composante est tirée suivant la $(i - L_{n-1})$ -ème composante du mélange de l'intensité de naissance. Pour $\mathbf{z} \in \mathcal{Z}'_n$, définissons

$$\tilde{q}(\mathbf{z}|i) = \begin{cases} 1 - p_{d,n}, & \mathbf{z} = \theta, \\ p_{d,n} p^{(i)}(\mathbf{z}) / [\kappa_n(\mathbf{z}) + \hat{B}(\mathbf{z})], & \mathbf{z} \in \mathcal{Z}_n, \end{cases} \quad (3.2.46)$$

où $\hat{B}(\mathbf{z})$ est une approximation de $\tilde{B}_1(\mathbf{z}) + \tilde{B}_2(\mathbf{z})$. Le calcul de $\hat{B}(\mathbf{z})$ sera discuté ci-dessous. La densité d'importance associée à la sélection de la composante est donnée par

$$q(i) = w_{n|n-1}^{(i)} \sum_{\mathbf{z} \in \mathcal{Z}'_n} \tilde{q}(\mathbf{z}|i) / C_n, \quad (3.2.47)$$

$$C_n = \sum_{i=1}^{L_{n|n-1}} w_{n|n-1}^{(i)} \sum_{\mathbf{z} \in \mathcal{Z}'_n} \tilde{q}(\mathbf{z}|i). \quad (3.2.48)$$

Remarquons que si les approximations (3.2.44)-(3.2.45) sont valides, alors les composantes tirées durant cette étape auront tendance à être liées aux objets persistants ou naissants ayant une vraisemblance importante avec les nouvelles mesures. Ainsi, contrairement à l'implémentation précédente, les nouvelles particules sont tirées en fonction des mesures disponibles.

Sélection d'une mesure : l'étape suivante consiste à sélectionner une mesure associée à la composante sélectionnée précédemment. Cela est fait en sélectionnant $\mathbf{z} \in \mathcal{Z}'_n$ suivant

$$q(\mathbf{z}|i) = \tilde{q}(\mathbf{z}|i) / \sum_{\mathbf{y} \in \mathcal{Z}'_n} \tilde{q}(\mathbf{y}|i). \quad (3.2.49)$$

Sélection d'une nouvelle particule : enfin, il nous reste à prolonger ou initialiser la i -ème trajectoire sélectionnée, associée à présent à la mesure \mathbf{z} qui vient d'être tirée, en tirant une particule suivant

$$q(\mathbf{x}^{\text{nl}}|i, \mathbf{z}) = \begin{cases} \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n+1|k}^{\text{nl},(i^j)}; \mathbf{P}_{n|n-1}^{\text{nl},(i^j)}), & \mathbf{z} = \theta, \\ p^{(i)}(\mathbf{x}^{\text{nl}}|\mathbf{z}), & \mathbf{z} \in \mathcal{Z}_n. \end{cases} \quad (3.2.50)$$

Calcul des poids : soit $(i^j, \mathbf{z}^j, \mathbf{x}_n^{\text{nl},(j)})$ le j -ème triplet (composante du mélange, mesure et particule finale) tiré. D'après (3.2.42), l'expression des poids est donnée par

$$w_n^{(j)} = \begin{cases} C_n / L_n, & \mathbf{z}^j = \theta, \\ \frac{C_n q_n^{(i^j)}(\mathbf{z}^j, \mathbf{x}_n^{\text{nl},(j)}) \mathcal{N}(\mathbf{x}_n^{\text{nl},(j)}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)})}{L_n p^{(i^j)}(\mathbf{z}^j) p^{(i^j)}(\mathbf{x}_n^{\text{nl},(j)}|\mathbf{z}^j)}, & \mathbf{z}^j \in \mathcal{Z}_n. \end{cases} \quad (3.2.51)$$

D'après (3.2.51), si les approximations (3.2.44) et (3.2.45) sont précises, les poids finaux sont à peu près égaux et $w_n^{(j)} \approx C_n/L_n$. Une approximation ND du PHD v_n est donnée par

$$\hat{v}_n(\mathbf{x}) = \sum_{j=1}^{L_n} w_n^{(j)} \mathcal{N}(\mathbf{x}^1; \mathbf{m}_n^{1,(j)}(\mathbf{z}^j, \mathbf{x}_n^{\text{nl},(j)}); \mathbf{P}_n^{1,(j)}(\mathbf{z}^j, \mathbf{x}_n^{\text{nl},(j)})) \delta_{\mathbf{x}_n^{\text{nl},(j)}}(\mathbf{x}^{\text{nl}}). \quad (3.2.52)$$

Dans l'implémentation Bootstrap, les quantités $\tilde{B}^j(\mathbf{z})$ et (3.2.32)-(3.2.33) peuvent être calculées à partir du jeu de particules tiré, avant d'être utilisées dans le calcul des poids. Ici, le calcul de $\tilde{B}^j(\mathbf{z})$ est nécessaire pour l'étape de tirage. Une implémentation reposant sur les mêmes distributions d'importance que ci-dessus peut être utilisée pour approcher $\tilde{B}^j(\mathbf{z})$. En particulier, pour $i = 1, \dots, L'$, nous tirons un index j^i tel que $\Pr(j^i = b) \propto w_{n|n-1}^{(b)} p^{(b)}(\mathbf{z})$, puis nous tirons $\tilde{\mathbf{x}}^{(i)} \sim p^{j^i}(\mathbf{x}^{\text{nl}}|\mathbf{z})$. Nous obtenons alors l'approximation

$$\hat{B}(\mathbf{z}) = \frac{\hat{B}_0(\mathbf{z})}{L'} \sum_{i=1}^{L'} \frac{\mathcal{N}(\tilde{\mathbf{x}}^{(i)}; \mathbf{m}_{n|n-1}^{\text{nl},(j^i)}; \mathbf{P}_{n|n-1}^{\text{nl},(j^i)}) q_n^{(j^i)}(\mathbf{z}, \tilde{\mathbf{x}}^{(i)})}{p^{j^i}(\mathbf{z}) p^{j^i}(\tilde{\mathbf{x}}^{(i)}|\mathbf{z})}, \quad (3.2.53)$$

où $\hat{B}_0(\mathbf{z}) = p_{d,n} \sum_{i=1}^{L_{n|n-1}} w_{n|n-1}^{(i)} p^{(i)}(\mathbf{z})$. Si les approximations (3.2.44) et (3.2.45) sont correctes, alors les termes de la somme de (3.2.53) ont tendance à être proches de 1 et une petite valeur pour L' peut être choisie. Dans ce cas, $\hat{B}(\mathbf{z}) \approx \hat{B}_0(\mathbf{z})$.

3.2.4 D'autres applications du filtre M-PHD

Jusqu'à présent, nous nous sommes intéressés aux modèles partiellement linéaires et gaussiens (3.1.12)-(3.1.13). Désormais, nous nous intéressons aux modèles à sauts linéaires et gaussiens (3.1.15)-(3.1.16) ainsi qu'à des modèles à sauts partiellement linéaires et gaussiens. Nous cherchons ici à propager le PHD en dimension augmentée $v_n(\mathbf{x}, r)$ puis à calculer $\sum_r \int v_n(\mathbf{x}, r) d\mathbf{x}$. Nous montrons alors que la propagation de $v_n(\mathbf{x}, r)$ peut être approchée par notre filtre M-PHD pour ces modèles, à quelques adaptations près.

Modèles à sauts linéaires et gaussiens

Considérons le modèle à sauts vérifiant (3.1.15)-(3.1.16). Nous supposons que les hypothèses suivantes sont vérifiées :

H.3 les probabilités de survie $p_{s,n}(r)$ et de détection $p_{d,n}(r)$ ne dépendent que du mode r ;

H.4 l'intensité des naissances est un mélange qui s'écrit

$$\gamma_n(\mathbf{x}, r) = \sum_{i=1}^{N_{\gamma_n}} w_{n|n-1}^{2,(i)} p_{i,n}(r) \mathcal{N}(\mathbf{x}; \mathbf{m}_{n|n-1}^{2,1,(i)}(r); \mathbf{P}_{n|n-1}^{2,1,(i)}(r)). \quad (3.2.54)$$

Dans ce modèle, si nous supposons que le PHD $v_n(\mathbf{x}, r)$ est un mélange gaussien, il est possible de montrer que ce mélange croît exponentiellement à cause du nombre de mesures disponibles, généralement supérieur à 1, ainsi qu'au nombre de valeurs possibles prises par la chaîne $\{R_n\}$. L'implémentation par mélange de gaussiennes [92] propage alors $v_n(\mathbf{x}, r)$ et est basée sur des approximations numériques de type fusion/élagage de gaussiennes. Nous proposons ici une alternative en adaptant notre M-PHD qui permet de réduire la croissance du mélange due aux différents modes.

Dans ces modèles, (3.2.1) et (3.2.2) restent vraies si nous considérons l'état augmenté $\mathbf{x} = (\mathbf{x}^1, r)$ [83]. Le rôle joué par la composante non linéaire est à présent tenu par la chaîne de Markov $\{R_n\}$; une implémentation marginale du PHD apparaît alors comme un cas particulier du cas précédent, dans lequel

nous remplaçons \mathbf{x}^{nl} par r . Par conséquent, nous propageons des particules qui agissent sur la composante discrète r du vecteur d'état, et nous propageons une approximation ND du PHD,

$$\hat{v}_{n-1}(\mathbf{x}, r) = \sum_{i=1}^{L_{n-1}} w_{n-1}^{(i)} \mathcal{N}(\mathbf{x}; \mathbf{m}_{n-1}^{(i)}; \mathbf{P}_{n-1}^{(i)}) \delta_{r_{n-1}^{(i)}}(r). \quad (3.2.55)$$

Partant de (3.2.55), les approximations prédites et corrigées (3.2.8) et (3.2.21), sont adaptées : les dépendances en $\mathbf{x}_n^{\text{nl},(i)}$ et \mathbf{x}^{nl} sont remplacées respectivement par $r_{n-1}^{(i)}$ et r ; la loi $\mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{1,\text{nl},(i)}; \mathbf{P}_{n|n-1}^{1,\text{nl},(i)})$ dans (3.2.8) devient alors $p(r|r_{n-1}^{(i)})$; $\mathbf{m}_{n|n-1}^{1,l,(i)}$ et $\mathbf{P}_{n|n-1}^{1,l,(i)}$ dans (3.2.14) et (3.2.15) sont réduits à $\mathbf{m}_{n|n-1}^{1,l,(i)} = \mathbf{F}_n^{1,(i)} \mathbf{m}_{n-1}^{1,(i)}$, et $\mathbf{P}_{n|n-1}^{1,l,(i)} = \mathbf{F}_n^{1,(i)} \mathbf{P}_n^{1,(i)} \mathbf{F}_n^{1,(i)T} + \mathbf{Q}_n^{1,(i)}$.

À partir de ce constat, les deux implémentations du filtre M-PHD proposées dans le Paragraphe 3.2.3 peuvent être utilisées, si ce n'est que nous tirons à présent des particules discrètes $\{r_n^{(i)}\}$:

- l'implémentation Bootstrap du Paragraphe 3.2.3 repose à présent sur les lois $p(r|r_n^{(i)})$ pour les objets persistants et sur $p_{i,n}(r)$ pour les objets naissants ;
- l'implémentation auxiliaire du Paragraphe 3.2.3 repose sur une distribution d'importance $q(\mathbf{z}, i, r)$; la distribution d'importance optimale est cependant calculable, contrairement au cas précédent, puisque les intégrales sur \mathbf{x}^{nl} qui n'étaient pas calculables sont maintenant remplacées par des sommes sur r .

Modèles à sauts partiellement linéaires et gaussiens

Enfin, nous adaptons notre algorithme marginalisé proposé au Paragraphe 3.2.2 à une classe de modèles plus générale. Nous considérons un modèle à sauts caractérisé par les transitions $p(r_n|r_{n-1})$, dans lequel le vecteur d'état multi-dimensionnel \mathbf{x}_n peut être décomposé en $\mathbf{x}_n = [\mathbf{x}_n^1, \mathbf{x}_n^{\text{nl}}]^T$ de telle sorte que, si nous notons $\zeta_n = (\mathbf{x}_{n-1}^{\text{nl}}, r_n)$ et $\psi_n = (\mathbf{x}_n^{\text{nl}}, r_n)$, les équations de transition et de vraisemblance du modèle s'écrivent

$$f_{n|n-1}(\mathbf{x}|\mathbf{x}_{n-1}, r_n) = \mathcal{N}\left(\mathbf{x}; \begin{bmatrix} \mathbf{f}_n^{\text{nl}}(\zeta_n) \\ \mathbf{f}_n^1(\zeta_n) \end{bmatrix} + \begin{bmatrix} \mathbf{F}_n^{\text{nl}}(\zeta_n) \\ \mathbf{F}_n^1(\zeta_n) \end{bmatrix} \mathbf{x}_{n-1}^1; \begin{bmatrix} \mathbf{Q}_n^{\text{nl}}(r_n) & (\mathbf{S}_n^{\text{nl}}(r_n))^T \\ (\mathbf{S}_n^{\text{nl}}(r_n)) & \mathbf{Q}_n^1(r_n) \end{bmatrix}\right), \quad (3.2.56)$$

$$g_n(\mathbf{z}|\mathbf{x}_n, r_n) = \mathcal{N}(\mathbf{z}; \mathbf{h}_n(\psi_n) + \mathbf{H}_n(\psi_n) \mathbf{x}_n^1; \mathbf{R}_n(r_n)). \quad (3.2.57)$$

Les hypothèses suivantes permettent d'adapter notre filtre M-PHD à cette classe de modèles :

- H.5** les probabilités de survie $p_{s,n}(\mathbf{x}^{\text{nl}}, r)$ et de détection $p_{d,n}(\mathbf{x}^{\text{nl}}, r)$ dépendent seulement du mode r ainsi que de la composante non linéaire du vecteur d'état ;
- H.6** l'intensité des naissances γ_n s'écrit

$$\gamma_n(\mathbf{x}, r) = \sum_{i=1}^{N_{\gamma_n}} w_{n|n-1}^{2,(i)} p_{i,n}(r) p_{i,n}(\mathbf{x}^{\text{nl}}|r) \mathcal{N}(\mathbf{x}^1; \mathbf{m}_{n|n-1}^{2,l,(i)}(r); \mathbf{P}_{n|n-1}^{2,l,(i)}(r)). \quad (3.2.58)$$

Sous ces hypothèses, la technique de marginalisation peut être de nouveau appliquée pour propager une approximation ND du PHD dans les modèles (3.2.56)-(3.2.57). Le vecteur d'état augmenté s'écrit $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^{\text{nl}}, r)$, avec comme composante non-linéaire $(\mathbf{x}^{\text{nl}}, r)$; nous propageons alors des particules en dimension augmentée $\{\mathbf{x}_n^{\text{nl},(i)}, r_n^{(i)}\}$. Partant d'une approximation ND du PHD à l'instant $n - 1$,

$$\hat{v}_{n-1}(\mathbf{x}, r) = \sum_{i=1}^{L_{n-1}} w_{n-1}^{(i)} \mathcal{N}(\mathbf{x}^1; \mathbf{m}_{n-1}^{1,(i)}; \mathbf{P}_{n-1}^{1,(i)}) \delta_{\mathbf{x}_{n-1}^{\text{nl},(i)}, r_{n-1}^{(i)}}(\mathbf{x}^{\text{nl}}, r), \quad (3.2.59)$$

les approximations (3.2.8)-(3.2.21) peuvent être utilisées à une adaptation près : les dépendances en $\mathbf{x}_{n-1}^{\text{nl},(i)}$ et \mathbf{x}^{nl} sont respectivement remplacées par $(\mathbf{x}_{n-1}^{\text{nl},(i)}, r_{n-1}^{(i)})$ et par $(\mathbf{x}^{\text{nl}}, r)$; la loi $\mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}; \mathbf{P}_{n|n-1}^{\text{nl},(i)})$ est remplacée par $p(r|r_{n-1}^{(i)}) \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}(r); \mathbf{P}_{n|n-1}^{\text{nl},(i)}(r))$. Les implémentations Bootstrap et auxiliaires peuvent alors être utilisées pour propager les particules $\{\mathbf{x}_n^{\text{nl},(i)}, r_n^{(i)}\}$:

- Dans l’implémentation Bootstrap du Paragraphe 3.2.3, les particules $(\mathbf{x}_n^{\text{nl},(i)}, r_n^{(i)})$ sont maintenant obtenues à partir de $p(r|r_{n-1}^{(i)}) \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}(r); \mathbf{P}_{n|n-1}^{\text{nl},(i)}(r))$ pour les objets persistants et à partir de $p_{i,n}(r)p_{i,n}(\mathbf{x}^{\text{nl}}|r)$ pour les objets naissants ;
- Dans l’implémentation auxiliaire, les particules sont tirées suivant une densité $q(\mathbf{z}, i, (\mathbf{x}^{\text{nl}}, r))$ qui prolonge celle développée dans le Paragraphe 3.2.3. Pour $r = 1, \dots, K$, nous supposons la disponibilité de l’approximation

$$\mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i)}(r); \mathbf{P}_{n|n-1}^{\text{nl},(i)}(r)) q_n^{(i)}(\mathbf{z}, \mathbf{x}^{\text{nl}}) \approx p^{(i)}(\mathbf{x}^{\text{nl}}|r, \mathbf{z}) p^{(i)}(\mathbf{z}|r) \quad (3.2.60)$$

où $\mathbf{m}_{n|n-1}^{\text{nl},(i)}(r)$ et $\mathbf{P}_{n|n-1}^{\text{nl},(i)}(r)$ sont les moyennes et covariances prédites sous le mode r , calculées à partir de (3.2.12) et (3.2.13). L’approximation (3.2.60) peut être obtenue à partir de la méthode de la correction progressive. Nous définissons alors

$$\tilde{q}(\mathbf{z}|i) = \begin{cases} 1 - p_{d,n}, & \mathbf{z} = \theta, \\ p_{d,n}/[\kappa_n(\mathbf{z}) + \widehat{B}(\mathbf{z})] \sum_{r=1}^K p^{(i)}(\mathbf{z}|r), & \mathbf{z} \in Z_n. \end{cases} \quad (3.2.61)$$

Les lois d’importance à partir desquelles sont tirées une composante ainsi qu’une mesure sont données par $q(i)$ et $q(\mathbf{z}|i)$ définies en (3.2.47) et (3.2.49), si ce n’est que nous utilisons (3.2.61) pour la définition de $\tilde{q}(\mathbf{z}|i)$. Le mode est alors tiré suivant

$$q(r|i, \mathbf{z}) \propto \begin{cases} p(r|r_{n-1}^{(i)}), & \mathbf{z} = \theta, \\ p(r|r_{n-1}^{(i)}) p^{(i)}(\mathbf{z}|r), & \mathbf{z} \in Z_n. \end{cases} \quad (3.2.62)$$

Enfin, la composante non linéaire est tirée suivant

$$q(\mathbf{x}^{\text{nl}}|i, \mathbf{z}, r) = \begin{cases} \mathcal{N}(\mathbf{x}^{\text{nl}}; \mathbf{m}_{n|n-1}^{\text{nl},(i^j)}(r); \mathbf{P}_{n|n-1}^{\text{nl},(i^j)}(r)), & \mathbf{z} = \theta, \\ p^{(i)}(\mathbf{x}^{\text{nl}}|r, \mathbf{z}), & \mathbf{z} \in Z_n. \end{cases} \quad (3.2.63)$$

Les poids associés sont calculés de la même façon que dans (3.2.51) à des ajustements près pour le mode.

3.2.5 Simulations

Nous comparons quatre algorithmes sur divers scénarios de poursuite :

1. l’implémentation Bootstrap du filtre PHD que nous notons ici (BF) [122] ; dans cette implémentation, le nombre de particules utilisées pour les objets persistants dépend d’un estimateur du nombre d’objets estimés. Pour un nombre de particules donné L_n , si \widehat{N}_{n-1} est cet estimateur du nombre d’objets à l’instant $n-1$, alors nous utilisons $L_n^1 = \min\{L_n, \widehat{N}_{n-1} \min\{500, L_n/20\}\}$ particules pour les objets persistants et $L_n^2 = L_n - L_n^1$ particules pour les objets naissants ; ainsi, l’algorithme peut être comparé avec des algorithmes utilisant un nombre de particules fixe ;

2. le filtre M-PHD avec implémentation Bootstrap (M-BF) ; il s'agit de l'implémentation développée dans les Paragraphes 3.2.3 et 3.2.4 ;
3. un filtre PHD basé sur une implémentation auxiliaire (APF) [125], à quelques différences près. Premièrement, les objets naissants sont tirés suivant un mélange gaussien plutôt qu'une unique gaussienne. Deuxièmement, la méthode de la correction progressive est utilisée pour approcher la loi d'importance optimale. Enfin, l'algorithme est adapté afin d'inclure les modèles à sauts (3.2.56)-(3.2.57) ;
4. notre M-PHD, basé sur l'implémentation auxiliaire (M-APF). Il s'agit de l'implémentation proposée dans les Paragraphes 3.2.3 et 3.2.4. Notre approximation de $\hat{B}(\mathbf{z})$ est basée sur (3.2.53) avec $L' = 100$.

Les deux implémentations auxiliaires utilisent à tout instant un nombre constant de particules. Les algorithmes sont comparés à partir du calcul de la distance OSPA, rappelée en (1.2.13), dans laquelle nous choisissons $\bar{c} = 100$ et $\bar{p} = 2$.

Scénario de poursuite angle-distance d'objets évoluant à vitesse constante

Dans ce scénario, les objets évoluent à vitesse constante dans la région $[-500, 500]^2$. Le modèle utilisé dans les filtres PHD est le suivant. Le vecteur d'état est $\mathbf{x}_n = [p_{x,n}, \dot{p}_{x,n}, p_{y,n}, \dot{p}_{y,n}]^T$, où $(p_{x,n}, p_{y,n})$ est la position et $(\dot{p}_{x,n}, \dot{p}_{y,n})$ la vitesse, en coordonnées cartésiennes. La transition de la loi d'évolution à l'instant n d'un objet est

$$f(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}\mathbf{x}_{n-1}; \mathbf{Q}), \quad (3.2.64)$$

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \sigma^2 \mathbf{I}_2 \otimes \begin{bmatrix} T_s^3/3 & T_s^2/2 \\ T_s^2/2 & T_s \end{bmatrix}. \quad (3.2.65)$$

Nous fixons $p_{s,n} = 0.95$, $T_s = 1$ et $\sigma^2 = 1/625$. L'intensité des naissances est

$$\gamma(\mathbf{x}_n) = \lambda_b / c^2 \sum_{b=1}^{c^2} \mathcal{N}(\mathbf{x}_n; \bar{\mathbf{x}}_b; \mathbf{\Omega}_b), \quad (3.2.66)$$

où λ_b est le taux d'arrivée des nouveaux objets, $\mathbf{\Omega}_b = \text{diag}(3086, 3086, 36, 36)$ pour $b = 1, \dots, c^2$ et, pour $i, j = 1, \dots, c$,

$$\bar{\mathbf{x}}_{i+(j-1)c} = \begin{bmatrix} -500 + 1000(i - 1/2)/(c - 1) \\ 0 \\ -500 + 1000(j - 1/2)/(c - 1) \\ 0 \end{bmatrix}. \quad (3.2.67)$$

Notons que les moyennes de ce mélange partitionnent la zone d'observation $[-500, 500]^2$ en carrés de taille $c \times c$ et nous prenons $c = 6$.

Les objets sont observés par un capteur placé à l'origine, qui produit des mesures en coordonnées polaires. Nous avons alors $\mathbf{h}(\mathbf{x}) = [p_x^2 + p_y^2, \arctan(p_y/p_x)]^T$. Le bruit de mesure est caractérisé par la matrice de covariance $\mathbf{R} = \text{diag}(16, (\pi/180)^2)$. Les objets sont détectés avec une probabilité $p_{d,n} = 0.9$. Le fouillis est uniformément distribué avec une densité λ_c . Les simulations sont exécutées avec $\lambda_c = 10^{-5}$ puis $\lambda_c = 4 \times 10^{-5}$. Ainsi, pour $\lambda_c = 10^{-5}$, le nombre moyen de fausses alarmes est de 10, et pour $\lambda_c = 4 \times 10^{-5}$, nous avons une moyenne de 40 fausses alarmes. Les modèles d'évolution et de mesure vérifient (3.1.12) et (3.1.13) avec $\mathbf{x}_n^{\text{nl}} = [p_{x,n}, p_{y,n}]^T$ et $\mathbf{x}_n^{\text{l}} = [\dot{p}_{x,n}, \dot{p}_{y,n}]^T$.

Le scénario considéré est tracé sur la Figure 3.1. Dans ce scénario, le nombre d'objets varie au cours du temps et est donné par la Figure 3.1(a). Les trajectoires de chaque objet sont présentées sur la Figure

3.1(b). Les trajectoires sont tracées en rouge (clair) lorsqu'elles apparaissent et évoluent vers le bleu (foncé) au cours du temps. Ce codage permet d'identifier les objets proches à un instant donné.

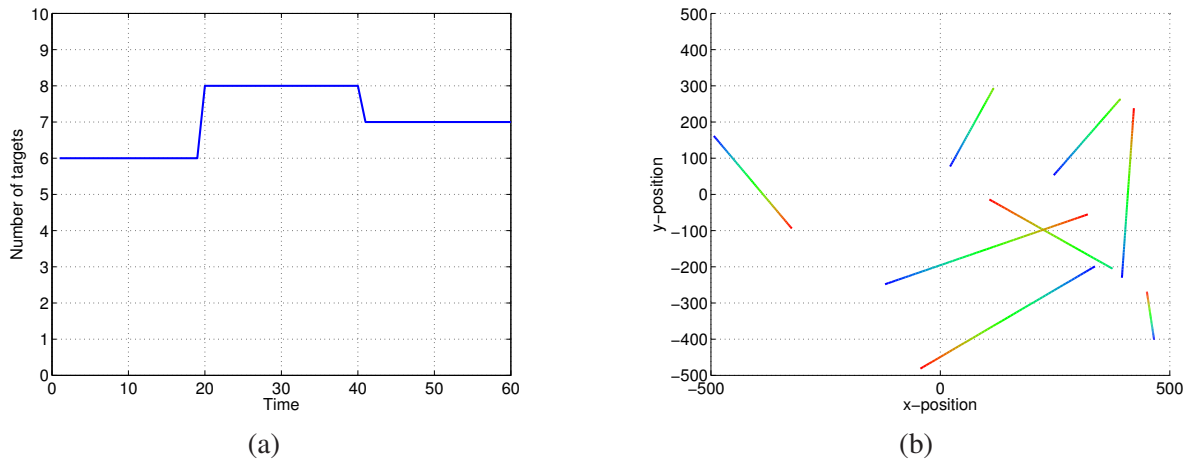


FIGURE 3.1 – Scénario pour des objets évoluant à vitesse constante : (a) Nombre d’objets au cours du temps. (b) Trajectoire des objets.

Les algorithmes sont implémentés avec un nombre de particules variant entre 100 et 10 000. Les performances sont mesurées par la distance OSPA, moyennée sur 200 réalisations.

La distance OSPA moyennée sur le temps est tracée sur la Figure 3.2, en fonction du nombre de particules. Les résultats montrent que l’implémentation marginalisée donne de meilleurs résultats en particulier lorsque la méthode est combinée à l’implémentation auxiliaire. Cependant, pour un nombre de particules donné, les implémentations auxiliaires prennent à peu près trois fois plus de temps que les implémentations bootstrap. Ce coût de calcul est néanmoins compensé par le fait que le nombre de particules des implémentations bootstrap doit être multiplié par plus de trois pour atteindre un niveau de performances similaire aux implémentations auxiliaires.

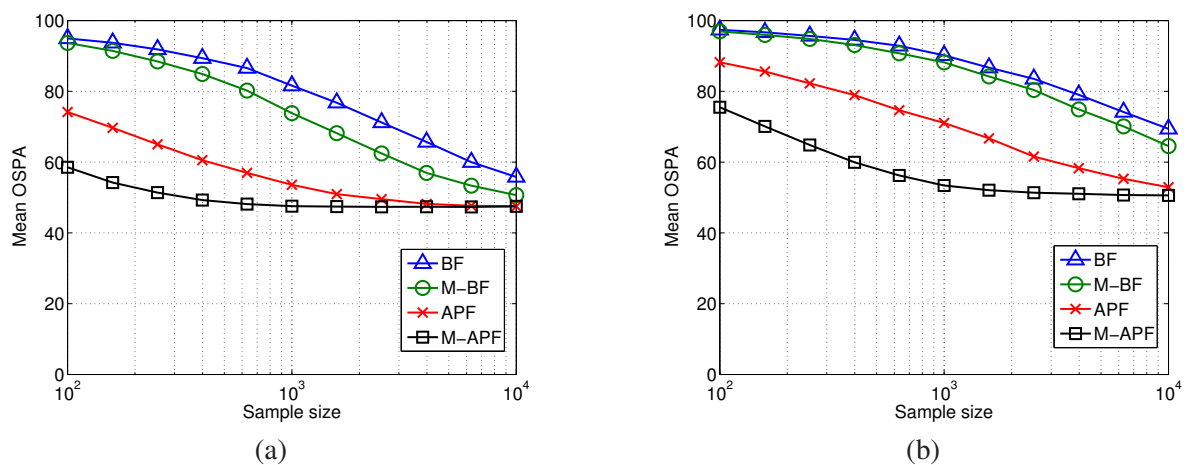


FIGURE 3.2 – Distance moyenne OSPA tracée en fonction du nombre de particules pour le scénario d’objets évoluant à vitesse constante. Les résultats sont présentés pour (a) $\lambda_c = 10^{-5}$ et (b) $\lambda_c = 4 \times 10^{-5}$.

Poursuite angle-distance d'objets manoeuvrants

Considérons à présent un scénario dans lequel les objets exécutent des virages manoeuvrants. Le vecteur d'état s'écrit maintenant $\mathbf{x}_n = [p_{x,n}, \dot{p}_{x,n}, p_{y,n}, \dot{p}_{y,n}, \omega_n]^T$ où ω_n représente la vitesse angulaire. La dynamique de l'objet est modélisée par un modèle à sauts partiellement linéaire et gaussien avec deux modes. Le premier mode ($r = 1$) correspond au modèle à vitesse constante précédent, donné par (3.2.64) avec

$$\mathbf{F} = \text{diag} \left(\mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, 0 \right), \quad (3.2.68)$$

$$\mathbf{Q} = \text{diag} \left(\sigma \mathbf{I}_2 \otimes \begin{bmatrix} T_s^3/3 & T_s^2/2 \\ T_s^2/2 & T_s \end{bmatrix}, 0 \right). \quad (3.2.69)$$

Le deuxième mode ($r = 2$) modélise le caractère manoeuvrant de l'objet pour lequel les matrices de transition et de covariance du bruit de processus s'écrivent

$$\mathbf{F}(\mathbf{x}) = \text{diag} \left(\begin{bmatrix} 1 & \sin(\omega_n T_s)/\omega_n & 0 & -[1 - \cos(\omega_n T_s)]/\omega_n \\ 0 & \cos(\omega_n T_s) & 0 & -\sin(\omega_n T_s) \\ 0 & [1 - \cos(\omega_n T_s)]/\omega_n & 1 & \sin(\omega_n T_s)/\omega_n \\ 0 & \sin(\omega_n T_s) & 0 & \cos(\omega_n T_s) \end{bmatrix}, 1 \right), \quad (3.2.70)$$

$$\mathbf{Q} = \text{diag} \left(\sigma \mathbf{I}_2 \otimes \begin{bmatrix} T_s^3/3 & T_s^2/2 \\ T_s^2/2 & T_s \end{bmatrix}, \beta \right). \quad (3.2.71)$$

Ce modèle satisfait donc (3.2.56)-(3.2.57) avec $\mathbf{x}_n^{\text{nl}} = [p_{x,n}, p_{y,n}, \omega_n]^T$ et $\mathbf{x}_n^{\text{l}} = [\dot{p}_{x,n}, \dot{p}_{y,n}]^T$. Les paramètres du modèle sont $T_s = 1$, $\sigma = 1/625$ et $\beta = (5\pi/180)^2$. La probabilité de survie est $p_{s,n} = 0.95$. Les probabilités de transitions pour la Chaîne de Markov $\{R_n\}$ sont $p(r_n = 1|r_{n-1} = 1) = 0.93$ et $p(r_n = 2|r_{n-1} = 2) = 0.8$.

L'intensité des naissances est donnée par (3.2.66), et la composante correspondant à la vitesse angulaire dans l'intensité des naissances est prise nulle. Le modèle associé aux mesures est identique au scénario précédent.

Les trajectoires sont générées en simulant le mode suivant le modèle de transition utilisé par le filtre : lorsque $r_n = 1$ le modèle à vitesse constante est utilisé alors que le modèle manoeuvrant est utilisé pour $r_n = 2$. La vitesse angulaire durant le virage est constante et est choisie au hasard suivant le mélange $[\mathcal{N}(5\pi/180, (2\pi/180)^2) + \mathcal{N}(-5\pi/180, (2\pi/180)^2)]/2$. Seules les trajectoires qui ne sortent pas de la région $[-500, 500]^2$ sont considérées. Une réalisation du scénario est présentée sur la Figure 3.3. Le nombre d'objets varie selon la Figure 3.3(a) et les trajectoires des objets sont présentées sur la Figure 3.3(b).

La distance OSPA, moyennée sur les réalisations ainsi que sur le temps, est tracée sur la Figure 3.4 en fonction du nombre de particules. Les résultats sont similaires à ceux obtenus pour le scénario précédent. L'impact de la marginalisation est donc important également pour les scénarios manoeuvrants. Comme nous nous y attendions, le nombre de particules requis pour atteindre un niveau optimal de performances est plus important pour les scénarios manoeuvrants. Cela est dû au fait que la dimension des particules est maintenant plus grande. Ainsi, pour $\lambda = 10^{-5}$, environ 4000 particules sont nécessaires pour que le M-APF réussisse à atteindre le niveau de performances optimal alors que seulement 1500 particules étaient requises pour le scénario précédent. De plus, la variabilité du nombre d'objets est plus importante pour ce second scénario, comme cela peut être observé à partir des Figures 3.1(a) et 3.3(a).

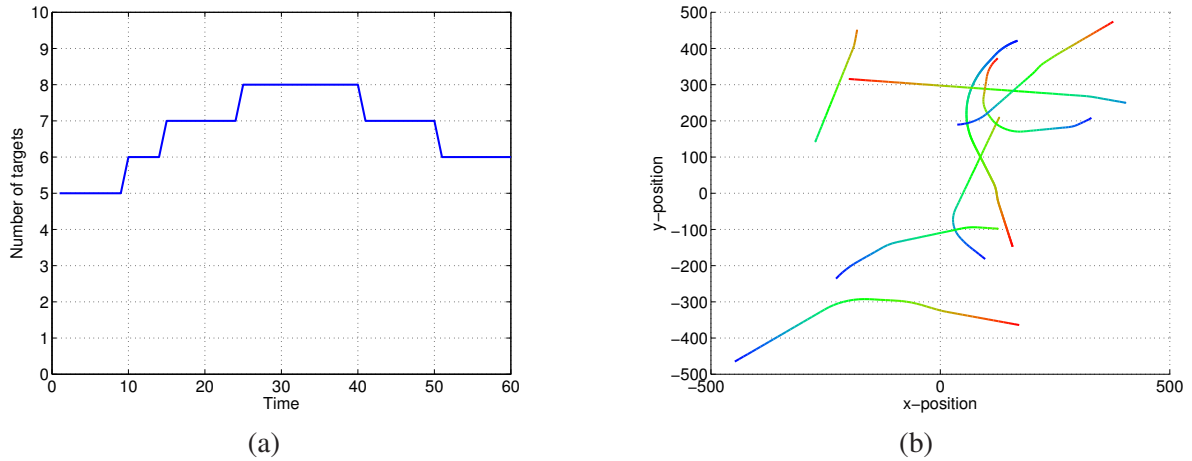


FIGURE 3.3 – Réalisation d’un scénario d’objets manoeuvrants : (a) Nombre d’objets au cours du temps. (b) Trajectoire des objets.

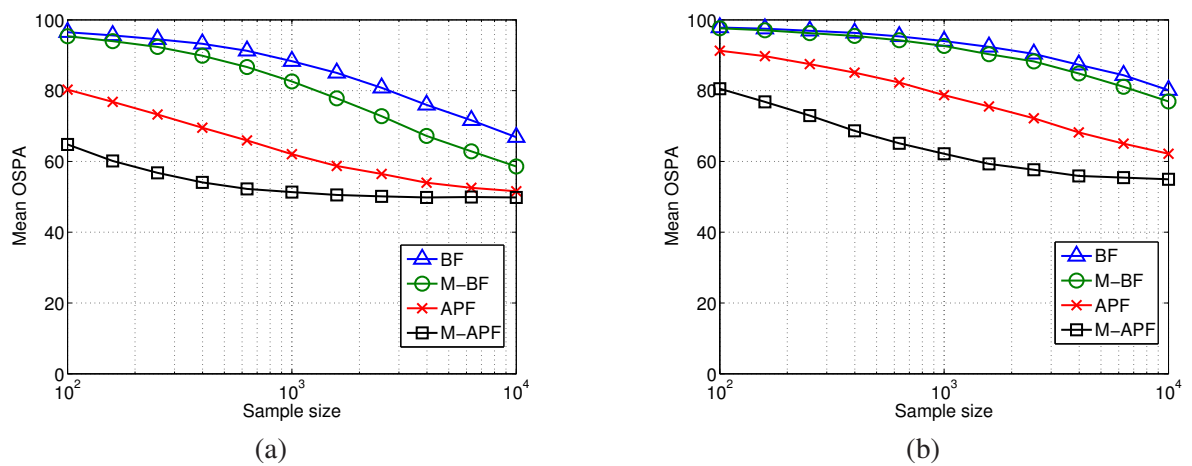


FIGURE 3.4 – Distance OSPA moyennée sur les réalisations et sur le temps pour la poursuite d’objets manoeuvrants. Les résultats sont présentés pour (a) $\lambda_c = 10^{-5}$ et (b) $\lambda_c = 4 \times 10^{-5}$.

3.3 Méthodes Monte Carlo conditionnelles temporelles

Dans cette section, nous cherchons toujours à appliquer les résultats présentés à la Section 3.1. Cependant nous montrons qu'une partition spatiale de $\mathbf{X}_{0:n}$ n'est pas la seule façon d'obtenir une réduction de variance des estimateurs d'intérêt. En effet, nous pouvons réécrire l'estimateur $\Theta_n = \int f(\mathbf{x}_{0:n})p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})d\mathbf{x}_{0:n}$ comme

$$\Theta_n = \int \left[\int f(\mathbf{x}_{0:n})p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)d\mathbf{x}_n \right] p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{0:n-1}. \quad (3.3.1)$$

Par construction même, les méthodes MC séquentielles propagent une approximation MC de la loi $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$; $\widehat{p}(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$ est donc disponible et cette approximation implique le vecteur aléatoire $\mathbf{X}_{0:n-1}$. On peut donc naturellement se demander comment exploiter cette approximation pour obtenir un estimateur de Θ_n , sans utiliser une approximation de $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$. Une telle méthodologie suggère d'appliquer les résultats généraux (3.1.1)-(3.1.2) à une partition temporelle $(X_1, X_2) = (\mathbf{X}_{0:n-1}, \mathbf{X}_n)$ de $\mathbf{X}_{0:n}$. Nous identifions alors dans l'expression de Θ_n

$$\Theta = \int f(x_1, x_2)p(x_1, x_2)dx_1dx_2 \quad (3.3.2)$$

$$= \int \left[\int f(x_1, x_2)p(x_2|x_1)dx_2 \right] p(x_1)dx_1. \quad (3.3.3)$$

Partons de l'approximation

$$p(x_1) \approx \widehat{p}(x_1) = \sum_{i=1}^N w^i(\mathbf{x}_1^{1:N})\delta_{x_1^i}. \quad (3.3.4)$$

En tirant, pour tout i , $1 \leq i \leq N$, $x_2^i \sim p(x_2|x_1^i)$, nous obtenons l'approximation suivante de $p(x_1, x_2)$

$$\widehat{p}(x_1, x_2) = \sum_{i=1}^N w^i(\mathbf{x}_1^{1:N})\delta_{(x_1^i, x_2^i)}. \quad (3.3.5)$$

Remarque 3.2. Contrairement au cas spatial, notons que chaque poids w^i peut dépendre de $\{x_1^i\}_{i=1}^N$, mais pas de $\{x_2^i\}_{i=1}^N$. En effet, dans le cas spatial, nous avons l'identification $X_1 = \mathbf{X}_{0:n}^{\text{nl}}$ et $X_2 = \mathbf{X}_{0:n}^1$. Par conséquent, afin d'obtenir une approximation MC de $p(x_1, x_2)$ qui ne dépendrait pas de $\{x_2^i\}_{i=1}^N$, ces particules devraient être tirées suivant $p(x_2|x_1) = p(\mathbf{x}_{0:n}^1|\mathbf{x}_{0:n}^{\text{nl}}, \mathbf{y}_{0:n})$. En d'autres termes, cela impliquerait de retirer à chaque instant les nouvelles trajectoires $\{\mathbf{x}_{0:n}^{1,i}\}_{i=1}^N$, plutôt que d'étendre simplement les trajectoires existantes, comme cela est fait dans les FP classiques, pour des raisons de coût de calcul.

Nous pouvons donc mettre en évidence deux options pour calculer un estimateur de Θ : approcher l'espérance dans (3.3.2) en utilisant l'approximation (3.3.5) ; ou n'utiliser que l'approximation (3.3.4) puis approcher (3.3.3) par le calcul de l'espérance conditionnelle $\int f(x_1, x_2)p(x_2|x_1)dx_2$. Notons alors les deux estimateurs

$$\widehat{\Theta}(\mathbf{x}_1^{1:N}, \mathbf{x}_2^{1:N}) = \sum_{i=1}^N w^i(\mathbf{x}_1^{1:N})f(x_1^i, x_2^i), \quad (3.3.6)$$

$$\widetilde{\Theta}(\mathbf{x}_1^{1:N}) = \sum_{i=1}^N w^i(\mathbf{x}_1^{1:N}) \left[\int f(x_1^i, x_2)p(x_2|x_1^i)dx_2 \right]. \quad (3.3.7)$$

Dans cette section, nous appellerons $\widehat{\Theta}(\mathbf{x}_1^{1:N}, \mathbf{x}_2^{1:N})$ (resp. $\widetilde{\Theta}(\mathbf{x}_1^{1:N})$) l'estimateur purement MC (resp. CMC) de Θ .

Comparons maintenant $\tilde{\Theta}$ à $\hat{\Theta}$. Comme dans la Section 3.1, $\tilde{\Theta}$ est préférable à $\hat{\Theta}$, pour tout N , mais pour des raisons différentes. En effet, nous avons

$$E(w^i(\mathbf{X}_1^{1:N})f(X_1^i, X_2^i)|\mathbf{x}_1^{1:N}) = w^i(\mathbf{x}_1^{1:N}) \int f(x_1^i, x_2)p(x_2|x_1^i)dx_2. \quad (3.3.8)$$

Par conséquent, d'après (3.1.2), la variance de chaque terme (3.3.7) est inférieure ou égale à celle du terme correspondant dans (3.3.6); cependant, ce n'est pas suffisant pour conclure que $\text{var}(\tilde{\Theta}) \leq \text{var}(\hat{\Theta})$ puisque les termes peuvent être ici dépendants entre eux. Heureusement, (3.3.8) implique que $\tilde{\Theta} = E(\hat{\Theta}|\mathbf{x}_1^{1:N})$, donc que $\tilde{\Theta}$ est préférable à $\hat{\Theta}$ d'après (3.1.1) et (3.1.2), quelle que soit l'approximation MC $\hat{p}(x_1)$ dans (3.3.4).

Considérons à présent le calcul pratique de l'estimateur $\tilde{\Theta}$, dans le cas temporel. Observons tout d'abord que contrairement aux FP marginalisés qui reposent sur une partition spatiale, le vecteur d'état n'a plus besoin d'être multi-dimensionnel lorsque l'on considère une décomposition temporelle. Dans le reste de cette section, nous montrerons que $\tilde{\Theta}$ peut être calculé dans certains modèles HMC et JMSS, dans lesquels une approximation de $p(x_1)$ est disponible et dans lesquels le calcul d'un moment suivant $p(x_2|x_1)$ est possible. Lorsque ces deux conditions ne sont pas réunies, l'estimateur CMC reste cependant d'intérêt pratique; en effet, nous proposerons deux classes d'approximations visant à approcher $\hat{\Theta}$. La première classe est basée sur des approximations numériques de type EKF/UKF; l'avantage d'utiliser une partition temporelle est que ces approximations ne sont pas propagées dans le temps; elles ne sont utilisées que localement. Dans l'esprit des récentes contributions [117] [65], nous proposons également une implémentation de $\tilde{\Theta}$ reposant sur une méthode MC locale, qui permet d'approcher simultanément $p(x_1)$ et $\int f(x_1, x_2)p(x_2|x_1)dx_1$.

Nous devons enfin prendre en compte la complexité du calcul de $\tilde{\Theta}$. Comme nous le verrons, dans certains cas, $\tilde{\Theta}$ peut être calculé sous les mêmes hypothèses et le même coût de calcul que $\hat{\Theta}$ (voir le Paragraphe 3.3.1). Notons que si l'on souhaite partitionner l'ensemble de variables $(\mathbf{X}_1, \mathbf{X}_2)$, il est préférable que \mathbf{X}_1 soit le plus petit possible. Plus précisément, soit $\Theta = E(f(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3))$ et soit une approximation $\hat{p}(\mathbf{x}_1, \mathbf{x}_2)$ donnée. Alors nous pouvons déduire deux estimateurs bayésiens CMC de Θ : $\tilde{\Theta}^{\mathbf{X}_3}$ construit à partir de $\hat{p}(\mathbf{x}_1, \mathbf{x}_2)$ et de $\Theta = E[E(f(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)|\mathbf{X}_1, \mathbf{X}_2)]$, dans laquelle l'espérance prise par rapport à \mathbf{X}_3 est calculée exactement; et $\tilde{\Theta}^{(\mathbf{X}_2, \mathbf{X}_3)}$ déduit de $\hat{p}(\mathbf{x}_1)$ et de $\Theta = E[E(f(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)|\mathbf{X}_1)]$. L'estimateur $\tilde{\Theta}^{(\mathbf{X}_2, \mathbf{X}_3)}$ est préférable à $\tilde{\Theta}^{\mathbf{X}_3}$ mais requiert le calcul supplémentaire d'une espérance puisque $E(f(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)|\mathbf{X}_1) = E[E(f(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)|\mathbf{X}_1, \mathbf{X}_2)]$. Dans certains modèles à sauts, les deux estimateurs peuvent être calculés, mais le calcul de $\tilde{\Theta}^{(\mathbf{X}_2, \mathbf{X}_3)}$ requiert un coût de calcul supplémentaire (voir Paragraphe 3.3.3).

3.3.1 Estimateurs CMC pour les modèles HMC

Nous nous plaçons dans un premier temps dans les modèles HMC,

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\mathbf{x}_0) \prod_{i=1}^n f_i|\mathbf{x}_{i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}) \prod_{i=1}^n g_i(\mathbf{y}_i|\mathbf{x}_i) \quad (3.3.9)$$

et nous cherchons à estimer Θ_n dans (3.3.9) à partir d'une technique de marginalisation temporelle.

Expression de l'estimateur CMC $\tilde{\Theta}_n$

Dans le modèle (3.3.9), Θ_n peut se réécrire

$$\Theta_n = \int f(\mathbf{x}_n)p(\mathbf{x}_{0:n-1}, \mathbf{x}_n|\mathbf{y}_{0:n})d\mathbf{x}_{0:n-1}d\mathbf{x}_n. \quad (3.3.10)$$

Par conséquent, (3.3.10) coïncide avec (3.3.2), avec $X_1 = \mathbf{X}_{0:n-1}$, $X_2 = \mathbf{X}_n$, $f(x_1, x_2)$ dépend seulement de x_2 et $p(x_1, x_2)$ est la loi jointe a posteriori

$$p(\mathbf{x}_{0:n-1}, \mathbf{x}_n | \mathbf{y}_{0:n}) = \underbrace{p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n})}_{p(x_1)} \underbrace{p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)}_{p(x_2|x_1)}. \quad (3.3.11)$$

Soit $\hat{p}(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ une approximation MC de $p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n})$ et $\mathbf{x}_n^i \sim p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n)$. D'après (3.3.6) et (3.3.7), les estimateurs purement MC et CMC de Θ_n dans (3.3.10) s'écrivent respectivement

$$\hat{\Theta}_n(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{x}_n^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i(\mathbf{x}_{0:n-1}^{1:N}) f(\mathbf{x}_n^i), \quad (3.3.12)$$

$$\tilde{\Theta}_{n-1}(\mathbf{x}_{0:n-1}^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i(\mathbf{x}_{0:n-1}^{1:N}) \int f(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n. \quad (3.3.13)$$

D'après (3.3.13), nous avons besoin d'une approximation de $p(x_1) = p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n})$. Comme nous l'avons montré au Chapitre 2, $p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n})$ est relié à $p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1})$ par

$$p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n | \mathbf{x}_{n-1}) p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1})}{\int p(\mathbf{y}_n | \mathbf{x}_{n-1}) p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1}) d\mathbf{x}_{0:n-1}}. \quad (3.3.14)$$

L'équation (3.3.14) suggère donc de calculer l'estimateur (3.3.13) de la façon suivante :

1. Propager une approximation $\hat{p}(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ de $p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1})$;
2. Dédire une approximation $\hat{p}(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ de la loi de lissage à un pas en mettant à jour les poids

$$\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_n | \mathbf{x}_{n-1}^i), \quad \sum_{i=1}^N \tilde{w}_{n-1}^i = 1; \quad (3.3.15)$$

3. Calculer les intégrales dans (3.3.13) et déduire $\tilde{\Theta}_n$.

Commentons maintenant l'implémentation pratique de ces trois étapes.

Tout d'abord, la propagation d'une approximation de $p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1})$ est un problème étudié au Chapitre 2 dont la solution inclut, par exemple, les algorithmes SIS/SIR et APF [97] [41].

Observons cependant que la loi $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$ impliquée dans le calcul de (3.3.13) coïncide avec la loi optimale des algorithmes SIS/SIR et FA (voir Algorithme 2.10). D'un point de vue pratique, il peut donc être intéressant de propager une approximation de $p(\mathbf{x}_{0:n} | \mathbf{y}_{0:n})$ en utilisant un de ces algorithmes, que nous avons par ailleurs comparés à la Section 2.4, puisque tous sont basés sur la loi $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$ ainsi que sur l'expression de $p(\mathbf{y}_n | \mathbf{x}_{n-1})$. Néanmoins, d'autres algorithmes de filtrage peuvent être utilisés et éventuellement donner de meilleurs résultats que ceux qui s'appuient sur $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$ et $p(\mathbf{y}_n | \mathbf{x}_{n-1})$ [41] [63] [7] (voir Remarque 2.1). Bien sûr, le choix de l'approximation $\hat{p}(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1})$ a un impact sur la qualité de l'estimateur, même si nous verrons qu'il existe des situations (voir Paragraphe 3.3.4) dans lesquelles une mauvaise approximation de $p(x_1)$ peut néanmoins produire un bon estimateur $\tilde{\Theta}_n$. Rappelons cependant que quelle que soit l'approximation $\hat{p}(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1})$, $\tilde{\Theta}_n$ a toujours une variance inférieure à celle de $\hat{\Theta}_n$.

Nous nous intéressons maintenant au calcul exact puis, lorsque cela est nécessaire, au calcul approché de $\tilde{\Theta}_n$. Dans ce dernier cas, nous distinguons deux approximations. Des approximations numériques pour des modèles dans lesquelles les non linéarités ne sont pas sévères, et des approximations basées sur une méthode MC locale pour les modèles plus généraux.

Implémentation exacte de $\tilde{\Theta}_n$

D'après (3.3.8), nous savons que la variance de $\tilde{\Theta}_n$ est inférieure à celle de $\hat{\Theta}_n$; mais $\tilde{\Theta}_n$ ne peut être utilisé que si \tilde{w}_n^i et l'intégrale $\int f(\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)d\mathbf{x}_n$ sont calculables. Ces conditions sont vérifiées pour certains modèles HMC et pour certaines fonctions $f(\cdot)$. Considérons par exemple les modèles semi-linéaires avec bruit additif gaussien, donnés par les équations

$$\mathbf{X}_n = \mathbf{f}_n(\mathbf{X}_{n-1}) + \mathbf{K}_n(\mathbf{X}_{n-1}) \times \mathbf{U}_n, \quad (3.3.16)$$

$$\mathbf{Y}_n = \mathbf{H}_n \mathbf{X}_n + \mathbf{V}_n, \quad (3.3.17)$$

dans lesquels $\{\mathbf{U}_n\}$ et $\{\mathbf{V}_n\}$ sont i.i.d., mutuellement indépendants et indépendants de \mathbf{X}_0 , $\mathbf{U}_n \sim \mathcal{N}(\cdot; 0; \mathbf{I})$ et $\mathbf{V}_n \sim \mathcal{N}(\cdot; 0; \mathbf{R}_n^v)$. Comme nous l'avons vu dans le Paragraphe 2.4.6, le modèle ARCH appartient à cette classe de modèles avec $f_n(x_{n-1}) = 0$, $k_n(x_{n-1}) = \sqrt{\beta_0 + \beta_1 x_{n-1}^2}$ et $H_n = 1$. Dans le modèle (3.3.16)-(3.3.17), $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ et $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ sont gaussiennes. Plus précisément, si $\mathbf{Q}_n(\mathbf{x}_{n-1}) = \mathbf{K}_n(\mathbf{x}_{n-1})\mathbf{K}_n(\mathbf{x}_{n-1})^T$, alors

$$\mathbf{L}_n(\mathbf{x}_{n-1}) = \mathbf{H}_n \mathbf{Q}_n(\mathbf{x}_{n-1}) \mathbf{H}_n^T + \mathbf{R}_n^v, \quad (3.3.18)$$

$$\mathbf{m}_n(\mathbf{x}_{n-1}, \mathbf{y}_n) = \mathbf{f}_n(\mathbf{x}_{n-1}) + \mathbf{Q}_n(\mathbf{x}_{n-1}) \mathbf{H}_n^T \mathbf{L}_n^{-1}(\mathbf{x}_{n-1}) (\mathbf{y}_n - \mathbf{H}_n \mathbf{f}_n(\mathbf{x}_{n-1})), \quad (3.3.19)$$

$$\mathbf{P}_n(\mathbf{x}_{n-1}) = \mathbf{Q}_n(\mathbf{x}_{n-1}) - \mathbf{Q}_n(\mathbf{x}_{n-1}) \mathbf{H}_n^T \mathbf{L}_n^{-1}(\mathbf{x}_{n-1}) \mathbf{H}_n \mathbf{Q}_n(\mathbf{x}_{n-1}), \quad (3.3.20)$$

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n) = \mathcal{N}(\mathbf{x}_n, \mathbf{m}_n(\mathbf{x}_{n-1}, \mathbf{y}_n), \mathbf{P}_n(\mathbf{x}_{n-1})), \quad (3.3.21)$$

$$p(\mathbf{y}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{y}_n, \mathbf{H}_n \mathbf{f}_n(\mathbf{x}_{n-1}), \mathbf{L}_n(\mathbf{x}_{n-1})). \quad (3.3.22)$$

Enfin, dans ces modèles, le calcul de l'estimateur CMC est faisable pour certaines fonctions $f(\cdot)$. Si $f(\mathbf{x})$ est un polynôme en \mathbf{x} , le problème se réduit au calcul des premiers moments de la gaussienne (3.3.21). Dans le cas particulier où $f(\mathbf{x}) = \mathbf{x}$ (que nous utilisons pour donner un estimateur de l'état caché) aucun calcul supplémentaire n'est nécessaire, car l'intégrale dans (3.3.13) est égale à $\mathbf{m}_n(\mathbf{x}_{n-1}^i, \mathbf{y}_n)$.

Remarque 3.3. Dans cette classe de modèles, les calculs de $\tilde{\Theta}_n$ et de $\hat{\Theta}_n$ requièrent le même coût de calcul si $f(\mathbf{x}) = \mathbf{x}$. En effet, les deux estimateurs s'appuient sur le calcul des paramètres $\mathbf{m}_n(\mathbf{x}_{n-1}^i, \mathbf{y}_n)$ et $\mathbf{P}_n(\mathbf{x}_{n-1}^i)$ de $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$, et utilisent ces lois pour tirer de nouveaux points \mathbf{x}_n^i , qui dans les deux cas sont nécessaires pour l'itération suivante. La seule différence provient du fait que $\hat{\Theta}_n = \sum_{i=1}^N \tilde{w}_{n-1}^i \mathbf{x}_n^i$, alors que $\tilde{\Theta}_n = \sum_{i=1}^N \tilde{w}_{n-1}^i \mathbf{m}_n(\mathbf{x}_{n-1}^i, \mathbf{y}_n)$.

Approximation numérique de $\tilde{\Theta}_n$

Discutons maintenant des cas où $\tilde{\Theta}_n$ ne peut pas être implémenté de façon exacte parce que $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ et/ou les moments de $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ ne sont pas calculables. Nous proposons dans un premier temps deux approximations numériques. La première approche numériquement $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ ainsi que les moments de $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$; la seconde contourne le problème de l'approximation de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$.

Ainsi que nous l'avons rappelé au Paragraphe 3.2.5, des techniques de linéarisations [43], des développements en série de Taylor [110] ou la transformation UT [66] peuvent être utilisées pour approcher $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ ainsi qu'un moment de $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$; nous pouvons donc utiliser ces approximations dans l'expression (3.3.13). L'algorithme résultant peut alors être vu comme une alternative aux EKF/UKF dans lesquels $p_{n|n}$ est approché par une gaussienne; ou aux FP classiques dans lesquels nous recherchons une approximation discrète de $p_{n|n}$. Ici, nous partons d'une approximation discrète de $p_{n-1|n-1}$ produite par une

méthode SMC, et de façon similaire à l'EKF/UKF, nous cherchons une approximation numérique de $\tilde{\Theta}_n$, conditionnellement à $\hat{p}_{n-1|n-1}$.

Cependant, l'approximation de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ peut être mauvaise puisque l'approximation de la fonction est déduite d'une approximation de ses moments. Afin d'éviter l'utilisation d'une approximation de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$, rappelons, par exemple, que l'algorithme SIR (cf Algorithme 2.1) fournit une approximation de $p_{0:n-1|n}$ donnée par $\hat{p}_{0:n-1|n} = \sum \tilde{w}_n^i \delta_{\mathbf{x}_{0:n-1}^i}$, où les poids \tilde{w}_n^i sont maintenant proportionnels à $w_{n-1}^i f_{n|n-1}(\mathbf{x}_n^i|\mathbf{x}_{n-1}^i) g_n(\mathbf{y}_n|\mathbf{x}_n^i) / q(\mathbf{x}_n^i|\mathbf{x}_{n-1}^i)$, et dépendent donc du jeu de points $\{\mathbf{x}_n^i\}_{i=1}^N$. Dans ce cas, les estimateurs MC et CMC deviennent

$$\hat{\Theta}_n(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{x}_n^{1:N}) = \sum_{i=1}^N \tilde{w}_n^i(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{x}_n^{1:N}) f(\mathbf{x}_n^i), \quad (3.3.23)$$

$$\tilde{\Theta}_n(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{x}_n^{1:N}) = \sum_{i=1}^N \tilde{w}_n^i(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{x}_n^{1:N}) \int f(\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n, \quad (3.3.24)$$

mais ne peuvent plus être comparés facilement (c'est le cas dans l'introduction de cette section parce que les poids w^i dans (3.3.6) et (3.3.7) dépendent seulement de $\{\mathbf{x}_1^i\}_{i=1}^N$). D'un autre côté, le calcul de (3.3.24) ne requiert pas celui de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$, mais seulement celui de $\int f(\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n$, ce qui peut être d'intérêt pratique dans les modèles où les moments de $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ peuvent être approchés aisément. D'autre part, (3.3.24) suggère de calculer $\int f(\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n$ en utilisant une méthode MC locale. Cependant, une approximation de $p_{0:n-1|n-1}$ repose déjà sur l'ensemble additionnel $\{\mathbf{x}_n^i\}_{i=1}^N$; par conséquent, une méthode MC supplémentaire n'assurerait pas nécessairement une amélioration des performances de l'estimateur classique (3.3.23). Quitte à utiliser une méthode MC locale, nous allons montrer qu'elle peut être utilisée pour améliorer l'approximation de $p_{0:n-1|n}$ et de $\int f(\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n$.

Approximation de $\tilde{\Theta}_n$ par méthodes MC locales

Il se peut que l'approche précédente ne soit pas satisfaisante car les particules $\{\mathbf{x}_{0:n-1}^i\}_{i=1}^N$ participant à l'approximation de $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n})$ ne sont pas tirées en prenant en compte l'observation \mathbf{y}_n , d'une part, et les approximations précédentes peuvent donner de mauvais résultats dans les modèles où les non linéarités sont trop sévères, d'autre part.

Nous pouvons palier le premier problème en cherchant à propager une approximation de $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n})$ qui n'est pas déduite d'une approximation de $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$. Par exemple, l'Algorithme 2-S du Chapitre 2 (cf. Algorithme 2.6) prend en compte l'observation \mathbf{y}_n pour tirer les particules $\{\mathbf{x}_{n-1}^i\}_{i=1}^N$. De façon plus générale, nous pouvons utiliser l'échantillonnage d'importance pour propager

$$p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n}) \propto p(\mathbf{x}_{0:n-2}|\mathbf{y}_{0:n-1}) \frac{f_{n-1|n-2}(\mathbf{x}_{n-1}|\mathbf{x}_{n-2}) g_{n-1}(\mathbf{y}_{n-1}|\mathbf{x}_{n-1})}{p(\mathbf{y}_{n-1}|\mathbf{x}_{n-2})} p(\mathbf{y}_n|\mathbf{x}_{n-1}). \quad (3.3.25)$$

Partant d'une approximation $\{\tilde{w}_{n-2}^i, \mathbf{x}_{0:n-2}^i\}_{i=1}^N$ de $p(\mathbf{x}_{0:n-2}|\mathbf{y}_{0:n-1})$ ainsi que d'une loi d'importance conditionnelle $q(\mathbf{x}_{n-1}|\mathbf{x}_{0:n-2}^i)$, les poids associés à $\mathbf{x}_{n-1}^i \sim q(\mathbf{x}_{n-1}|\mathbf{x}_{0:n-2}^i)$ s'écrivent

$$\tilde{w}_{n-1}^i \propto \tilde{w}_{n-2}^i \frac{f_{n-1|n-2}(\mathbf{x}_{n-1}^i|\mathbf{x}_{n-2}^i) g_{n-1}(\mathbf{y}_{n-1}|\mathbf{x}_{n-1}^i) p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)}{q(\mathbf{x}_{n-1}^i|\mathbf{x}_{0:n-2}^i) p(\mathbf{y}_{n-1}|\mathbf{x}_{n-2}^i)}, \quad \sum_{i=1}^N \tilde{w}_{n-1}^i = 1 \quad (3.3.26)$$

L'expression de l'estimateur CMC est toujours donnée par (3.3.13), mais les poids \tilde{w}_{n-1}^i sont données par (3.3.26). Comme dans l'Algorithme 2.6, nous retrouvons la loi d'importance conditionnelle optimale,

$q^{\text{opt}}(\mathbf{x}_{n-1}|\mathbf{x}_{0:n-2}) \propto p(\mathbf{x}_{n-1}|\mathbf{x}_{n-2}, \mathbf{y}_n)p(\mathbf{y}_n|\mathbf{x}_{n-1})$; dans ce cas, les poids $\tilde{w}(\mathbf{x}_{0:n-1}^{1:N})$ intervenant dans (3.3.13) ne dépendent plus de $\mathbf{x}_{n-1}^{1:N}$. Cette loi peut être approchée par les techniques de linéarisations rappelées ci-dessus. En pratique, le calcul de (3.3.26) implique également celui de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$. Comme nous le voyons à présent, les techniques d'approximation basées sur des méthodes MC locales [65] [117] peuvent être adaptées ici.

Dans le contexte de la marginalisation spatiale, les poids associés à l'approximation de $p(x_1)$ ainsi que l'intégrale $\int f(x_1, x_2)p(x_2|x_1)dx_2$ ne sont pas calculables. Une méthode MC locale permet alors d'approcher à la fois l'intégrale ainsi que les poids nécessaires au calcul de $\hat{p}(x_1)$. Plus précisément, l'argument suivant est exploité dans [65]. Si $p(x_1)p(x_2|x_1)=p(x_1, x_2)$ est connu, alors pour x_1^i donné, une méthode de simulation basée sur l'échantillonnage d'importance produit M points $\{x_2^{i,j}\}_{j=1}^M$; ces M points permettent d'approcher un moment suivant $p(x_2|x_1^i)$ et produisent également un estimateur non biaisé de $p(x_1^i)$. Ainsi, les poids $\{w_1^i\}_{i=1}^N$ associés à l'approximation de $p(x_1)$ (et qui reposent sur l'expression de $\{p(x_1^i)\}_{i=1}^N$) peuvent être approchés par des poids $\{\tilde{w}_1^i\}_{i=1}^N$ dont l'espérance par rapport à $\{X_2^{i,j}\}_{j=1}^M$ est $\{w_1^i\}_{i=1}^N$. Dans le contexte du filtrage séquentiel utilisant une partition spatiale, l'expression $p(x_1, x_2)$ n'est connue qu'à une constante près; cependant, l'application séquentielle de la technique décrite ci-dessus garantit que la loi marginale ciblée par les poids approximés est encore $p(x_1)$. Cet argument est donc adapté au cas où nous souhaitons calculer les poids dans (3.3.26) ainsi que les moments d'intérêt dans (3.3.13).

Pour une trajectoire donnée $\mathbf{x}_{0:n-1}^i$, nous tirons M particules $\mathbf{x}_n^{i,j} \sim q^{\text{CMC}}(\mathbf{x}_n|\mathbf{x}_{0:n-1}^i)$ (l'exposant CMC indique que cette loi d'importance est dédiée à l'étape CMC); nous obtenons ainsi les approximations des intégrales dans (3.3.13) et des expressions de $p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ données par

$$\int f(\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)d\mathbf{x}_n \approx \sum_{j=1}^M w_n^{\text{CMC},i,j} f(\mathbf{x}_n^{i,j}), \quad (3.3.27)$$

$$w_n^{\text{CMC},i,j} \propto \frac{f_{n|n-1}(\mathbf{x}_n^{i,j}|\mathbf{x}_{n-1}^i)g_n(\mathbf{y}_n|\mathbf{x}_n^{i,j})}{q^{\text{CMC}}(\mathbf{x}_n^{i,j}|\mathbf{x}_{0:n-1}^i)}, \quad \sum_{j=1}^M w_n^{\text{CMC},i,j} = 1, \quad (3.3.28)$$

$$\hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}^i) = \frac{1}{M} \sum_{j=1}^M \frac{f_{n|n-1}(\mathbf{x}_n^{i,j}|\mathbf{x}_{n-1}^i)g_n(\mathbf{y}_n|\mathbf{x}_n^{i,j})}{q^{\text{CMC}}(\mathbf{x}_n^{i,j}|\mathbf{x}_{0:n-1}^i)}. \quad (3.3.29)$$

Enfin, l'expression approchée des poids (3.3.26) s'écrit

$$\hat{w}_{n-1}^i \propto \tilde{w}_{n-2}^i \frac{f_{n-1|n-2}(\mathbf{x}_{n-1}^i|\mathbf{x}_{n-2}^i)g_{n-1}(\mathbf{y}_{n-1}|\mathbf{x}_{n-1}^i)}{q(\mathbf{x}_{n-1}^i|\mathbf{x}_{0:n-2}^i)\hat{p}(\mathbf{y}_{n-1}|\mathbf{x}_{n-2}^i)} \hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}^i), \quad \sum_{i=1}^N \hat{w}_{n-1}^i = 1. \quad (3.3.30)$$

et celle de l'estimateur CMC approché

$$\tilde{\Theta}_n \approx \sum_{i=1}^N \hat{w}_{n-1}^i \left[\sum_{j=1}^M w_n^{\text{CMC},i,j} f(\mathbf{x}_n^{i,j}) \right]. \quad (3.3.31)$$

Remarque 3.4. Posons $M = 1$ et $q^{\text{CMC}}(\mathbf{x}_n|\mathbf{x}_{0:n-1}) = q(\mathbf{x}_n|\mathbf{x}_{0:n-1})$; nous notons ainsi $\mathbf{x}_n^{i,1} = \mathbf{x}_n^i$. $\hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ dans (3.3.29) devient $f_{n|n-1}(\mathbf{x}_n^i|\mathbf{x}_{n-1}^i)g_n(\mathbf{y}_n|\mathbf{x}_n^i)/q(\mathbf{x}_n^i|\mathbf{x}_{0:n-1}^i)$, $w_n^{\text{CMC},i,1} = 1$, $\hat{w}_{n-1}^i \propto \tilde{w}_{n-2}^i f_{n|n-1}(\mathbf{x}_n^i|\mathbf{x}_{n-1}^i)g_n(\mathbf{y}_n|\mathbf{x}_n^i)/q(\mathbf{x}_n^i|\mathbf{x}_{0:n-1}^i)$, et donc $\tilde{\Theta}_n$ dans (3.3.31) s'écrit $\sum_{i=1}^N \hat{w}_{n-1}^i f(\mathbf{x}_n^i)$, qui n'est autre que l'estimateur SIS dans lequel $q(\mathbf{x}_n|\mathbf{x}_{0:n-1})$ est utilisée comme loi d'importance.

La remarque suivante montre que les poids $\hat{w}(\mathbf{x}_{0:n-1}, \mathbf{x}_n^{1:M})$ ciblent marginalement la loi $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n})$, malgré l'approximation MC utilisée (nous nous focalisons ici sur une trajectoire $\mathbf{x}_{0:n-1}^i$ donnée; par clarté,

l'exposant i est supprimé d'où la notation $\mathbf{x}_n^{1:M}$ pour désigner le jeu de M particules associée à la i -ème trajectoire).

Remarque 3.5. Nous supposons que l'étape de rééchantillonnage n'a pas été effectuée dans l'algorithme de lissage à un pas. A l'instant $n - 1$, les poids non normalisés s'écrivent

$$\hat{w}_{n-1}^u(\mathbf{x}_{0:n-1}, \mathbf{x}_n^{1:M}) = \frac{p(\mathbf{x}_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}) g_i(\mathbf{y}_i|\mathbf{x}_i)}{q(\mathbf{x}_{0:n-1})} \hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}), \quad (3.3.32)$$

$$= \frac{p(\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1})}{q(\mathbf{x}_{0:n-1})} \hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}). \quad (3.3.33)$$

La loi cible s'écrit donc

$$\pi(\mathbf{x}_{0:n-1}, \mathbf{x}_n^{1:M}) \propto p(\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1}) \hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}) \prod_{j=1}^M q^{\text{CMC}}(\mathbf{x}_n^{j}|\mathbf{x}_{n-1}), \quad (3.3.34)$$

où

$$\hat{p}(\mathbf{y}_n|\mathbf{x}_{n-1}) = \frac{1}{M} \sum_{k=1}^M \frac{f_{n|n-1}(\mathbf{x}_n^k|\mathbf{x}_{n-1}) g_n(\mathbf{y}_n|\mathbf{x}_n^k)}{q^{\text{CMC}}(\mathbf{x}_n^k|\mathbf{x}_{n-1})} \quad (3.3.35)$$

est un estimateur non biaisé de $p(\mathbf{y}_n|\mathbf{x}_{n-1})$. La densité marginale $\pi(\mathbf{x}_{0:n-1}) = \int \pi(\mathbf{x}_{0:n-1}, \mathbf{x}_n^{1:M}) d\mathbf{x}_n^{1:M}$ s'écrit alors

$$\pi(\mathbf{x}_{0:n-1}) \propto p(\mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1}) p(\mathbf{y}_n|\mathbf{x}_{n-1}) \quad (3.3.36)$$

$$= p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n}). \quad (3.3.37)$$

Simulations

Nous discutons des performances de l'estimateur CMC (possiblement approximé) dans divers modèles. Nous considérons le modèle ARCH ainsi que le modèle de Volatilité Stochastique (*Stochastic Volatility* = SV). Pour chaque simulation, nous calculons l'EQM ; la vraie moyenne est calculée par un filtre Bootstrap avec $N = 10^5$ particules.

Le modèle ARCH est un modèle particulier du modèle (3.3.16)-(3.3.17). Nous fixons $R_n^v = 3$, $\beta_0 = 1$ and $\beta_1 = 0.1$, et nous estimons X_n (donc $f(x_n) = x_n$). D'après (3.3.21), le moment d'ordre 1 de $p(x_n|x_{n-1}, y_n)$ est calculable. Nous comparons alors six estimateurs : les estimateurs FA et Bootstrap, avec $N = 1000$ particules ; un estimateur CMC basé sur l'algorithme Bootstrap, avec $N = 1000$; deux estimateurs CMC basés sur l'algorithme FA avec $N = 1000$ et $N = 100$; un estimateur CMC approché avec $q(x_n|x_{n-1}) = q^{\text{CMC}}(x_n|x_{n-1}) = f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$, $N = 1000$ et $M = 5$. Les EQM sont tracées sur la Figure 3.5. Tous les estimateurs CMC donnent de meilleurs résultats que les estimateurs MC classiques. Notons que l'estimateur CMC qui repose sur $N = 100$ particules est lui aussi plus performant que l'estimateur FA, avec un coût de calcul moindre. Enfin, l'estimateur CMC approché, qui ne repose pas sur l'expression de $p(y_n|x_{n-1})$ et $p(x_n|x_{n-1}, y_n)$ est lui aussi plus performant que les estimateurs FA et Bootstrap, même avec une faible valeur de M .

Intéressons nous maintenant au modèle SV :

$$\begin{cases} X_n &= \Phi X_{n-1} + U_n \\ Y_n &= \beta \exp(X_n/2) \times V_n \end{cases}, \quad (3.3.38)$$

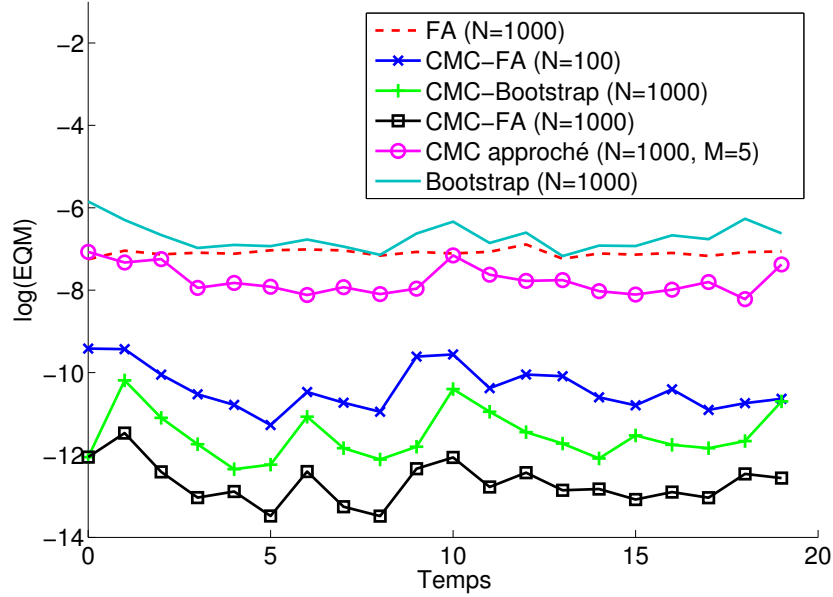


FIGURE 3.5 – EQM - Modèle ARCH - $\beta_0 = 1$, $\beta_1 = 0.1$ et $R_n^v = 3$ - $f(x_n) = x_n$. L'estimateur CMC avec $N = 100$ particules améliore les estimateurs purement MC et est proche de l'estimateur CMC avec $N = 1000$ particules. L'estimateur CMC approché améliore l'estimateur purement MC avec $M = 5$.

avec $U_n \sim \mathcal{N}(\cdot; 0; \sigma^2)$ et $V_n \sim \mathcal{N}(\cdot; 0; 1)$. Dans ce modèle, $p(y_n|x_{n-1})$ et donc $\tilde{\Theta}_n$ n'est pas calculable quelle que soit la fonction $f(\cdot)$. Nous comparons alors nos approximations de l'estimateur CMC. Notre première approximation, numérique, est basée sur l'approximation de $\int f(x_n)p(x_n|x_{n-1}y_n) dx_n$ et de $p(y_n|x_{n-1})$; la seconde ne repose que sur celle de $\int f(x_n)p(x_n|x_{n-1}, y_n)dx_n$. Ces approximations sont obtenues par un développement en série de Taylor au premier ordre de $\log(g_n(y_n|x_n))$ en Φx_{n-1} . Si l'approximation déduite de $g_n(y_n|x_n)$ est notée $\hat{g}_n(y_n|x_n)$, alors $\hat{p}(y_n|x_{n-1}) = \int f_{n|n-1}(x_n|x_{n-1})\hat{g}_n(y_n|x_n)dx_n$, avec $f_{n|n-1}(x_n|x_{n-1}) = \mathcal{N}(x_n; \Phi x_{n-1}; \sigma^2)$, est maintenant calculable. Si σ est petit, $f_{n|n-1}(x_n|x_{n-1})$ est approximativement non nulle pour des valeurs proches de Φx_{n-1} , et pour de telles valeurs $\hat{g}_n(y_n|x_n)$ est une bonne approximation de $g_n(y_n|x_n)$. Par conséquent, nous pouvons considérer que l'approximation $\hat{p}(y_n|x_{n-1})$ est correcte lorsque σ est petit. Enfin, une approximation déduite de $p(x_n|x_{n-1}, y_n)$ est donnée par une gaussienne dont les paramètres sont déduits de l'approximation précédente (voir [97]). Nous calculons également deux estimateurs CMC approchés, basés sur une méthode MC locale. Les deux estimateurs utilisent $q(x_n|x_{n-1}) = q^{\text{CMC}}(x_n|x_{n-1}) = f_{n|n-1}(x_n|x_{n-1})$. Pour le premier estimateur approché, nous utilisons $N = 750$ et $M = 10$ tandis que pour le second, nous fixons $N = 1000$ et $M = 10$.

Nous prenons d'abord $\Phi = 0.8$, $\beta = 0.6$, $\sigma = 0.18$. Les résultats sont tracés sur la Figure 3.6(a). Qu'il s'agisse des approximations numériques ou de celles basées sur une méthode MC locale, les estimateurs CMC approchés présentent de meilleures performances que l'estimateur SIS/SIR. Parmi les approximations numériques, celle qui donne de meilleurs résultats est celle qui ne repose pas sur le jeu de points supplémentaire $\tilde{\mathbf{x}}_n^{1:N}$ pour le calcul de $\hat{\Theta}_n$ (voir (3.3.24)). Enfin, les estimateurs approchés qui reposent sur une méthode MC locale avec M particules pour calculer $p_{0:n-1|n}$ et $\int f(x_n)p(x_n|x_{n-1}, y_n)dx_n$ sont les plus efficaces.

Sur la Figure 3.6(b), nous augmentons maintenant le bruit de processus en prenant $\sigma = 0.38$. Cette augmentation a des conséquences sur l'approximation de $p(y_n|x_{n-1})$; comme prévu, l'approximation numé-

rique de l'estimateur CMC qui repose sur $\hat{p}(y_n|x_{n-1})$ n'est plus fiable. La seconde approximation numérique est plus robuste et présente des performances similaires à l'estimateur classique. Enfin, les approximations basées sur la méthode MC locale ne sont pas affectées par l'augmentation de σ .

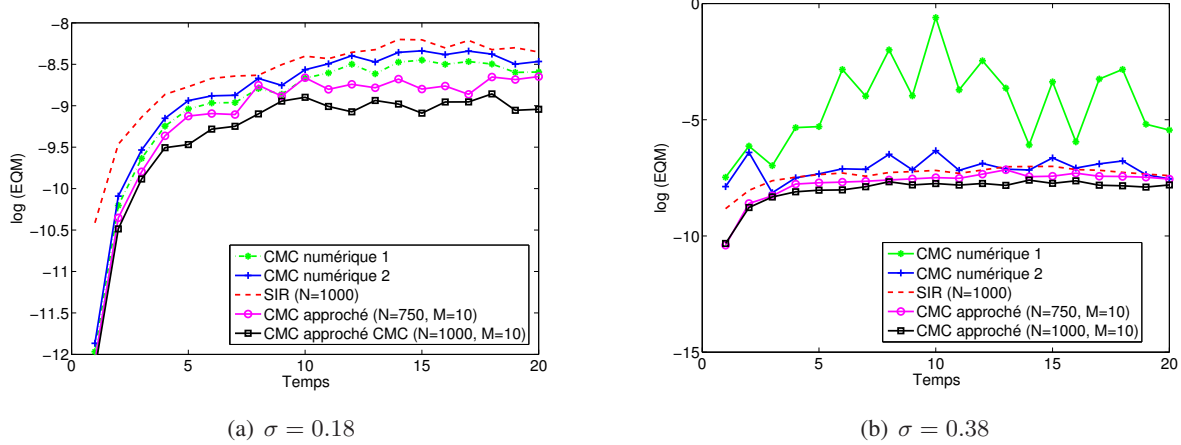


FIGURE 3.6 – EQM - Modèle SV - $\Phi = 0.8$, $\beta = 0.6$. Les estimateurs CMC basés sur une méthode MC locale sont préférables à l'estimateur MC classique, même lorsque $N \leq 1000$; les approximations numériques de l'estimateur CMC sont d'intérêts pratiques tant que σ est faible, auquel cas les linéarisations utilisées permettent d'approcher correctement les quantités recherchées.

3.3.2 Estimateurs CMC pour des modèles à sauts (classe 2)

Nous nous plaçons maintenant dans le contexte des modèles à sauts et nous considérons toujours le problème de l'estimation de $\Theta_n = \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{y}_{0:n}) d\mathbf{x}_n$ (l'intérêt de la notation $\phi(\cdot)$ pour la fonction d'intérêt apparaîtra ci-dessous). Nous avons rappelé dans la Section 1.1.2 que le problème de l'approximation de Θ_n pouvait être divisé en deux classes. Nous considérons tout d'abord la seconde classe de modèles, dans laquelle l'approximation de (3.1.5) est déduite de

$$\Theta_n = \sum_{\mathbf{r}_{0:n}} \underbrace{p(\mathbf{r}_{0:n} | \mathbf{y}_{0:n})}_{\text{FP}} \int \phi(\mathbf{x}_n) \underbrace{p(\mathbf{x}_n | \mathbf{r}_{0:n}, \mathbf{y}_{0:n})}_{\text{FK}} d\mathbf{x}_n. \quad (3.3.39)$$

Dans ce cas, (3.3.39) coïncide avec (3.3.2) (l'intégrale est remplacée par une somme puisque R_n est une v.a. discrète), sous l'identification : $X_1 = \mathbf{R}_{0:n-1}$, $X_2 = R_n$, $f(x_1, x_2) = \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{r}_{0:n-1}, r_n, \mathbf{y}_{0:n}) d\mathbf{x}_n$, et $p(x_1, x_2)$ est la loi jointe a posteriori

$$p(\mathbf{r}_{0:n} | \mathbf{y}_{0:n}) = \underbrace{p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})}_{p(x_1)} \underbrace{p(r_n | \mathbf{r}_{0:n-1}, \mathbf{y}_{0:n})}_{p(x_2|x_1)}. \quad (3.3.40)$$

Les deux facteurs doivent être alors calculés (notons que la chaîne marginale (R_n, \mathbf{Y}_n) n'est pas un HMC). Une approximation de $p(\mathbf{x}_1)$, c'est à dire de

$$p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n | \mathbf{r}_{0:n-1}, \mathbf{y}_{0:n-1}) p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n-1})}{\sum_{\mathbf{r}_{0:n-1}} p(\mathbf{y}_n | \mathbf{r}_{0:n-1}, \mathbf{y}_{0:n-1}) p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n-1})}. \quad (3.3.41)$$

peut être obtenue une nouvelle fois par le mécanisme de Rubin, si l'on dispose d'une approximation MC de $p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n-1})$. Partant alors de $\hat{p}(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{r}_{0:n-1}^i}$, nous avons l'approximation suivante : $\hat{p}(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i \delta_{\mathbf{r}_{0:n-1}^i}$, avec

$$\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i), \quad \sum_{i=1}^N \tilde{w}_{n-1}^i = 1. \quad (3.3.42)$$

Le facteur $p(x_2|x_1)$ dans (3.3.40) peut se réécrire sous la forme

$$p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n)p(r_n|r_{n-1}^i)}{\sum_{r_n} p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n)p(r_n|r_{n-1}^i)}. \quad (3.3.43)$$

Enfin, en posant $f(\mathbf{r}_{0:n-1}^i, r_n) = \mathbb{E}(\phi(\mathbf{x}_n)|\mathbf{y}_{0:n}, \mathbf{r}_{0:n-1}^i, r_n)$ et en tirant $r_n^i \sim p(r_n|\mathbf{y}_{0:n}, \mathbf{r}_{0:n-1}^i)$, les estimateurs purement MC et CMC s'écrivent respectivement

$$\hat{\Theta}_n(\mathbf{r}_{0:n-1}^{1:N}, \mathbf{r}_n^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i(\mathbf{r}_{0:n-1}^{1:N}) f(\mathbf{r}_{0:n-1}^i, r_n^i), \quad (3.3.44)$$

$$\tilde{\Theta}_n(\mathbf{r}_{0:n-1}^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i(\mathbf{r}_{0:n-1}^{1:N}) \sum_{r_n} f(\mathbf{r}_{0:n-1}^i, r_n) p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n}). \quad (3.3.45)$$

Calcul pratique de $\tilde{\Theta}_n$

L'implémentation de $\tilde{\Theta}_n$ requiert le calcul des équations (3.3.42) et (3.3.43) ainsi que celui de l'espérance conditionnelle $f(\mathbf{r}_{0:n-1}^i, r_n)$. Nous devons ainsi calculer $p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n)$, ce qui n'est généralement pas possible dans les modèles à sauts. Supposons alors que le modèle à sauts soit linéaire et gaussien, c'est à dire que les transitions et vraisemblances vérifient (3.1.15)-(3.1.16). Dans ces modèles, $p(\mathbf{x}_{n-1}|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_{n-1}; \mathbf{m}_{n-1|n-1}^i; \mathbf{P}_{n-1|n-1}^i)$. Par conséquent, $p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n)$ est déduit des étapes de prédiction du FK,

$$p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n) = \mathcal{N}(\mathbf{y}_n; \tilde{\mathbf{y}}_n^i(r_n); \mathbf{S}_n^i(r_n)), \quad (3.3.46)$$

où

$$\tilde{\mathbf{y}}_n^i(r_n) = \mathbf{y}_n - \mathbf{H}_n(r_n)\mathbf{F}_n(r_n)\mathbf{m}_{n-1|n-1}^i, \quad (3.3.47)$$

$$\mathbf{S}_n^i(r_n) = \mathbf{H}_n(r_n)\mathbf{P}_{n|n-1}^i(r_n)\mathbf{H}_n(r_n)^T + \mathbf{R}_n(r_n) \quad (3.3.48)$$

$$\mathbf{P}_{n|n-1}^i(r_n) = \mathbf{F}_n(r_n)\mathbf{P}_{n-1|n-1}^i\mathbf{F}_n^T(r_n) + \mathbf{Q}_n(r_n). \quad (3.3.49)$$

En résumé, (3.3.46)-(3.3.49) permettent le calcul de (3.3.42), (3.3.43), et finalement de (3.3.45).

Remarque 3.6. L'estimateur $\tilde{\Theta}_n$ dans (3.3.45) est l'estimateur CMC de $\hat{\Theta}_n$ dans (3.3.44), qui lui-même est un estimateur RB pour les modèles à sauts [44]. En effet, $\hat{\Theta}_n$ correspond ici à $\hat{\Theta}^{\text{RB}}$ dans (3.1.7) avec $\mathbf{x}_1 = \mathbf{r}_{0:n}$, $\mathbf{x}_2 = \mathbf{x}_{0:n}$, $f(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_n)$ et $q_1(r_n|\mathbf{r}_{0:n-1}) = p(r_n|\mathbf{r}_{0:n-1}, \mathbf{y}_{0:n})$. Par conséquent, (3.3.45) peut être vu comme une étape RB supplémentaire d'un estimateur RB ; l'étape RB qui donne (3.3.44) est spatiale, tandis que cette seconde étape RB est temporelle puisque le FP agit seulement sur les variables $\mathbf{r}_{0:n-1}$, plutôt que sur $\mathbf{r}_{0:n}$. Voici donc un exemple dans lequel les deux techniques de Rao-Blackwellisation (temporel + spatial) peuvent être combinées.

Remarque 3.7. Observons que $\tilde{\Theta}_n$ est calculable dès lors que $\hat{\Theta}_n$ l'est aussi. Par conséquent, cette réduction de variance peut être faite sous les mêmes hypothèses que celles sous lesquelles l'estimateur RB classique est calculable. D'autre part, cette variance induit un coût de calcul supplémentaire, qui n'est pas prohibitif à condition que K (qui représente le nombre de modes possibles) ne soit pas trop grand, comme nous pouvons le voir à partir de (3.3.44) et de (3.3.45). En effet, les poids \tilde{w}_{n-1}^i dans (3.3.42) doivent être calculés dans chacun des algorithmes. Ensuite, pour chaque i , $1 \leq i \leq N$, les deux algorithmes calculent $\{p(r_n | \mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})\}_{r_n=1}^K$. La différence vient alors du fait que l'estimateur CMC calcule directement $\sum_{r_n} f(\mathbf{r}_{0:n-1}^i, r_n) p(r_n | \mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})$, qui requiert K étapes de mise à jour du FK par trajectoire $\mathbf{r}_{0:n-1}^i$, tandis que l'estimateur classique est basé sur l'extension aléatoire de chaque trajectoire, avant le calcul des espérances conditionnelles.

Remarque 3.8. Notons que l'estimateur CMC repose sur une approximation particulière de $p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})$, caractérisée par les poids dans (3.3.42). Comme dans le paragraphe précédent, nous pouvons utiliser des approximations alternatives de $p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})$, basées par exemple sur l'équation de propagation

$$p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n}) \propto p(\mathbf{r}_{0:n-2} | \mathbf{y}_{0:n-1}) p(\mathbf{r}_{n-1} | \mathbf{r}_{0:n-2}, \mathbf{y}_{0:n-1}) p(\mathbf{y}_n | \mathbf{r}_{0:n-1}, \mathbf{y}_{0:n-1}). \quad (3.3.50)$$

En particulier, la densité optimale proportionnelle à $p(\mathbf{r}_{n-1} | \mathbf{r}_{0:n-2}, \mathbf{y}_{0:n-1}) p(\mathbf{y}_n | \mathbf{r}_{0:n-1}, \mathbf{y}_{0:n-1})$, pour laquelle une approximation de $p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})$ ne dépendrait pas de $\{\mathbf{r}_{n-1}^i\}_{i=1}^N$ est calculable d'après (3.3.43) et (3.3.46). Partant alors de $\hat{p}(\mathbf{r}_{0:n-2} | \mathbf{y}_{0:n-1}) = \sum_{i=1}^N \tilde{w}_{n-2}^i \delta_{\mathbf{r}_{0:n-2}^i}$, $r_{n-1}^i \sim q^{\text{opt}}(r_{n-1} | \mathbf{r}_{0:n-2}^i)$, $q^{\text{opt}}(r_{n-1} | \mathbf{r}_{0:n-2}^i) \propto p(r_{n-1} | \mathbf{r}_{0:n-2}^i, \mathbf{y}_{0:n-1}) p(\mathbf{y}_n | \mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n-1})$, alors une approximation de $p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})$ est donnée par

$$\hat{p}(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i \delta_{\mathbf{r}_{0:n-1}^i}, \quad (3.3.51)$$

$$\tilde{w}_{n-1}^i \propto \tilde{w}_{n-2}^i \sum_{r_{n-1}} p(r_{n-1} | \mathbf{r}_{0:n-2}^i, \mathbf{y}_{0:n-1}) p(\mathbf{y}_n | \mathbf{r}_{0:n-1}^i, r_{n-1}, \mathbf{y}_{0:n-1}), \quad \sum_{i=1}^N \tilde{w}_{n-1}^i = 1. \quad (3.3.52)$$

L'estimateur CMC correspondant est alors donné par

$$\tilde{\Theta}_n = \sum_{i=1}^N \tilde{w}_{n-1}^i \sum_{r_n} f(\mathbf{r}_{0:n-1}^i, r_n) p(r_n | \mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n}).$$

Remarque 3.9. Notre estimateur CMC repose lui même sur le FP marginalisé spatialement, qui suppose la linéarité et la gaussiannité du modèle à sauts. Si le modèle n'est plus linéaire, mais que les non linéarités ne sont pas trop sévères, l'espérance $E(\phi(\mathbf{x}_n) | \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ peut être approchée par EKF/UKF afin de pouvoir calculer ensuite $\tilde{\Theta}_n$ en utilisant une approximation de $p(\mathbf{y}_n | \mathbf{y}_{0:n-1}, \mathbf{r}_{0:n})$ à partir de l'EKF ou l'UKF.

Dans le cas contraire, nous pouvons approcher dans un premier temps le filtre marginalisé spatialement en utilisant des FP locaux (voir notre discussion dans la sous-section du Paragraphe 3.3.1 dédié aux méthodes MC locales) [65] puis appliquer notre marginalisation temporelle. Plus précisément, l'estimateur CMC repose ici sur les poids définis dans (3.3.42) et sur les intégrales $\int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{r}_{0:n-1}^i, r_n, \mathbf{y}_{0:n}) d\mathbf{x}_n$. Si ces quantités ne sont pas calculables, une approximation peut être faite à partir du filtre particulaire local proposé dans [65]; néanmoins, cette méthode locale doit être maintenant faite pour tout $r_n \in \{1, \dots, K\}$ et pas seulement pour la particule r_n^i qui a été tirée.

Simulations

Nous testons à présent notre approche sur le modèle décrit par (3.1.15)-(3.1.16) avec

$$\mathbf{F}_n(r) = \begin{bmatrix} 1 & \frac{\sin(\omega_r T_s)}{\omega_r} & 0 & -\frac{1-\cos(\omega_r T_s)}{\omega_r} \\ 0 & \cos(\omega_r T_s) & 0 & -\sin(\omega_r T_s) \\ 0 & \frac{1-\cos(\omega_r T_s)}{\omega_r} & 1 & \frac{\sin(\omega_r T_s)}{\omega_r} \\ 0 & \sin(\omega_r T_s) & 0 & \cos(\omega_r T_s) \end{bmatrix}, \mathbf{Q}_n = \sigma_v^2 \begin{bmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} & 0 & 0 \\ \frac{T_s^2}{2} & T_s & 0 & 0 \\ 0 & 0 & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ 0 & 0 & \frac{T_s^2}{2} & T_s \end{bmatrix},$$

$\mathbf{H}_n = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, $\mathbf{R}_n = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$, $T_s = 2$, $\sigma_v = 3$ et $\sigma_x = \sigma_y = 10m$. Nous pistons un objet manoeuvrant décrit par sa position et sa vitesse en coordonnées cartésiennes, c'est à dire $\mathbf{x}_n = [p_{x,n}, \dot{p}_{x,n}, p_{y,n}, \dot{p}_{y,n}]^T$. Le mode r_n représente le comportement de la cible : trajectoire rectiligne, virage à gauche et virage à droite. Nous posons $\omega_1 = 0 \text{ rad.s}^{-1}$, $\omega_2 = 3\pi/180 \text{ rad.s}^{-1}$ et $\omega_3 = -3\pi/180 \text{ rad.s}^{-1}$. Les probabilités de transition de la chaîne $\{R_n\}$ sont données $p(r_n|r_{n-1}) = 0.6$ si $r_n = r_{n-1}$ et $p(r_n|r_{n-1}) = 0.2$ si $r_n \neq r_{n-1}$.

Nous comparons l'estimateur basé sur l'algorithme SIR, avec densité optimale d'importance ainsi que deux estimateurs CMC qui ne diffèrent que par l'approximation de $p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n})$. Le premier repose sur (3.3.42) (estimateur CMC 1) tandis que dans le second (estimateur CMC 2), $p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n})$ est propagé par un algorithme d'échantillonnage d'importance dans lequel les particules sont tirées suivant la loi optimale proportionnelle à $p(\mathbf{r}_{n-1}|\mathbf{r}_{0:n-2}, \mathbf{y}_{0:n-1})p(\mathbf{y}_n|\mathbf{r}_{0:n-1}, \mathbf{y}_{0:n-1})$ (voir Remarque 3.8).

Les EQM des différents estimateurs sont tracées sur la Figure 3.7. Comme prévu, les estimateurs CMC donnent de meilleurs résultats que l'estimateur SIR. Le second estimateur CMC basé sur la propagation de $p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n})$ améliore l'estimateur basé sur la propagation de $p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n-1})$, ce qui est un point particulièrement intéressant. Comme nous l'avons vu en introduction de cette section, le calcul des estimateurs CMC requiert un coût de calcul supplémentaire par rapport à l'estimateur classique. Nous calculons et moyennons sur le temps l'efficacité définie par $\text{Eff}(n) = 1/(\text{MSE}(n)\text{E}(C(n)))$, où $C(n)$ est le temps d'exécution pour le calcul d'un estimateur donné [74]. Pour le premier estimateur CMC, nous obtenons 3×10^4 , pour le second 2×10^4 et pour l'estimateur classique 6×10^3 . Les estimateurs CMC sont alors plus efficaces que l'estimateur classique mais remarquons que le second estimateur CMC requiert un coût de calcul supplémentaire par rapport au premier. Cela est dû au fait que $p(\mathbf{y}_n|\mathbf{r}_{0:n-2}^i, \mathbf{r}_{n-1}, \mathbf{y}_{0:n-1})$ dans (3.3.52) doit être calculé pour tout r_{n-1} .

3.3.3 Estimateurs CMC pour des modèles à sauts (classe 1)

Dans ce paragraphe, nous nous orientons vers les modèles à sauts généraux sans faire d'hypothèses sur la forme du modèle. Nous supposons cette fois qu'il n'est pas possible d'approcher $\text{E}(\phi(\mathbf{x}_n)|\mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ avec un coût raisonnable. Par conséquent, nous nous plaçons dans la première classe de modèles SMC pour les modèles à sauts, dans laquelle une approximation MC de $p(\mathbf{x}_{0:n}, \mathbf{r}_{0:n}|\mathbf{y}_{0:n})$ est propagée [44] [6].

Expression des estimateurs CMC

Réécrivons d'abord Θ_n comme

$$\Theta_n = \sum_{\mathbf{r}_{0:n-1}, r_n} \int \phi(\mathbf{x}_n) p(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1}, \mathbf{x}_n, r_n | \mathbf{y}_{0:n}) d\mathbf{x}_{0:n-1} dx_n. \quad (3.3.53)$$

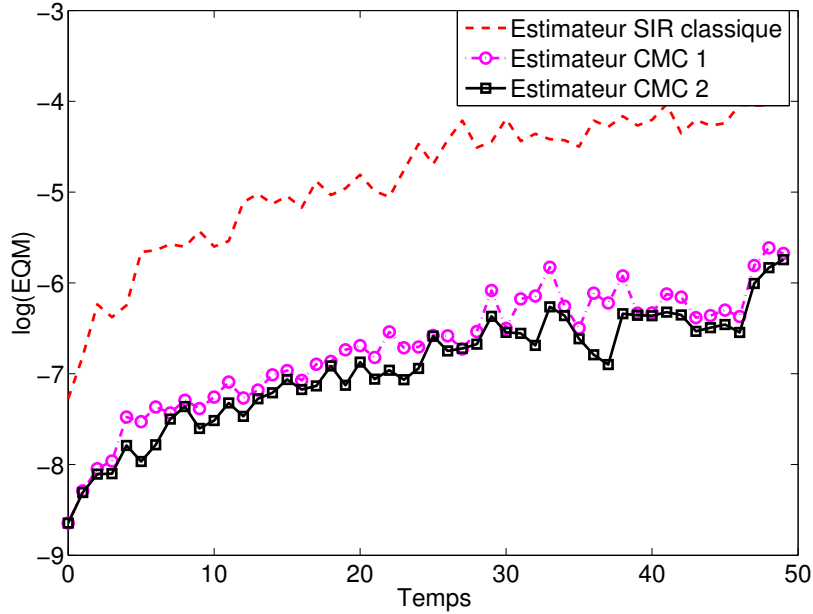


FIGURE 3.7 – EQM - Modèle à saut linéaire et gaussien - $\phi(\mathbf{x}_n) = \mathbf{x}_n$. Les estimateurs CMC sont préférables à l'estimateur classique. L'estimateur basé sur la propagation directe de $p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n})$ (estimateur CMC 2) est plus performant, au prix d'un coût de calcul plus important.

Soit maintenant $\sum_{i=1}^N w_{n-1}^i(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}) \delta_{\mathbf{x}_{0:n-1}^i, \mathbf{r}_{0:n-1}^i}$ une approximation MC de la loi $p(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1} | \mathbf{y}_{0:n-1})$. Alors $\hat{p}(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1} | \mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i, \mathbf{r}_{0:n-1}^i}$, avec

$$\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_{n-1}^i), \quad \sum_{i=1}^N \tilde{w}_{n-1}^i = 1 \quad (3.3.54)$$

est une approximation MC de $p(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})$. Posons également

$$(\mathbf{x}_n^i, r_n^i) \sim p(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n). \quad (3.3.55)$$

Alors l'estimateur purement MC est donné par [44] [6] :

$$\hat{\Theta}_n(\mathbf{x}_{0:n}^{1:N}, \mathbf{r}_{0:n}^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}) \phi(\mathbf{x}_n^i). \quad (3.3.56)$$

Nous proposons deux estimateurs CMC de Θ_n , associés à deux partitions différentes de $(\mathbf{X}_{0:n}, \mathbf{R}_{0:n})$. Le choix $X_1 = (\mathbf{X}_{0:n-1}, \mathbf{R}_{0:n})$ et $X_2 = \mathbf{X}_n$ donne $\tilde{\Theta}_n^{\mathbf{X}_n}$; le choix $X_1 = (\mathbf{X}_{0:n-1}, \mathbf{R}_{0:n-1})$, $X_2 = (\mathbf{X}_n, R_n)$ et $f(\mathbf{x}_{n-1}^i, \mathbf{y}_n, \mathbf{r}_n) = \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n, \mathbf{y}_n) d\mathbf{x}_n$ donne $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$, avec

$$\tilde{\Theta}_n^{\mathbf{X}_n}(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n}^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i f(\mathbf{x}_{n-1}^i, \mathbf{y}_n, \mathbf{r}_n^i), \quad (3.3.57)$$

$$\begin{aligned} \tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}) &= \sum_{i=1}^N \tilde{w}_{n-1}^i \sum_{r_n} \int \phi(\mathbf{x}_n) p(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n \\ &= \sum_{i=1}^N \tilde{w}_{n-1}^i \sum_{r_n} p(r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n) f(\mathbf{x}_{n-1}^i, \mathbf{y}_n, \mathbf{r}_n), \end{aligned} \quad (3.3.58)$$

où $\tilde{w}_{n-1}^i(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N})$ est défini dans (3.3.54), et $r_n^i \sim p(r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n)$ dans (3.3.57) (une marginale de (3.3.55)). Bien sûr, $\tilde{\Theta}_n^{\mathbf{X}_n} = E(\hat{\Theta}_n | \mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}, \mathbf{y}_{0:n})$ et $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)} = E(\tilde{\Theta}_n^{\mathbf{X}_n} | \mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}, \mathbf{y}_{0:n})$, par conséquent $\text{var}(\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}) \leq \text{var}(\tilde{\Theta}_n^{\mathbf{X}_n}) \leq \text{var}(\hat{\Theta}_n)$.

Calcul pratique de $\tilde{\Theta}_n^{\mathbf{X}_n}$ et $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$

Discutons à présent du calcul pratique de (3.3.57) et de (3.3.58). Dans le modèle JMSS

$$\begin{aligned} p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_{n-1}^i) &= \sum_{r_n} p(r_n | r_{n-1}^i) p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_n), \\ p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_{n-1}^i) p(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n) &= p(r_n | r_{n-1}^i) p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n, \mathbf{y}_n). \end{aligned}$$

Posons alors

$$\bar{w}_{n-1}^i(r_n) = \frac{w_{n-1}^i p(r_n | r_{n-1}^i) p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_n)}{\sum_{r_n} \sum_{i=1}^N w_{n-1}^i p(r_n | r_{n-1}^i) p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_n)} \quad (3.3.59)$$

(notons que $\sum_{r_n} \bar{w}_{n-1}^i(r_n) = \tilde{w}_{n-1}^i$). Alors $\tilde{\Theta}_n^{\mathbf{X}_n}$ et $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$ peuvent être réécrits comme

$$\tilde{\Theta}_n^{\mathbf{X}_n} = \sum_{i=1}^N \left[\sum_{r_n} \bar{w}_{n-1}^i(r_n) \right] \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n, r_n) d\mathbf{x}_n, \quad (3.3.60)$$

$$\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)} = \sum_{i=1}^N \left[\sum_{r_n} \bar{w}_{n-1}^i(r_n) \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n, r_n) d\mathbf{x}_n \right]. \quad (3.3.61)$$

Par conséquent, (3.3.61) est calculable dès lors que (3.3.60) est calculable. De plus, $\tilde{\Theta}_n^{\mathbf{X}_n}$ généralise $\tilde{\Theta}_n$ dans (3.3.13) : $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n, r_n)$ et $p(\mathbf{y}_n | \mathbf{x}_{n-1}, r_n)$ jouent les mêmes rôles que $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$ et $p(\mathbf{y}_n | \mathbf{x}_{n-1})$ dans le Paragraphe 3.3.1, excepté que nous avons maintenant introduit une dépendance en r_n . Cela signifie que $\tilde{\Theta}_n^{\mathbf{X}_n}$ et $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$ sont calculables dès lors que l'estimateur bayésien CMC (3.3.1) est calculable dans le modèle HMC sous-jacent (c'est à dire le modèle HMC auquel est réduit le modèle JMSS, lorsque les sauts sont fixés). Par exemple, les modèles semi-linéaires (incluant les modèles de ARCH) à sauts sont une classe de modèles dans laquelle (3.3.60) et (3.3.61) sont calculables.

La seule différence entre les estimateurs (3.3.60) et (3.3.61) est le coût de calcul nécessaire dont nous discutons maintenant. Pour un i donné dans (3.3.61), le calcul de $\int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n, \mathbf{y}_n) d\mathbf{x}_n$ doit être fait pour tout r_n , alors que dans (3.3.60), le calcul n'est fait que pour la particule r_n^i qui vient d'être tirée. Comme nous nous y attendions $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$ a une variance plus faible que $\tilde{\Theta}_n^{\mathbf{X}_n}$ mais requiert un coût de calcul supplémentaire. D'un autre côté, comparer le coût de calcul de $\tilde{\Theta}_n^{\mathbf{X}_n}$ et $\hat{\Theta}_n$ revient à comparer celui de l'estimateur CMC (3.3.13) avec celui de l'estimateur MC (3.3.12) dans le Paragraphe 3.3.1 et est donc dépendant du problème traité. Cependant, nous pouvons observer que dans le cas décrit à la Remarque 3.3 (c'est à dire lorsque le tirage de points suivant $p(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, \mathbf{y}_n)$ requiert le calcul de $\int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, \mathbf{y}_n) d\mathbf{x}_n$), alors le calcul de $\tilde{\Theta}_n^{\mathbf{X}_n}$ n'induit pas de coût de calcul supplémentaire comparé à $\hat{\Theta}_n$.

Calcul approché

À présent, nous discutons du calcul approché de $\tilde{\Theta}_n$ lorsque $p(\mathbf{y}_n | \mathbf{x}_{n-1}, r_n)$ et $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n, r_n)$ dans (3.3.60) (3.3.61) ne sont pas connues. D'abord, notons que (3.3.61) peut être calculé à partir des mêmes approximations numériques que celles utilisées pour le calcul de (3.3.13) dans le Paragraphe 3.3.1, excepté

qu'elles doivent être faites pour toutes les valeurs de r_n . Cependant, comme r_n est discret, d'autres méthodes d'approximations peuvent être développées :

- Dans (3.1.14), la loi conditionnelle à $(\mathbf{x}_{n-1}, r_{n-1})$ du triplet $(\mathbf{X}_n, \mathbf{Y}_n, R_n)$ se factorise sous la forme $p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, r_{n-1}) = p(r_n | r_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n)$; par conséquent, le numérateur de (3.3.61) peut se réécrire

$$\sum_{r_n} \int \phi(\mathbf{x}_n) \left[\sum_{i=1}^N p(r_n | r_{n-1}^i) w_{n-1}^i f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) \right] d\mathbf{x}_n.$$

Si pour un r_n donné l'intégrale n'est pas calculable, il est possible de l'approximer en tirant $\mathbf{x}_n^{r_n, i} \sim q(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n)$ pour tout $i, 1 \leq i \leq N$ et pour tout r_n . Une approximation du numérateur est alors donnée par

$$\sum_{r_n} \sum_{i=1}^N \phi(\mathbf{x}_n^{r_n, i}) p(r_n | r_{n-1}^i) w_{n-1}^i f_{n|n-1}(\mathbf{x}_n^{r_n, i} | \mathbf{x}_{n-1}^i, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n^{r_n, i}, r_n) / q(\mathbf{x}_n^{r_n, i} | \mathbf{x}_{n-1}^i, r_n).$$

Par conséquent, nous n'utilisons pas de tirages pour la partie discrète r_n . Nous appliquons enfin la même approximation pour le dénominateur qui peut être réécrit

$$\sum_{r_n} \sum_{i=1}^N w_{n-1}^i p(r_n | r_{n-1}^i) \int f_{n|n-1}(\mathbf{x}_n | r_n, \mathbf{x}_{n-1}^i) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) d\mathbf{x}_n ;$$

- Dans le cas général, l'expression de l'estimateur basé sur l'algorithme SIS/SIR s'écrit

$$\widehat{\Theta}_n^{\text{SIR}} = \sum_{i=1}^N \frac{w_n^i}{\sum_{i=1}^N w_n^i} \phi(\mathbf{x}_n^i), \quad (3.3.62)$$

$$w_n^i \propto w_{n-1}^i \frac{p(r_n^i | r_{n-1}^i) f_{n|n-1}(\mathbf{x}_n^i | \mathbf{x}_{n-1}^i, r_n^i) g_n(\mathbf{y}_n | \mathbf{x}_n^i, r_n^i)}{q(\mathbf{x}_n^i, r_n^i | \mathbf{x}_{n-1}^i, r_{n-1}^i)}, \quad \sum_{i=1}^N w_n^i = 1, \quad (3.3.63)$$

où (\mathbf{x}_n^i, r_n^i) est tiré selon $q(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i) = q(r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i) q(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, r_n)$. Puisque $p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, r_{n-1})$ n'est pas calculable, nous essayons alors d'agir directement sur la variance de $\widehat{\Theta}_n^{\text{SIR}}$. Remarquons qu'il n'est pas possible de calculer $E(\widehat{\Theta}_n^{\text{SIR}} | \{\mathbf{x}_{n-1}^i, \mathbf{x}_n^i, r_{n-1}^i\}_{i=1}^N, \mathbf{y}_n)$, c'est à dire l'espérance de $\widehat{\Theta}_n^{\text{SIR}}$ en tant que fonction de r_n . En effet, puisque $\phi(\cdot)$ ne dépend pas de r_n , le calcul de l'espérance est ramené au calcul de l'espérance des poids normalisés. Cependant, il est possible de calculer séparément l'espérance conditionnelle du numérateur et du dénominateur, ce qui est d'autant plus facile que r_n prend ses valeurs dans un ensemble discret et que

$$q(r_n | \mathbf{x}_n, \mathbf{x}_{n-1}, r_{n-1}) = \frac{q(r_n | \mathbf{x}_{n-1}, r_{n-1}) q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_{n-1}, r_n)}{\sum_{r_n} q(r_n | \mathbf{x}_{n-1}, r_{n-1}) q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_{n-1}, r_n)}$$

est connue. Cette réduction de variance des poids non normalisés peut être vue comme une implémentation par échantillonnage d'importance de (3.3.61), réécrite sous la forme

$$\widetilde{\Theta}_n = \frac{\int \phi(\mathbf{x}_n) \sum_{i=1}^N \left[\sum_{r_n} p(r_n | r_{n-1}^i) w_{n-1}^i f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) \right] d\mathbf{x}_n}{\int \sum_{i=1}^N \left[\sum_{r_n} p(r_n | r_{n-1}^i) w_{n-1}^i f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) \right] d\mathbf{x}_n}, \quad (3.3.64)$$

avec la loi d'importance

$$q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_{n-1}) = \sum_{r_n} q(r_n | \mathbf{x}_{n-1}, r_{n-1}) q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, r_{n-1}).$$

Notons que le calcul des nouveaux poids n'est pas prohibitif tant que $K \ll N$. En résumé, bien qu'il ne soit pas possible d'agir directement sur la variance de $\hat{\Theta}_n^{\text{SIR}}$, il est toujours possible de réduire la variance des poids associés au calcul de $\hat{\Theta}_n^{\text{SIR}}$ dans un modèle à sauts (3.1.14). Les analyses existantes comparant (3.1.6)-(3.1.7) sont alors valides pour comparer $\hat{\Theta}_n^{\text{SIR}}$ et $\tilde{\Theta}_n$.

- Enfin, l'approximation générale basée sur une méthode MC locale proposée au Paragraphe 3.3.1 est aussi valide pour approcher $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$, en adaptant $\mathbf{X}_n \leftarrow (\mathbf{X}_n, R_n)$. En effet, $p(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})$ peut être propagée par une méthode d'échantillonnage d'importance dans laquelle les poids sont déduits à partir d'une approximation MC locale de $\sum_{r_n} \int \phi(\mathbf{x}_n) p(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n$.

3.3.4 Estimateurs CMC pour le filtrage multi-objets

Dans ce dernier paragraphe, nous appliquons les méthodes CMC temporelles au problème de filtrage multi-objets. Quelques adaptations sont nécessaires ici, car nous ne travaillons plus avec des densités de probabilités. Cependant, la discussion de l'Introduction de cette section reste valide, comme nous allons le voir.

Estimateur bayésien CMC PHD

Le problème que nous traitons est le calcul du moment $\Theta_n = \int f(\mathbf{x}_n) v_n(\mathbf{x}_n) d\mathbf{x}_n$ (typiquement, nous prendrons $f(\mathbf{x}_n) = 1$ ou $f(\mathbf{x}_n) = \mathbb{1}_S(\mathbf{x}_n)$, où S est une région d'intérêt). Nous supposons que $p_{d,n}(\mathbf{x}_n) = p_{d,n}$ est indépendant de \mathbf{x}_n . Repartons des équations du PHD (3.2.1)-(3.2.2). En injectant (3.2.1) dans (3.2.2), le PHD à l'instant n peut s'écrire comme

$$v_n(\mathbf{x}_n) = \sum_{i=1}^4 v_n^i(\mathbf{x}_n), \quad (3.3.65)$$

où

$$v_n^1(\mathbf{x}_n) = [1 - p_{d,n}] \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1}, \quad (3.3.66)$$

$$v_n^2(\mathbf{x}_n) = [1 - p_{d,n}] \gamma_n(\mathbf{x}_n), \quad (3.3.67)$$

$$v_n^3(\mathbf{x}_n) = \sum_{\mathbf{z} \in \mathcal{Z}_n} \frac{p_{d,n} g_n(\mathbf{z} | \mathbf{x}_n) \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1}}{B_n(\mathbf{z})}, \quad (3.3.68)$$

$$v_n^4(\mathbf{x}_n) = \sum_{\mathbf{z} \in \mathcal{Z}_n} \frac{p_{d,n} g_n(\mathbf{z} | \mathbf{x}_n) \gamma_n(\mathbf{x}_n)}{B_n(\mathbf{z})}, \quad (3.3.69)$$

et où

$$B_n(\mathbf{z}) = \kappa_n(\mathbf{z}) + B_n^1(\mathbf{z}) + B_n^2(\mathbf{z}), \quad (3.3.70)$$

$$\begin{aligned} B_n^1(\mathbf{z}) &= \int p_{d,n} g_n(\mathbf{z} | \mathbf{x}_n) \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1} d\mathbf{x}_n \\ &= p_{d,n} \int p_{s,n}(\mathbf{x}_{n-1}) p(\mathbf{z} | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1}, \end{aligned} \quad (3.3.71)$$

$$B_n^2(\mathbf{z}) = \int p_{d,n} g_n(\mathbf{z} | \mathbf{x}_n) \gamma_n(\mathbf{x}_n) d\mathbf{x}_n. \quad (3.3.72)$$

Le terme v_n^1 (resp. v_n^2) est dû aux objets persistants (resp. naissants) non détectés, tandis que v_n^3 (resp. v_n^4) est dû aux objets persistants (resp. naissants) détectés.

A partir de (3.3.65), nous avons

$$\Theta_n = \sum_{i=1}^4 \underbrace{\int f(\mathbf{x}_n) v_n^i(\mathbf{x}_n) d\mathbf{x}_n}_{\Theta_n^i}, \quad (3.3.73)$$

et nous nous concentrons sur l'adaptation de la méthodologie proposée dans l'introduction de cette section pour les moments Θ_n^i . Tout d'abord, notons que $v_n^2(\mathbf{x}_n)$ et $v_n^4(\mathbf{x}_n)$ ne dépendent pas de $v_{n-1}(\mathbf{x}_{n-1})$; par conséquent, nous utilisons une méthode MC classique pour approcher Θ_n^2 et Θ_n^4 . Soient $\hat{v}_{n-1} = \sum_{i=1}^{L_{n-1}} w_{n-1}^i \delta_{\mathbf{x}_{n-1}^i}$ et $\hat{\gamma}_n = \sum_{i=1}^{L_{\gamma_n}} w_{\gamma_n}^i \delta_{\mathbf{x}_{\gamma_n}^i}$ des approximations MC de $v_{n-1}(\mathbf{x}_{n-1})$ et de $\gamma_n(\mathbf{x}_n)$. Alors des estimateurs de Θ_n^2 et Θ_n^4 sont donnés par

$$\hat{\Theta}_n^2 = \sum_{i=1}^{L_{\gamma_n}} w_n^{2,i} f(\mathbf{x}_{\gamma_n}^i), \text{ avec } w_n^{2,i} = w_{\gamma_n}^i [1 - p_{d,n}], \quad (3.3.74)$$

$$\hat{\Theta}_n^4 = \sum_{i=1}^{L_{\gamma_n}} \left[\sum_{\mathbf{z} \in \mathcal{Z}_n} w_n^{4,i}(\mathbf{z}) \right] f(\mathbf{x}_{\gamma_n}^i), \text{ avec } w_n^{4,i}(\mathbf{z}) = w_{\gamma_n}^i \frac{p_{d,n} g_n(\mathbf{z} | \mathbf{x}_{\gamma_n}^i)}{\tilde{B}_n(\mathbf{z})}, \quad (3.3.75)$$

$$\tilde{B}_n(\mathbf{z}) = \kappa_n(\mathbf{z}) + p_{d,n} \sum_{i=1}^{L_{\gamma_n}} w_{\gamma_n}^i g_n(\mathbf{x}_{\gamma_n}^i | \mathbf{z}) + p_{d,n} \sum_{i=1}^{L_{n-1}} w_{n-1}^i p_{s,n}(\mathbf{x}_{n-1}^i) p(\mathbf{z} | \mathbf{x}_{n-1}^i). \quad (3.3.76)$$

Par ailleurs, le calcul de $v_n^1(\mathbf{x}_n)$ et de $v_n^3(\mathbf{x}_n)$ dépend de $v_{n-1}(\mathbf{x}_{n-1})$, ce qui suggère d'adapter la méthodologie générale de l'introduction de la Section 3.3, bien que le PHD ne soit pas une densité de probabilité (il s'agit d'une fonction positive mais son intégrale n'est pas nécessairement égale à 1 d'après (1.2.9)) et que les poids $\{w_{n-1}^i\}_{i=1}^{L_{n-1}}$ puissent dépendre de variables différentes de $\mathbf{x}_{n-1}^{1:L_{n-1}}$, qui sont néanmoins connues à l'instant $n-1$. Ces différences mineures n'ont néanmoins aucun impact sur la discussion générale du début de section. En effet, nous avons $f_{n|n-1}(\mathbf{x}_n) g_n(\mathbf{z} | \mathbf{x}_n) = p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{z}) p(\mathbf{z} | \mathbf{x}_{n-1})$, donc Θ_n^1 et Θ_n^3 peuvent se réécrire sous la forme

$$\Theta_n^1 = [1 - p_{d,n}] \int \mathbb{E}(f(\mathbf{X}_n) | \mathbf{x}_{n-1}) \times [p_{s,n}(\mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1})] d\mathbf{x}_{n-1}, \quad (3.3.77)$$

$$\Theta_n^3 = \sum_{\mathbf{z} \in \mathcal{Z}_n} \int \mathbb{E}(f(\mathbf{X}_n) | \mathbf{x}_{n-1}, \mathbf{z}) \left[\frac{p_{d,n} p_{s,n}(\mathbf{x}_{n-1}) p(\mathbf{z} | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1})}{B_n(\mathbf{z})} \right] d\mathbf{x}_{n-1}. \quad (3.3.78)$$

Commençons par le calcul de Θ_n^1 dans (3.3.77). Même s'il n'est pas une densité de probabilité, le terme $p_{s,n}(\mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1})$ à l'intérieur des crochets joue le rôle de $p(x_1)$ dans (3.3.3), et peut être approché par $\sum_{i=1}^{L_{n-1}} w_n^{1,i} \delta_{\mathbf{x}_{n-1}^i}$ avec $w_n^{1,i} = [1 - p_{d,n}] p_{s,n}(\mathbf{x}_{n-1}^i) w_{n-1}^i$. L'estimateur purement MC et l'estimateur CMC de Θ_n^1 s'écrivent respectivement

$$\hat{\Theta}_{1,n} = \sum_{i=1}^{L_{n-1}} w_n^{1,i} f(\mathbf{x}_{n-1}^i), \quad (3.3.79)$$

$$\tilde{\Theta}_{1,n} = \sum_{i=1}^{L_{n-1}} w_n^{1,i} \mathbb{E}(f(\mathbf{x}_n) | \mathbf{x}_{n-1}^i), \quad (3.3.80)$$

avec $\mathbf{x}_n^i \sim f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$. Traitons maintenant le problème du calcul de Θ_n^3 dans (3.3.78). Pour chaque mesure $\mathbf{z} \in \mathcal{Z}_n$, le facteur $\frac{p_{d,n}p_{s,n}(\mathbf{x}_{n-1})p(\mathbf{z}|\mathbf{x}_{n-1})v_{n-1}(\mathbf{x}_{n-1})}{\tilde{B}_n(\mathbf{z})}$ à l'intérieur des crochets joue le rôle de $p(x_1)$ dans (3.3.3), et peut être approché par $\sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z})\delta_{\mathbf{x}_{n-1}^i}$ avec $w_n^{3,i}(\mathbf{z}) = p_{d,n}p_{s,n}(\mathbf{x}_{n-1}^i)p(\mathbf{z}|\mathbf{x}_{n-1}^i)w_{n-1}^i \times 1/\tilde{B}_n(\mathbf{z})$. Les estimateurs purement MC et CMC de Θ_n^3 s'écrivent respectivement

$$\hat{\Theta}_n^3 = \sum_{\mathbf{z}} \sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z})f(\mathbf{x}_n^{\mathbf{z},i}), \quad (3.3.81)$$

$$\tilde{\Theta}_n^3 = \sum_{\mathbf{z} \in \mathcal{Z}_n} \sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z})E(f(\mathbf{x}_n)|\mathbf{x}_{n-1}^i, \mathbf{z}), \quad (3.3.82)$$

avec $\mathbf{x}_n^{\mathbf{z},i} \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{z})$.

En résumé, l'estimateur $\hat{\Theta}_n$ MC de Θ_n est la somme de 4 estimateurs MC, $\hat{\Theta}_n = \sum_{i=1}^4 \hat{\Theta}_n^i$, tandis que l'estimateur CMC $\tilde{\Theta}_n$ est la somme de deux estimateurs MC et deux estimateurs CMC, $\tilde{\Theta}_n = \tilde{\Theta}_n^1 + \hat{\Theta}_n^2 + \tilde{\Theta}_n^3 + \hat{\Theta}_n^4$. Puisque $\tilde{\Theta}_n^1$ et $\tilde{\Theta}_n^3$ sont calculés à partir de la même approximation MC de $v_{n-1}(\mathbf{x}_{n-1})$, $\tilde{\Theta}_n = E(\hat{\Theta}_n|\{\mathbf{x}_{n-1}^i\}_{i=1}^{L_{n-1}}, \{\mathbf{x}_{\gamma_n}^i\}_{i=1}^{L_{\gamma_n}}, \mathcal{Z}_n)$. Par conséquent notre discussion introductive permet de conclure que $\tilde{\Theta}_n$ est en effet préférable à $\hat{\Theta}_n$.

Remarque 3.10. Le calcul de $\hat{\Theta}_n^1 + \hat{\Theta}_n^3$ implique le tirage de $L_{n-1}(|\mathcal{Z}_n| + 1)$ particules. Il est cependant possible de calculer une approximation reposant sur L_{n-1} particules en tirant $\mathbf{x}_n^i \sim q^i(\mathbf{x}_n)$ avec $q^i(\mathbf{x}_n) \propto w_n^{1,i}f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}^i) + \sum_{\mathbf{z}} w_n^{3,i}(\mathbf{z})p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{z})$ pour tout $i, 1 \leq i \leq L_{n-1}$, et en prenant

$$\hat{\Theta}_n^1 + \hat{\Theta}_n^3 = \sum_{i=1}^{L_{n-1}} \left[w_n^{1,i} + \sum_{\mathbf{z}} w_n^{3,i}(\mathbf{z}) \right] f(\mathbf{x}_n^i). \quad (3.3.83)$$

Remarque 3.11. Selon la forme de $\gamma_n(\mathbf{x}_n)$ et $g_n(\mathbf{z}|\mathbf{x}_n)$, $\int g_n(\mathbf{z}|\mathbf{x}_n)\gamma_n(\mathbf{x}_n)d\mathbf{x}_n$ peut être calculé directement ; de ce fait $B_n^2(\mathbf{z})$, $\Theta_{2,n}$ et $\Theta_{4,n}$ peuvent l'être également. Dans ce cas, $\tilde{B}_n(\mathbf{z})$ dans (3.3.76) peut être remplacé par $\kappa_n(\mathbf{z}) + B_n^2(\mathbf{z}) + p_{d,n} \sum_{i=1}^{L_{n-1}} w_{n-1}^i p_{s,n}(\mathbf{x}_{n-1}^i)p(\mathbf{z}|\mathbf{x}_{n-1}^i)$.

Calcul de l'estimateur CMC $\tilde{\Theta}_n$ en pratique

Nous avons besoin, dans le contexte multi-objets, d'un estimateur du nombre d'objets et des états cachés de chaque objet. D'après (1.2.9), un estimateur du nombre d'objets est donné par

$$\tilde{N}_n = \sum_{i=1}^{L_{n-1}} w_n^{1,i} + \sum_{\mathbf{z} \in \mathcal{Z}_n} \sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z}) + \sum_{i=1}^{L_{\gamma_n}} w_n^{2,i} + \sum_{\mathbf{z} \in \mathcal{Z}_n} \sum_{i=1}^{L_{\gamma_n}} w_n^{4,i}(\mathbf{z}). \quad (3.3.84)$$

La procédure pour extraire les états des objets persistants consiste à rechercher les maxima locaux de $\sum_{i=1}^{L_{n-1}} w_n^{1,i}p(\mathbf{x}_n|\mathbf{x}_{n-1}^i) + \sum_{\mathbf{z} \in \mathcal{Z}_n} \sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z})p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{z})$. Pour les objets naissants, cette procédure ne peut pas être utilisée si le PHD dû aux objets naissants a été calculé à partir d'une approximation MC. Il est possible d'utiliser des techniques de clustering [122] ou d'utiliser la procédure décrite dans [100], qui consiste à rechercher les mesures \mathbf{z} telles que le terme $\sum_{i=1}^{L_{\gamma_n}} w_n^{4,i}(\mathbf{z})$ soit au dessus d'un seuil donné (on choisira typiquement 0.5) ; un estimateur de l'état caché associé à \mathbf{z} est donné par $\sum_{i=1}^{L_{\gamma_n}} w_n^{4,i}(\mathbf{z})\mathbf{x}_{\gamma_n}^i$. Cependant, les composantes dues aux objets naissants deviennent des composantes associées à des objets persistants à l'itération suivante ; leur extraction devient alors plus facile puisqu'un algorithme de clustering peut être évité.

Remarque 3.12. La procédure d'extraction décrite ci-dessus [100] peut être adaptée pour l'extraction des objets persistants en recherchant les mesures \mathbf{z} telles que $\sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z})$ est supérieur à un seuil donné, et en estimant l'état associé par $\int \mathbf{x}_n p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{z}) d\mathbf{x}_n$. L'avantage de cette procédure est que le calcul de $\int \mathbf{x}_n p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{z}) d\mathbf{x}_n$ n'a besoin d'être fait que pour de telles mesures \mathbf{z} .

Détaillons maintenant quelques applications du filtre CMC-PHD. Nous supposons dans un premier temps que $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n)$, $g_n(\mathbf{z} | \mathbf{x}_n) = \mathcal{N}(\mathbf{z}; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n)$ et γ_n est un mélange de gaussiennes, $\gamma_n(\mathbf{x}_n) = \sum_{i=1}^{N_{\gamma_n}} w_{\gamma_n}^i \mathcal{N}(\mathbf{x}_n; \mathbf{m}_{\gamma_n}^i; \mathbf{P}_{\gamma_n}^i)$. Nous rappelons que pour ce modèle, une implémentation par mélange de gaussiennes a été proposée [121], et repose sur des techniques de fusion et d'élagage. Cette implémentation repose sur le fait que $p_{d,n}$ et $p_{s,n}$ sont constantes (ou éventuellement un mélange Gaussien [121]). Dans notre algorithme, nous n'avons pas besoin d'hypothèses portant sur $p_s(\mathbf{x}_{n-1})$. Dans ce modèle $B_n^2(\mathbf{z})$ est directement calculable et la procédure CMC pour estimer le nombre d'objets et pour extraire les états cachés peut être utilisée, puisque $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{z})$ et $p(\mathbf{z} | \mathbf{x}_{n-1})$ sont calculables (voir (3.3.16)-(3.3.17) et (3.3.21)-(3.3.22)). En résumé, dans le cas où $p_s(\mathbf{x}_{n-1})$ est constant, nous avons à notre disposition trois implémentations du filtre PHD : l'implémentation par mélange de gaussiennes [121], l'implémentation SMC [122] et nos implémentations CMC, qui seront comparées par simulations ci-dessous.

À présent, si γ_n n'est pas un mélange de gaussiennes, l'implémentation par mélange de gaussiennes du PHD ne peut plus être utilisée. Néanmoins, notre méthode est adaptable si nous calculons $\Theta_{2,n}$, $\Theta_{4,n}$ et $B_n^2(\mathbf{z})$ par une approximation MC. Contrairement à une méthode purement MC, notre implémentation CMC permet de garder la structure de mélange de gaussiennes pour les objets persistants.

Enfin, dans des modèles non linéaires l'implémentation par mélange de gaussiennes ne peut plus être utilisée directement. Le filtre PHD étendu (resp. UT) [121] approxime alors le PHD par un mélange de gaussiennes dont les paramètres sont calculées par un EKF (resp. UKF). Nous proposons alors d'adapter notre implémentation CMC en approchant $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{z})$ et $p(\mathbf{z} | \mathbf{x}_{n-1})$ par une des techniques décrites dans le Paragraphe 3.3.1 La différence principale est que nous partons d'une approximation MC du PHD à l'instant $n - 1$ et que nous calculons un estimateur des états cachés sans utiliser les techniques de clustering de l'implémentation MC. De cette manière, nous obtenons une approximation du PHD qui ne repose pas sur une approximation numérique à l'instant $n - 1$ et qui permet d'extraire les états cachés facilement. De plus, contrairement aux implémentations EKF/UKF du filtre PHD, les approximations numériques ne sont pas propagées dans le temps car elles ne sont utilisées que localement pour extraire les états.

Simulations

Nous comparons à présent notre estimateur CMC pour le PHD aux différents estimateurs classiques. Nous utiliserons la distance OSPA dont l'expression a été définie en (1.2.13) pour mesurer les différences entre les estimateurs, avec $\bar{p} = 2$ et $\bar{c} = 100$.

Nous nous focalisons sur le modèle linéaire et gaussien, pour lequel l'implémentation par mélange de gaussiennes du filtre PHD est utilisée comme filtre de référence, ce qui nous permettra d'apprécier les performances de l'estimateur CMC. Nous comparons l'implémentation par mélange de gaussiennes avec les implémentations SMC et CMC du filtre PHD. Nous estimons les positions et vitesses d'un nombre inconnu d'objets, $\mathbf{x}_n = [p_{x,n}, \dot{p}_{x,n}, p_{y,n}, \dot{p}_{y,n}]^T$ et nous avons $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}, \mathbf{Q}_n)$,

$g_n(\mathbf{z}_n | \mathbf{x}_n) = \mathcal{N}(\mathbf{z}_n; \mathbf{H}_n \mathbf{x}_n, \mathbf{R}_n)$, où $\mathbf{F}_n = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}$, et les autres paramètres (\mathbf{H}_n , \mathbf{Q}_n et \mathbf{R}_n)

sont identiques à ceux des simulations effectuées dans le Paragraphe 3.3.2.

Nous comparons les implémentations CMC et SMC dans le cas où les deux algorithmes utilisent la loi $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$ pour tirer les nouvelles particules (rappelons que dans notre approche, nous avons besoin de propager une approximation MC du PHD, même si elle n'est pas utilisée directement pour estimer le nombre d'objets). Nous prenons $T_s = 2$, $\sigma_v = 3$ mais $\sigma_x = \sigma_y = 0.3$: ainsi, la fonction de vraisemblance $g_n(\mathbf{z}|\mathbf{x}_n)$ est étroite ; puisque la loi de transition $f_{n|n-1}$ ne prend pas en compte les mesures, il est alors difficile de guider les particules dans des régions prometteuses. Par conséquent ce scénario expérimental représente un challenge pour l'implémentation SMC. Les particules sont initialisées autour des mesures disponibles à un instant donné, comme cela est proposé dans [100]. Pour chaque algorithme, nous utilisons $N_b = 20$ particules pour chaque objet naissant et $N = 200$ particules pour chaque objet persistant. La probabilité de détection est $p_{d,n} = 0.95$ et la probabilité de survie est $p_{s,n} = 0.98$, pour tout $n \in [1, \dots, 100]$, nous générons une moyenne de 10 mesures de fausses alarmes. Nous considérons un scénario avec 6 objets qui apparaissent à $n = 0$, $n = 20$ ou $n = 50$. Dans l'implémentation par mélange de gaussiennes, nous prenons $T_p = 10^{-5}$ pour le seuil d'élagage, $T_m = 4m$ pour le seuil de fusion, et nous gardons au plus $N_{\max} = 100$ gaussiennes.

La distance OSPA et le nombre d'objets estimé sont tracés sur les Figures 3.8 et 3.9(a). L'approche CMC est préférable à l'approche purement MC et règle le problème du guidage des particules dans des régions prometteuses. Même si nous utilisons la loi $f_{n|n-1}$ pour propager les particules et obtenir une approximation MC de v_{n-1} , l'approche CMC fournit un estimateur correct du nombre d'objets, contrairement à l'approche MC dans laquelle le nouveau jeu de points $\{\mathbf{x}_n^i, w_n^i\}_{i=1}^{L_n}$ est utilisé pour déduire une approximation MC de v_n , puis un estimateur du nombre d'objets. Enfin, le nombre d'objets est correctement estimé par les implémentations GM et CMC, mais l'estimateur CMC semble légèrement plus précis comme le montre la Figure 3.9(b).

3.4 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur les techniques de Rao-Blackwellisation qui permettent de réduire la variance des estimateurs MC dans le contexte des méthodes SMC. Nous avons dans un premier temps rappelé que l'égalité générale

$$\text{var}(\text{E}(f(X_2)|X_1)) = \text{var}(f(X_2)) - \text{E}(\text{var}(f(X_2)|X_1)) \quad (3.4.1)$$

avait été utilisée, dans le contexte du filtrage bayésien, pour des décompositions spatiales du vecteur d'état \mathbf{X}_n .

Bien que les équations du PHD qui, dans le contexte multi-objets, permettent d'estimer le nombre d'objets ainsi que leurs états n'impliquent pas nécessairement des densités de probabilité, nous avons montré que les estimateurs déduits des approximations du PHD pouvaient également bénéficier des techniques de marginalisation spatiale utilisées habituellement en filtrage mono-objet. Sous certaines conditions qui dépendent du problème étudié, ces techniques de marginalisation ont été couplées avec des méthodes auxiliaires visant à guider les particules dans des zones de l'espace d'état où les objets sont présents, ce qui est d'autant plus important ici que le nombre d'objets est inconnu. Nous avons ensuite validé nos algorithmes par simulations dans des scénarios divers, en fonction de l'intensité des mesures de fausses alarmes et/ou du caractère manoeuvrant des objets suivis.

Dans la seconde partie de ce chapitre, nous avons appliqué le résultat (3.4.1) aux problèmes de filtrage mono- et multi-objet(s), mais en considérant cette fois une partition temporelle $X_1 = \mathbf{X}_{0:n-1}$ et $X_2 = \mathbf{X}_n$

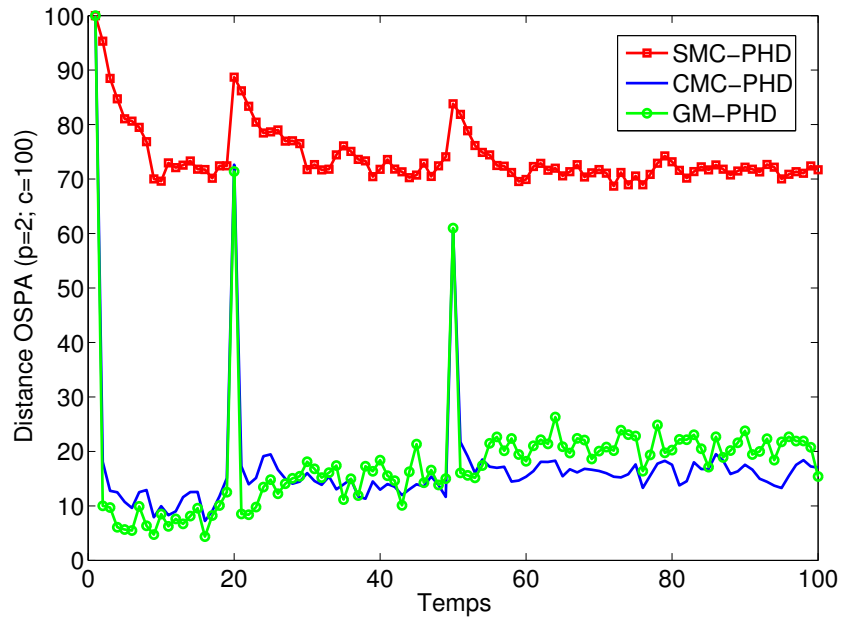
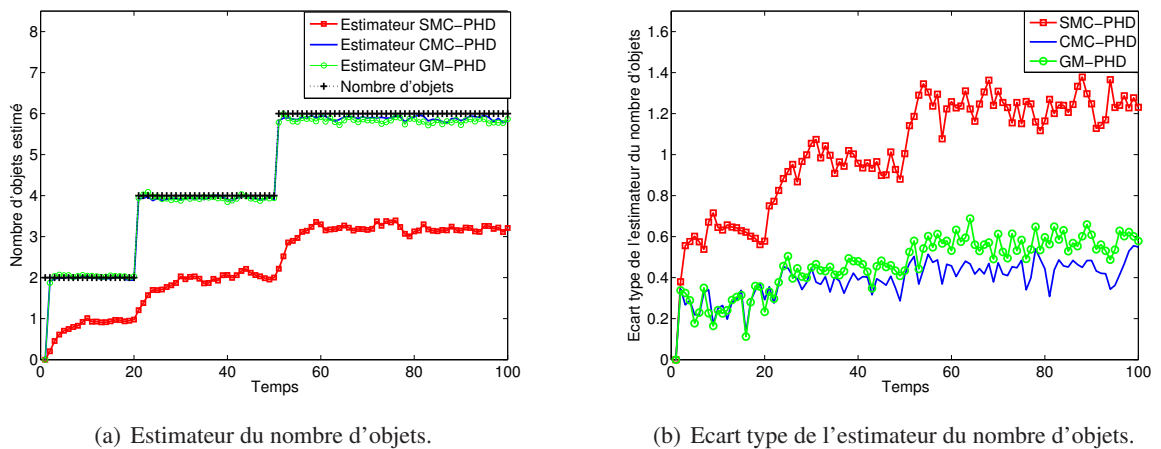


FIGURE 3.8 – Distance OSPA pour modèles linéaires et gaussiens - Les implémentations par mélange de gaussiennes et CMC sont largement préférables à l'implémentation purement MC à cause de la forme de la densité de vraisemblance $g_n(\mathbf{z}|\mathbf{x})$.



(a) Estimateur du nombre d'objets.

(b) Écart type de l'estimateur du nombre d'objets.

FIGURE 3.9 – Estimateur du nombre d'objets et son écart type ; le nombre d'objets estimé par l'implémentation CMC est correct et précis bien que l'approximation discrète sur laquelle il est repose soit mauvaise.

3.4 Conclusion

du vecteur d'état $\mathbf{X}_{0:n}$. Cette partition est en effet naturelle dans le contexte des méthodes SMC car nous disposons à chaque instant d'une approximation MC de $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$ ou de $v_{n-1}(\mathbf{x})$. Cette technique alternative (qui peut par ailleurs être combinée avec une technique de marginalisation spatiale) peut être implémentée exactement dans un certain nombre de modèles stochastiques. Dans le cas où le calcul exact de l'estimateur CMC n'est plus possible, nous avons proposé des implémentations qui sont soit basées sur des approximations numériques, soit, dans les cas les plus sévères, sur des techniques récentes de tirages MC locaux. Nous avons enfin validé nos algorithmes par simulations.

Modèles de Markov couple et triplet

Sommaire

4.1	Généralités sur les modèles de Markov couple et triplet	120
4.2	Filtrage multi-objets pour les chaînes de Markov couple	130
4.3	Une classe d’algorithmes de filtrage exact rapide pour des modèles de Markov à sauts	140
4.4	Conclusion	150

Dans les chapitres précédents, nous nous sommes penchés sur la question du filtrage mono- et multi-objet(s) dans des classes de modèles statistiques spécifiques, décrits par

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\mathbf{x}_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1}) \prod_{i=0}^n g_i(\mathbf{y}_i | \mathbf{x}_i) \quad (4.0.1)$$

pour les modèles HMC et par

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n}) = p(r_0) \prod_{i=1}^n p(r_i | r_{i-1}) p(\mathbf{x}_0 | r_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1}, r_i) \prod_{i=0}^n g_i(\mathbf{y}_i | \mathbf{x}_i, r_i) \quad (4.0.2)$$

pour les modèles à sauts dépendants d’un processus latent et discret $\{R_n\}_{n \geq 0}$. Le but de ce chapitre est de montrer que les algorithmes d’estimation de $\Theta_n = E(f(\mathbf{X}_n) | \mathbf{y}_{0:n})$ peuvent être étendus à des modèles statistiques plus généraux ; nous verrons également qu’il est possible d’exploiter ce degré de généralité supplémentaire pour faire émerger des approximations originales de l’estimateur bayésien optimal.

Tout d’abord, nous généralisons les modèles couramment utilisés : nous montrons que certaines des hypothèses intrinsèques à ces modèles peuvent être relâchées de façon à pouvoir prendre en compte des problèmes plus complexes. La généralisation de ces modèles classiques repose sur l’emploi des modèles de Markov couple et triplet, qui ne supposent que la markovianité du couple $\{\mathbf{X}_n, \mathbf{Y}_n\}_{n \geq 0}$ ou du triplet $\{\mathbf{X}_n, \mathbf{Y}_n, R_n\}_{n \geq 0}$. Dans ce chapitre, une réalisation du couple $(\mathbf{X}_n, \mathbf{Y}_n)$ sera notée $(\mathbf{x}_n, \mathbf{y}_n) = \boldsymbol{\xi}_n$. Ensuite, nous rappelons qu’il est possible d’étendre les algorithmes de calcul existants pour le problème d’estimation de Θ_n , c’est à dire le FK et le FP.

La deuxième section de ce chapitre est consacrée au problème de filtrage multi-objets pour les modèles de Markov couple. Dans une troisième section, nous montrons que des solutions alternatives au problème de l’estimation de Θ_n peuvent émerger en considérant les modèles triplet, en particulier lorsque l’on considère le problème d’estimation dans les modèles à sauts linéaires et gaussiens,

$$f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n(r_n) \mathbf{x}_{n-1}; \mathbf{Q}_n(r_n)), \quad (4.0.3)$$

$$g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n(r_n) \mathbf{x}_n; \mathbf{R}_n(r_n)). \quad (4.0.4)$$

Brièvement, au lieu d'approcher l'estimateur optimal dans le modèle JMSS vérifiant (4.0.3)-(4.0.4), nous approchons ce modèle originel par un modèle qui conserve certaines de ses propriétés physiques, et dans lequel le calcul exact de Θ_n est faisable avec une complexité linéaire en n , n étant le nombre d'observations. La technique de filtrage obtenue permet en plus de traiter des modèles plus généraux que le JMSS linéaire et gaussien.

Ce chapitre est donc découpé de la manière suivante.

- La Section 4.1 s'intéresse aux limitations statistiques des modèles HMC (4.0.1) ainsi que des modèles à sauts (4.0.2). Afin de contourner les limitations de ces modèles, les modèles de Markov couple et triplet sont présentés. Nous discutons des méthodes de filtrage mono-objet qu'il est possible d'utiliser dans ces modèles ;
- la Section 4.2 est consacrée au filtrage multi-objets lorsque chaque état caché et l'observation qui lui est associée suivent un modèle couple. Nous verrons que ce modèle statistique suggère une reformulation du problème de filtrage multi-objets, de laquelle sera déduite le filtre PHD pour les modèles couple ainsi qu'une implémentation par mélange de gaussiennes de ce filtre PHD ;
- enfin, dans la Section 4.3, nous exploitons la structure particulière de certains modèles triplet, dans lesquels le calcul exact de l'estimateur Θ_n est faisable à coût linéaire en nombre d'observations, même si ces modèles dépendent de v.a. discrètes. Nous utilisons ces modèles triplet pour proposer une classe d'algorithmes rapides de filtrage exact pour des modèles à sauts dans lesquels les propriétés physiques sont identiques à celles des modèles JMSS (4.0.3)-(4.0.4). Cette classe d'algorithmes peut être utilisée pour l'approximation de Θ_n dans les modèles à sauts (4.0.3)-(4.0.4) et constitue une alternative au FP et aux algorithmes IMM.

4.1 Généralités sur les modèles de Markov couple et triplet

Dans cette première section, nous nous intéressons aux propriétés statistiques des modèles HMC (4.0.1) et JMSS (4.0.2) et nous montrons que ces propriétés peuvent être généralisées par les modèles de Markov couple et triplet. Nous discutons ensuite du filtrage exact et approché dans ces modèles, puis nous illustrons leur intérêt sur divers exemples.

4.1.1 Les limitations des modèles HMC

Nous considérons le modèle HMC (4.0.1) décrit par les lois $p(\mathbf{x}_0)$, $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $g_n(\mathbf{y}_n|\mathbf{x}_n)$. Tout au long des Chapitres 2 et 3, nous avons utilisé ces modèles pour deux raisons. Premièrement, leur structure statistique permet de calculer ou d'approcher des estimateurs d'intérêt Θ_n de façon récursive. Deuxièmement, un vaste champ d'applications peut être considéré à travers une modélisation statistique impliquant ces modèles. En effet, comme nous l'avons montré dans nos diverses simulations, cette modélisation passe par le choix des fonctions $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $g_n(\mathbf{y}_n|\mathbf{x}_n)$ qui modélisent, d'un point de vue probabiliste, la façon dont le processus caché évolue dans le temps ainsi que la relation entre une observation et l'état caché. Cependant, même si ces modèles sont d'intérêt pratique, la forme de la loi jointe $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ donnée par la factorisation (4.0.1) est équivalente aux deux propriétés suivantes, qui sont généralement utilisées pour définir le modèle HMC :

- D.1** $\{\mathbf{X}_n\}_{n \geq 0}$ est une chaîne de Markov dont les transitions sont données par $\{f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})\}_{n \geq 1}$;
- D.2** Conditionnellement à une réalisation $\mathbf{X}_{0:n} = \mathbf{x}_{0:n}$, les observations $\mathbf{Y}_{0:n}$ sont indépendantes et l'observation \mathbf{Y}_i ne dépend que de \mathbf{x}_i .

Notons que le jeu de propriétés ci-dessus décrit dans un premier temps les propriétés du processus $\{\mathbf{X}_n\}_{n \geq 0}$, puis celles de $\{\mathbf{Y}_n\}_{n \geq 0}$, conditionnellement à $\mathbf{X}_{0:n} = \mathbf{x}_{0:n}$. De manière équivalente, il est possible de définir un HMC à partir d'un jeu de propriétés qui se focalise directement sur le processus $\{\mathbf{X}_n, \mathbf{Y}_n\}$. En effet, le jeu de propriétés suivant est lui aussi équivalent à la factorisation (4.0.1) :

P.1 Conditionnellement à $\mathbf{X}_{n-1} = \mathbf{x}_{n-1}$ et $\mathbf{Y}_{n-1} = \mathbf{y}_{n-1}$, \mathbf{X}_n ne dépend que de \mathbf{x}_{n-1} , $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) = f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$ (notons que **P.1** est une conséquence directe de **D.2**) ;

P.2 Conditionnellement à $\mathbf{X}_n = \mathbf{x}_n$, $\mathbf{X}_{n-1} = \mathbf{x}_{n-1}$ et $\mathbf{Y}_{n-1} = \mathbf{y}_{n-1}$, \mathbf{Y}_n ne dépend que de \mathbf{x}_n , $p(\mathbf{y}_n | \mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) = g_n(\mathbf{y}_n | \mathbf{x}_n)$;

P.3 Le couple $(\mathbf{X}_n, \mathbf{Y}_n)$ est une chaîne de Markov, $p(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1}) = p(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$. Remarquons que dans les modèles HMC, l'observation \mathbf{Y}_{n-1} n'affecte pas la loi du couple $(\mathbf{X}_n, \mathbf{Y}_n)$, conditionnelle à $\mathbf{X}_{n-1} = \mathbf{x}_{n-1}$; en d'autres termes, $p(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) = p(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1})$.

Les modèles de Markov couple ne vérifient pas nécessairement les propriétés **P.1** et **P.2** comme nous le voyons à présent.

4.1.2 Modèles de Markov couple

Un modèle de Markov couple est un modèle statistique qui vérifie la propriété **P.3** [94] [36]. En d'autres termes, le processus $\{(\mathbf{X}_n, \mathbf{Y}_n)\}_{n \geq 0}$ est un modèle de Markov couple (ou PMC) si et seulement si la loi jointe $p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\boldsymbol{\xi}_{0:n})$ de $(\mathbf{X}_{0:n}, \mathbf{Y}_{0:n})$ se factorise de la manière suivante :

$$\underbrace{p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})}_{p(\boldsymbol{\xi}_{0:n})} = \underbrace{p(\mathbf{x}_0, \mathbf{y}_0)}_{p(\boldsymbol{\xi}_0)} \prod_{i=1}^n \underbrace{p_{i|i-1}(\mathbf{x}_i, \mathbf{y}_i | \mathbf{x}_{i-1}, \mathbf{y}_{i-1})}_{p_{i|i-1}(\boldsymbol{\xi}_i | \boldsymbol{\xi}_{i-1})}. \quad (4.1.1)$$

Le modèle HMC est un modèle PMC dans lequel, de plus, les propriétés **P.1** et **P.2** sont valides. De plus, dans ce cas particulier les lois $p(\mathbf{y}_0 | \mathbf{x}_0)$ et les transitions $\{p_{i|i-1}(\boldsymbol{\xi}_i | \boldsymbol{\xi}_{i-1})\}_{i \geq 1}$ s'écrivent

$$p(\mathbf{y}_0 | \mathbf{x}_0) \stackrel{\text{HMC}}{=} g_0(\mathbf{y}_0 | \mathbf{x}_0), \quad (4.1.2)$$

$$p_{i|i-1}(\mathbf{x}_i, \mathbf{y}_i | \mathbf{x}_{i-1}, \mathbf{y}_{i-1}) \stackrel{\text{HMC}}{=} f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1}) g_i(\mathbf{y}_i | \mathbf{x}_i). \quad (4.1.3)$$

Notons alors que dans un modèle PMC, $\{\mathbf{X}_n\}_{n \geq 0}$ n'est pas nécessairement une chaîne de Markov : la propriété **D.1**, qui est une propriété supposée vérifiée dans la plupart des algorithmes de filtrage, ne l'est donc plus nécessairement ici [73]. De plus, pour $0 \leq i \leq n$, conditionnellement à $(\mathbf{x}_{0:n}, \mathbf{y}_{0:i-1})$, la loi de \mathbf{Y}_i peut dépendre, en plus de \mathbf{x}_i , de \mathbf{x}_{i-1} , \mathbf{y}_{i-1} et $\mathbf{x}_{i+1:n}$ [39]. En d'autres termes, la loi $p(\mathbf{y}_{0:n} | \mathbf{x}_{0:n})$ s'écrit en général

$$p(\mathbf{y}_{0:n} | \mathbf{x}_{0:n}) = p(\mathbf{y}_0 | \mathbf{x}_{0:n}) \prod_{i=1}^n p(\mathbf{y}_i | \mathbf{x}_{i-1:n}, \mathbf{y}_{i-1}) ; \quad (4.1.4)$$

la propriété **D.2** n'est donc pas nécessairement vérifiée. Par conséquent, des scénarios plus complexes peuvent être considérés à travers les modèles PMC, comme nous le verrons au Paragraphe 4.1.4.

La généralisation apportée par les modèles PMC peut être appréciée à travers l'exemple simple suivant. Plaçons nous dans le modèle linéaire et gaussien

$$\mathbf{X}_n = \mathbf{F}_n \mathbf{X}_{n-1} + \mathbf{U}_n, \quad (4.1.5)$$

$$\mathbf{Y}_n = \mathbf{H}_n \mathbf{X}_n + \mathbf{V}_n, \quad (4.1.6)$$

où \mathbf{F}_n est une matrice de taille $p \times p$, \mathbf{H}_n une matrice de taille $q \times p$, $\mathbf{X}_0, \mathbf{U}_1, \dots, \mathbf{U}_n, \mathbf{V}_0, \dots, \mathbf{V}_n$ sont indépendants, $\mathbf{X}_0 \sim \mathcal{N}(\cdot; \mathbf{m}_0; \mathbf{P}_0)$, $\mathbf{U}_n \sim \mathcal{N}(\cdot; \mathbf{0}; \mathbf{Q}_n)$ et $\mathbf{V}_n \sim \mathcal{N}(\cdot; \mathbf{0}; \mathbf{R}_n)$. Nous supposons que \mathbf{Q}_n et \mathbf{R}_n sont définies positives pour tout n . Les hypothèses sur les bruits $\{\mathbf{U}_n\}_{n \geq 1}$ et $\{\mathbf{V}_n\}_{n \geq 0}$ impliquent que $(\mathbf{X}_n, \mathbf{Y}_n)$ est un HMC dont la transition et la vraisemblance sont données respectivement par

$$f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n), \quad (4.1.7)$$

$$g_n(\mathbf{y}_n | \mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n). \quad (4.1.8)$$

Le modèle (4.1.5)-(4.1.6) peut se réécrire sous une forme matricielle : en injectant (4.1.5) dans (4.1.6), puis en posant $\mathbf{W}_n = [\mathbf{U}_n^T, (\mathbf{H}_n \mathbf{U}_n + \mathbf{V}_n)^T]^T$, le modèle (4.1.5)-(4.1.6) s'écrit

$$\begin{bmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{bmatrix} = \begin{bmatrix} \mathbf{F}_n & \mathbf{0} \\ \mathbf{H}_n \mathbf{F}_n & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{n-1} \\ \mathbf{Y}_{n-1} \end{bmatrix} + \mathbf{W}_n, \quad (4.1.9)$$

où les bruits gaussiens $\mathbf{W}_1, \dots, \mathbf{W}_n$ sont indépendants, de moyennes nulles, indépendants de $(\mathbf{X}_0, \mathbf{Y}_0)$ et vérifient

$$\mathbb{E}(\mathbf{W}_n \mathbf{W}_n^T) = \begin{bmatrix} \mathbf{Q}_n & (\mathbf{H}_n \mathbf{Q}_n)^T \\ \mathbf{H}_n \mathbf{Q}_n & \mathbf{R}_n + \mathbf{H}_n \mathbf{Q}_n \mathbf{H}_n^T \end{bmatrix}. \quad (4.1.10)$$

Le modèle HMC linéaire et gaussien classiquement utilisé apparait donc comme un cas particulier du modèle

$$\begin{bmatrix} \mathbf{X}_n \\ \mathbf{Y}_n \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{F}_n^1 & \mathbf{F}_n^2 \\ \mathbf{H}_n^1 & \mathbf{H}_n^2 \end{bmatrix}}_{\mathbf{B}_n} \begin{bmatrix} \mathbf{X}_{n-1} \\ \mathbf{Y}_{n-1} \end{bmatrix} + \mathbf{W}_n, \quad (4.1.11)$$

dans lequel les bruits $\mathbf{W}_1, \dots, \mathbf{W}_n$ sont des bruits gaussiens centrés, indépendants et indépendants de $(\mathbf{X}_0, \mathbf{Y}_0) \sim \mathcal{N}(\cdot; \tilde{\mathbf{m}}_0; \tilde{\mathbf{P}}_0)$, avec

$$\mathbb{E}(\mathbf{W}_n \mathbf{W}_n^T) = \Sigma_n = \begin{bmatrix} \Sigma_n^{11} & \Sigma_n^{21^T} \\ \Sigma_n^{21} & \Sigma_n^{22} \end{bmatrix}. \quad (4.1.12)$$

Les hypothèses d'indépendance sur les bruits \mathbf{W}_n impliquent que le modèle (4.1.11) est un modèle PMC dont la loi initiale et les transitions sont données par

$$p(\boldsymbol{\xi}_0) = \mathcal{N}(\boldsymbol{\xi}_0; \tilde{\mathbf{m}}_0; \tilde{\mathbf{P}}_0), \quad (4.1.13)$$

$$p_{n|n-1}(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}) = \mathcal{N}(\boldsymbol{\xi}_n; \mathbf{B}_n \boldsymbol{\xi}_{n-1}; \Sigma_n). \quad (4.1.14)$$

Ces modèles gaussiens seront particulièrement utilisés aux Paragraphes 4.1.4 et 4.1.5.

4.1.3 Filtrage dans les modèles de Markov couple

Bien que les modèles de Markov couple vérifient des propriétés statistiques plus générales que les modèles HMC (rappelons que **D.1**, **D.2** ainsi que **P.1** et **P.2** ne sont pas nécessairement satisfaites dans ces modèles), le calcul récursif de la loi de filtrage $p(\mathbf{x}_n | \mathbf{y}_{0:n})$ reste possible en utilisant la formule

$$p(\mathbf{x}_n | \mathbf{y}_{0:n}) = \frac{\int p_{n|n-1}(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}) p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}}{\int p_{n|n-1}(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}) p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1:n}}, \quad (4.1.15)$$

qui coïncide avec le filtre bayésien classique

$$p(\mathbf{x}_n | \mathbf{y}_{0:n}) \propto g_n(\mathbf{y}_n | \mathbf{x}_n) \int f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1} \quad (4.1.16)$$

lorsque les égalités (4.1.2)-(4.1.3) sont satisfaites. Notons que dans l'équation (4.1.15), le nouvel état \mathbf{x}_n n'est plus introduit par la loi $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$ mais par $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$, qui dépend également de \mathbf{y}_{n-1} ; de même, la mise à jour du filtre à partir de la nouvelle observation \mathbf{y}_n est faite par la loi $p(\mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{y}_{n-1})$. Par conséquent, contrairement au filtre bayésien pour les modèles HMC, le calcul de la loi de filtrage ne peut pas être décomposé simplement en deux parties distinctes (prédiction + mise à jour), excepté dans le cas où la loi $p(\mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{y}_{n-1})$ ne dépend pas de \mathbf{x}_{n-1} .

Le calcul exact du filtre bayésien (4.1.15) pour les modèles de Markov couple n'est généralement pas faisable. Pour les modèles couple linéaires et gaussiens vérifiant (4.1.13)-(4.1.14), $p(\mathbf{x}_n | \mathbf{y}_{0:n})$ est une gaussienne dont les moments sont calculables récursivement par le FK pour PMC [77] [96] et qui correspond à l'Algorithme 4.1.

Algorithme 4.1: FK pour les modèles PMC linéaires et gaussiens

Soit un modèle couple linéaire et gaussien vérifiant (4.1.13)- (4.1.14) et

$$p(\mathbf{x}_{n-1} | \mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_{n-1}; \mathbf{m}_{n-1|n-1}; \mathbf{P}_{n-1|n-1});$$

Poser

$$\mathbf{m}_{n|n-1} = \mathbf{F}_n^1 \mathbf{m}_{n-1|n-1} + \mathbf{F}_n^2 \mathbf{y}_{n-1}, \quad (4.1.17)$$

$$\mathbf{P}_{n|n-1} = \Sigma_n^{11} + \mathbf{F}_n^1 \mathbf{P}_{n-1|n-1} (\mathbf{F}_n^1)^T, \quad (4.1.18)$$

$$\mathbf{m}_{n|n} = \mathbf{m}_{n|n-1} + \mathbf{K}_n (\mathbf{y}_n - \tilde{\mathbf{y}}_n), \quad (4.1.19)$$

$$\mathbf{P}_{n|n} = \mathbf{P}_{n|n-1} - \mathbf{K}_n (\Sigma_n^{21} + \mathbf{H}_n^1 \mathbf{P}_{n-1|n-1} (\mathbf{F}_n^1)^T), \quad (4.1.20)$$

$$\tilde{\mathbf{y}}_n = \mathbf{H}_n^1 \mathbf{m}_{n-1|n-1} - \mathbf{H}_n^2 \mathbf{y}_{n-1}, \quad (4.1.21)$$

$$\mathbf{K}_n = (\Sigma_n^{11} + \mathbf{F}_n^1 \mathbf{P}_{n-1|n-1} (\mathbf{H}_n^1)^T) \mathbf{L}_n^{-1}, \quad (4.1.22)$$

$$\mathbf{L}_n = \mathbf{H}_n^1 \mathbf{P}_{n-1|n-1} \mathbf{H}_n^1 + (\Sigma_n^{22})^T. \quad (4.1.23)$$

Alors $p(\mathbf{x}_n | \mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_n; \mathbf{m}_{n|n}; \mathbf{P}_{n|n})$.

Pour les modèles plus généraux, le FP peut être utilisé [39] et correspond à l'Algorithme 4.2. Observons que dans l'Algorithme 4.2, la densité d'importance conditionnelle qui minimise la variance conditionnelle des poids w_n^i est donnée par

$$q(\mathbf{x}_n | \mathbf{x}_{0:n-1}) = p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{y}_n), \quad (4.1.24)$$

auquel cas les poids s'écrivent

$$w_n^i \propto w_{n-1}^i p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_{n-1}), \quad \sum_{i=1}^N w_n^i = 1. \quad (4.1.25)$$

4.1.4 De l'importance des modèles couple

Nous présentons ici une liste non exhaustive de modèles PMC dans lesquels les algorithmes de filtrage décrits ci-dessus peuvent être utilisés pour calculer ou estimer un moment Θ_n [13] [62] [109].

Algorithme 4.2: Algorithme de FP pour les modèles PMC

Soit $\{w_{n-1}^i, \mathbf{x}_{0:n-1}^i\}_{i=1}^N$ une approximation de $p(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1})$.

pour $i = 1 \leq i \leq N$ **faire**

- P.** Tirer $\mathbf{x}_n^i \sim q(\mathbf{x}_n|\mathbf{x}_{0:n-1}^i)$;
- W.** Calculer $w_n^i \propto w_{n-1}^i p(\mathbf{x}_n^i, \mathbf{y}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_{n-1})/q(\mathbf{x}_n^i|\mathbf{x}_{0:n-1}^i)$, $\sum_{i=1}^N w_n^i = 1$;

fin

Alors $\hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_{0:n}^i}$ approxime $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$.

si Rééchantillonnage **alors**

- pour** $i = 1 \leq i \leq N'$ **faire**
- R.** $\tilde{\mathbf{x}}_{0:n}^i \sim \sum_{j=1}^N w_n^j \delta_{\mathbf{x}_{0:n}^j}$, $\tilde{w}_n^i = 1/N'$;

fin

Poser $\{\mathbf{x}_{0:n}^i, w_n^i\}_{i=1}^{N'} = \{\tilde{\mathbf{x}}_{0:n}^i, \tilde{w}_n^i\}_{i=1}^{N'}$; $\hat{p}(\mathbf{x}_{0:n}|\mathbf{y}_{0:n}) = \frac{1}{N'} \sum_{i=1}^{N'} \delta_{\mathbf{x}_{0:n}^i}$ approxime $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$

fin

Exemple 1 - Bruits de processus et d'observation corrélés

Dans le modèle linéaire et gaussien (4.1.5)-(4.1.6), nous avons supposé que les bruits de processus \mathbf{U}_n et d'observation \mathbf{V}_n étaient indépendants entre eux. Supposons maintenant que \mathbf{U}_n et \mathbf{V}_n sont dépendants, avec $E(\mathbf{U}_n \mathbf{V}_n^T) = \mathbf{S}_n \neq \mathbf{0}$. Une telle corrélation peut apparaître par exemple durant l'étape de discrétisation d'un modèle continu [109]. Ce modèle linéaire et gaussien dans lequel la matrice \mathbf{B}_n est identique à celle du modèle classique (4.1.9), se différencie par la forme de la matrice $\Sigma_n = E(\mathbf{W}_n \mathbf{W}_n^T)$ qui s'écrit à présent

$$\Sigma_n = \begin{bmatrix} \mathbf{Q}_n & (\mathbf{H}_n \mathbf{Q}_n + \mathbf{S}_n^T)^T \\ \mathbf{H}_n \mathbf{Q}_n + \mathbf{S}_n^T & \mathbf{R}_n + \mathbf{H}_n \mathbf{Q}_n \mathbf{H}_n^T + \mathbf{H}_n \mathbf{S}_n + \mathbf{S}_n^T \mathbf{H}_n^T \end{bmatrix}.$$

En utilisant le Lemme A.1 en Annexe A, la loi $p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$ dépend de \mathbf{x}_n et de \mathbf{x}_{n-1} de par sa moyenne $\mathbf{m}^{\mathbf{y}_n}(\mathbf{x}_{n-1}, \mathbf{x}_n)$, qui s'écrit

$$\mathbf{m}^{\mathbf{y}_n}(\mathbf{x}_{n-1}, \mathbf{x}_n) = \mathbf{H}_n \mathbf{F}_n \mathbf{x}_{n-1} + (\mathbf{H}_n \mathbf{Q}_n + \mathbf{S}_n^T) \mathbf{Q}_n^{-1} (\mathbf{x}_n - \mathbf{F}_n \mathbf{x}_{n-1}).$$

$\mathbf{m}^{\mathbf{y}_n}(\mathbf{x}_{n-1}, \mathbf{x}_n)$ coïncide évidemment avec $\mathbf{H}_n \mathbf{x}_n$ si $\mathbf{S}_n = \mathbf{0}$.

Exemple 2 - Bruit d'observation coloré

Un autre exemple concerne le cas où les hypothèses d'indépendance mutuelle des bruits \mathbf{V}_n dans (4.1.6) ne sont plus vérifiées. Nous supposons à présent que $\{\mathbf{V}_n\}_{n \geq 0}$ est une chaîne de Markov (le bruit \mathbf{V}_n est coloré),

$$\mathbf{V}_n = \tilde{\mathbf{H}}_n^T \mathbf{V}_{n-1} + \tilde{\mathbf{V}}_n,$$

où $\tilde{\mathbf{V}}_n \sim \mathcal{N}(\cdot; \mathbf{0}; \tilde{\mathbf{R}}_n)$ est une suite de bruits mutuellement indépendants, indépendants de $\mathbf{U}_1, \dots, \mathbf{U}_n$ et de \mathbf{X}_0 . En utilisant (4.1.5) et $\mathbf{V}_{n-1} = \mathbf{Y}_{n-1} - \mathbf{H}_{n-1} \mathbf{X}_{n-1}$, (4.1.6) devient

$$\begin{aligned} \mathbf{Y}_n &= \mathbf{H}_n \mathbf{X}_n + \tilde{\mathbf{H}}_n^T \mathbf{V}_{n-1} + \tilde{\mathbf{V}}_n \\ &= (\mathbf{H}_n \mathbf{F}_n - \tilde{\mathbf{H}}_n^T \mathbf{H}_{n-1}) \mathbf{X}_{n-1} + \tilde{\mathbf{H}}_n^T \mathbf{Y}_{n-1} + \mathbf{H}_n \mathbf{U}_n + \tilde{\mathbf{V}}_n. \end{aligned}$$

Ce modèle est donc un modèle PMC linéaire et gaussien dans lequel

$$\mathbf{B}_n = \begin{bmatrix} \mathbf{F}_n & \mathbf{0} \\ \mathbf{H}_n \mathbf{F}_n - \tilde{\mathbf{H}}_n \mathbf{H}_{n-1} & \tilde{\mathbf{H}}_n \end{bmatrix}, \quad (4.1.26)$$

$$\Sigma_n = \begin{bmatrix} \mathbf{Q}_n & (\mathbf{H}_n \mathbf{Q}_n)^T \\ \mathbf{H}_n \mathbf{Q}_n & \tilde{\mathbf{R}}_n + \mathbf{H}_n \mathbf{Q}_n \mathbf{H}_n^T \end{bmatrix}. \quad (4.1.27)$$

En effet, $p(\mathbf{y}_n | \mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$ dépend de \mathbf{x}_n , mais aussi de \mathbf{x}_{n-1} et \mathbf{y}_{n-1} , à travers sa moyenne $\mathbf{m}^{\mathbf{y}_n}(\mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{y}_{n-1})$ qui s'écrit ici $\mathbf{m}^{\mathbf{y}_n}(\mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{y}_{n-1}) = \mathbf{H}_n \mathbf{x}_n - \tilde{\mathbf{H}}_n \mathbf{H}_{n-1} \mathbf{x}_{n-1} + \tilde{\mathbf{H}}_n \mathbf{y}_{n-1}$.

Exemple 3 - Modèles couple non linéaires

Il est bien sûr possible de construire des modèles couple non linéaires, en adaptant les exemples précédents. Considérons par exemple le modèle

$$\mathbf{X}_n = f_n(\mathbf{X}_{n-1}) + \mathbf{U}_n, \quad (4.1.28)$$

$$\mathbf{Y}_n = h_n(\mathbf{X}_n) + \mathbf{V}_n, \quad (4.1.29)$$

où les bruits gaussiens centrés $\{\mathbf{U}_n\}_{n \geq 1}$, de variance $\{\mathbf{Q}_n\}_{n \geq 1}$ sont indépendants entre eux et indépendants de \mathbf{X}_0 . Considérons alors la corrélation suivante entre \mathbf{V}_n et \mathbf{V}_{n-1} ,

$$\mathbf{V}_n = \tilde{h}_n(\mathbf{V}_{n-1}) + \tilde{\mathbf{V}}_n, \quad (4.1.30)$$

où les bruits gaussiens centrés $\tilde{\mathbf{V}}_n$ sont indépendants entre eux, indépendants de \mathbf{X}_0 et $\text{var}(\tilde{\mathbf{V}}_n) = \tilde{\mathbf{R}}_n$. L'équation (4.1.29) peut être réécrite

$$\mathbf{Y}_n = h_n(\mathbf{X}_n) + \tilde{h}_n(\mathbf{Y}_{n-1} - h_{n-1}(\mathbf{X}_{n-1})) + \tilde{\mathbf{V}}_n; \quad (4.1.31)$$

On déduit alors que $(\mathbf{X}_n, \mathbf{Y}_n)$ est une chaîne de Markov couple dont la transition s'écrit

$$p(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) = p(\mathbf{x}_n | \mathbf{x}_{n-1}) p(\mathbf{y}_n | \mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) \quad (4.1.32)$$

$$= \mathcal{N}(\mathbf{x}_n, f_n(\mathbf{x}_{n-1}); \mathbf{Q}_n) \mathcal{N}(\mathbf{y}_n; h_n(\mathbf{x}_n) + \tilde{h}_n(\mathbf{y}_{n-1} - h_{n-1}(\mathbf{x}_{n-1})); \tilde{\mathbf{R}}_n) \quad (4.1.33)$$

Dans le contexte du FP, la loi d'importance conditionnelle optimale (4.1.24) est calculable si la fonction $h_n(\mathbf{x}_n)$ est linéaire et est alors déduite du Lemme A.2 en Annexe A.

4.1.5 Modèles de Markov couple physiquement contraints

Nous montrons dans ce paragraphe que les modèles couple peuvent être tout aussi pertinents, d'un point de vue physique, que les modèles HMC. Une des raisons pour lesquelles les modèles HMC sont utilisés par les praticiens est que les fonctions $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$ et $g_n(\mathbf{y}_n | \mathbf{x}_n)$ décrivent les propriétés physiques du modèle et sont donc choisies en fonction d'une problématique donnée. Néanmoins, rappelons que l'utilisation d'un modèle HMC implique que les données que l'on traite vérifient implicitement les propriétés statistiques **P.1-P.3**. Par ailleurs, les modèles PMC possèdent des propriétés statistiques plus générales mais le choix d'une transition $p_{i|i-1}(\xi_i | \xi_{i-1})$ en fonction d'un problème donné n'est pas nécessairement évident.

Notre objectif dans ce paragraphe est donc de proposer, dans le cas linéaire et gaussien, une classe de modèles couple dans lesquels les propriétés physiques sont identiques à celles d'un modèle HMC décrit par

(4.1.7)-(4.1.8) (ce résultat sera également utilisé dans la Section 4.3). Ici, pour différencier la jointe du couple $(\mathbf{X}_{0:n}, \mathbf{Y}_{0:n})$ des modèles HMC et PMC, nous utilisons la notation suivante : nous notons $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ la loi associée au modèle HMC ($p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ vérifie (4.0.1)) et $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ la loi associée au modèle PMC ($p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ vérifie (4.1.1)). Nous construisons donc une classe de modèles PMC qui n'est pas définie à partir des transitions du couple $p_{i|i-1}(\boldsymbol{\xi}|\boldsymbol{\xi}_{i-1})$ mais à partir des lois $p^2(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $p^2(\mathbf{y}_n|\mathbf{x}_n)$, qui coïncident respectivement avec les lois du modèle HMC $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $g_n(\mathbf{y}_n|\mathbf{x}_n)$, et telle que la transition $p_{n|n-1}^2(\boldsymbol{\xi}_n|\boldsymbol{\xi}_{n-1})$ ne dépende pas des paramètres de la loi initiale $p^2(\boldsymbol{\xi}_0)$.

Notons que dans un modèle PMC (4.1.1), il peut être difficile de déduire directement l'expression de ces lois à partir des transitions $p_{n|n-1}^2(\boldsymbol{\xi}_n|\boldsymbol{\xi}_{n-1})$,

$$p^2(\mathbf{x}_n|\mathbf{x}_{n-1}) = \frac{\int p^2(\boldsymbol{\xi}_0) \prod_{i=1}^n p_{i|i-1}^2(\boldsymbol{\xi}_i|\boldsymbol{\xi}_{i-1}) d\mathbf{x}_{0:n-2} d\mathbf{y}_{0:n}}{\int p^2(\boldsymbol{\xi}_0) \prod_{i=1}^{n-1} p_{i|i-1}^2(\boldsymbol{\xi}_i|\boldsymbol{\xi}_{i-1}) d\mathbf{x}_{0:n-2,n} d\mathbf{y}_{0:n}}, \quad (4.1.34)$$

$$p^2(\mathbf{y}_n|\mathbf{x}_n) = \frac{\int p^2(\boldsymbol{\xi}_0) \prod_{i=1}^n p_{i|i-1}^2(\boldsymbol{\xi}_i|\boldsymbol{\xi}_{i-1}) d\mathbf{x}_{0:n-1} d\mathbf{y}_{0:n-1}}{\int p^2(\boldsymbol{\xi}_0) \prod_{i=1}^n p_{i|i-1}^2(\boldsymbol{\xi}_i|\boldsymbol{\xi}_{i-1}) d\mathbf{x}_{0:n-1} d\mathbf{y}_{0:n}}; \quad (4.1.35)$$

Le résultat suivant repose donc sur un raisonnement par récurrence et une preuve peut être trouvée en Annexe C.1.

Proposition 4.1. *Considérons le modèle HMC avec*

$$p^1(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \mathbf{m}_0; \mathbf{P}_0), \quad (4.1.36)$$

$$f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n) \text{ et} \quad (4.1.37)$$

$$g_n(\mathbf{y}_n|\mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n), \quad (4.1.38)$$

pour tout n . Alors la classe de modèles couple linéaires et gaussiens $p^2(\cdot)$ qui vérifient les contraintes

$$p^2(\mathbf{x}_0) = p^1(\mathbf{x}_0), \quad (4.1.39)$$

$$p^2(\mathbf{x}_n|\mathbf{x}_{n-1}) = f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}), \quad (4.1.40)$$

$$p^2(\mathbf{y}_n|\mathbf{x}_n) = g_n(\mathbf{y}_n|\mathbf{x}_n), \quad (4.1.41)$$

et tels que $p_{n|n-1}^2(\mathbf{x}_n, \mathbf{y}_n|\mathbf{x}_{n-1}, \mathbf{y}_{n-1})$ ne dépend pas de \mathbf{m}_0 , est décrite par les équations suivantes :

$$p^2(\boldsymbol{\xi}_0) = \mathcal{N}\left(\boldsymbol{\xi}_0; \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{H}_0 \mathbf{m}_0 \end{bmatrix}; \begin{bmatrix} \mathbf{P}_0 & (\mathbf{H}_0 \mathbf{P}_0)^T \\ \mathbf{H}_0 \mathbf{P}_0 & \mathbf{R}_0 + \mathbf{H}_0 \mathbf{P}_0 \mathbf{H}_0^T \end{bmatrix}\right), \quad (4.1.42)$$

$$p_{n|n-1}^2(\boldsymbol{\xi}_n|\boldsymbol{\xi}_{n-1}) = \mathcal{N}(\boldsymbol{\xi}_n; \mathbf{B}_n \boldsymbol{\xi}_{n-1}; \boldsymbol{\Sigma}_n), \quad (4.1.43)$$

où les matrices \mathbf{B}_n et $\boldsymbol{\Sigma}_n$ sont définies par

$$\mathbf{B}_n = \begin{bmatrix} \mathbf{F}_n - \mathbf{F}_n^2 \mathbf{H}_{n-1} & \mathbf{F}_n^2 \\ \mathbf{H}_n \mathbf{F}_n - \mathbf{H}_n^2 \mathbf{H}_{n-1} & \mathbf{H}_n^2 \end{bmatrix}, \quad (4.1.44)$$

$$\boldsymbol{\Sigma}_n = \begin{bmatrix} \boldsymbol{\Sigma}_n^{11} & \boldsymbol{\Sigma}_n^{21T} \\ \boldsymbol{\Sigma}_n^{21} & \boldsymbol{\Sigma}_n^{22} \end{bmatrix}, \quad (4.1.45)$$

$$\boldsymbol{\Sigma}_n^{11} = \mathbf{Q}_n - \mathbf{F}_n^2 \mathbf{R}_{n-1} (\mathbf{F}_n^2)^T, \quad (4.1.46)$$

$$\boldsymbol{\Sigma}_n^{21} = \mathbf{H}_n \mathbf{Q}_n - \mathbf{H}_n^2 \mathbf{R}_{n-1} (\mathbf{F}_n^2)^T, \quad (4.1.47)$$

$$\boldsymbol{\Sigma}_n^{22} = \mathbf{R}_n - \mathbf{H}_n^2 \mathbf{R}_{n-1} (\mathbf{H}_n^2)^T + \mathbf{H}_n \mathbf{Q}_n \mathbf{H}_n^T, \quad (4.1.48)$$

et où $\{(\mathbf{F}_n^2, \mathbf{H}_n^2)\}_{n \geq 1}$ peuvent être choisis arbitrairement à condition que Σ_n soit une matrice de covariance définie positive pour tout n .

Remarque 4.1. Posons $\mathbf{H}_n^2 = \mathbf{H}_n \mathbf{F}_n^2$. D'après le Lemme A.2 en Annexe A, $p^2(\mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{y}_{n-1})$ coïncide avec $g_n(\mathbf{y}_n | \mathbf{x}_n)$. De plus, si $\mathbf{F}_n^2 = \mathbf{0}_{m \times p}$ alors $p^2(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$ coïncide avec $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$ et le modèle coïncide alors avec le modèle HMC (4.1.7)-(4.1.8).

Nous étudions maintenant les propriétés d'invariance de ces modèles particuliers à travers les deux propositions suivantes dont les preuves sont données en Annexe C.2.

Proposition 4.2. La loi $p^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1})$ est indépendante du choix de \mathbf{F}_n^2 et de \mathbf{H}_n^2 . Plus précisément, nous avons pour tout $\{\mathbf{F}_n^2, \mathbf{H}_n^2\}_{n \geq 1}$,

$$p^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}) = p^1(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}) = f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) g_n(\mathbf{y}_n | \mathbf{x}_n). \quad (4.1.49)$$

Cependant, rappelons que dans un HMC nous avons $p^1(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}) = p^1(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$, alors qu'en général, $p^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$ est différent de $p^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1})$.

D'après la Proposition 4.2, nous savons que $p^2(\mathbf{y}_n | \mathbf{x}_n, \mathbf{x}_{n-1}) = p^2(\mathbf{y}_n | \mathbf{x}_n) = g_n(\mathbf{y}_n | \mathbf{x}_n)$. La Proposition 4.3 se focalise sur la loi $p^2(\mathbf{y}_n | \mathbf{x}_{0:n})$:

Proposition 4.3. $p^2(\mathbf{y}_n | \mathbf{x}_{0:n})$ et $p^2(\mathbf{x}_{0:n})$ sont indépendants de \mathbf{F}_n^2 et \mathbf{H}_n^2 :

$$p^2(\mathbf{y}_n | \mathbf{x}_{0:n}) = p^1(\mathbf{y}_n | \mathbf{x}_n) = g_n(\mathbf{y}_n | \mathbf{x}_n), \quad (4.1.50)$$

$$p^2(\mathbf{x}_{0:n}) = p^1(\mathbf{x}_{0:n}) = p^1(\mathbf{x}_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1}). \quad (4.1.51)$$

Enfin, intéressons nous à la structure globale de $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$. D'après (4.1.51), quels que soient les paramètres \mathbf{F}_n^2 et \mathbf{H}_n^2 , $\{\mathbf{X}_n\}_{n \geq 0}$ est une chaîne de Markov dont les transitions sont données par les fonctions $\{f_{n|n-1}\}_{n \geq 1}$. En d'autres termes, $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ ne diffère de $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ qu'à travers la loi $p^2(\mathbf{y}_{0:n} | \mathbf{x}_{0:n})$ qui, rappelons le, admet dans un modèle PMC la factorisation (4.1.4). Cependant, certaines simplifications peuvent intervenir. D'après la Proposition 4.1, si $\mathbf{H}_n^2 = \mathbf{H}_n \mathbf{F}_n^2$, la loi $p^2(\mathbf{y}_i | \mathbf{y}_{i-1}, \mathbf{x}_{i-1:n})$ coïncide avec $p^2(\mathbf{y}_i | \mathbf{x}_{i:i+1})$. D'autre part, si $\mathbf{F}_n^2 = \mathbf{0}$, alors $p^2(\mathbf{y}_i | \mathbf{y}_{i-1}, \mathbf{x}_{i-1:n})$ se réduit à $p^2(\mathbf{y}_i | \mathbf{y}_{i-1}, \mathbf{x}_{i-1:i})$. Enfin, si nous posons $\mathbf{H}_n^2 = \mathbf{H}_n \mathbf{F}_n^2$ et $\mathbf{F}_n^2 = \mathbf{0}$, alors $p^2(\mathbf{y}_i | \mathbf{y}_{i-1}, \mathbf{x}_{i-1:n})$ devient $p^2(\mathbf{y}_i | \mathbf{x}_i) = g_i(\mathbf{y}_i | \mathbf{x}_i)$, et (4.1.4) coïncide avec la loi conditionnelle $p^1(\mathbf{y}_{0:n} | \mathbf{x}_{0:n})$ du modèle HMC.

4.1.6 Modèles de Markov triplet

Nous considérons maintenant des modèles statistiques dans lesquels le processus $(\mathbf{X}_{0:n}, \mathbf{Y}_{0:n})$ dépend d'un processus discret $\{R_n\}_{n \geq 0}$. Le modèle statistique le plus répandu est le modèle à sauts (4.0.2), dans lequel $(\mathbf{X}_{0:n}, \mathbf{Y}_{0:n})$ est un HMC conditionnellement à $\mathbf{R}_{0:n}$. Par conséquent, les propriétés D.1-D.2 ainsi que P.1-P.3 des modèles HMC sont encore vérifiées pour ces modèles, excepté qu'elles sont à présent conditionnelles à $\mathbf{R}_{0:n} = \mathbf{r}_{0:n}$.

De la même façon que les modèles PMC généralisent les modèles HMC, il est possible de généraliser les modèles à sauts (4.0.2) par la classe des modèles triplet; en effet, dans les modèles (4.0.2), le couple $((\mathbf{X}_n, R_n), \mathbf{Y}_n)$ est un HMC dont les transitions s'écrivent $p(r_n | r_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n)$ et les vraisemblances $g_n(\mathbf{y}_n | \mathbf{x}_n, r_n)$; de ce fait, l'une des propriétés des modèles (4.0.2) est que le triplet $\{\mathbf{X}_n, \mathbf{Y}_n, R_n\}_{n \geq 0}$ est une chaîne de Markov, c'est à dire

$$p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{0:n-1}, \mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}) = p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{r}_{n-1}) \quad (4.1.52)$$

avec

$$p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{r}_{n-1}) = p(r_n | r_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n). \quad (4.1.53)$$

Le modèle à sauts classique est donc un modèle de Markov triplet particulier, un modèle de Markov triplet étant un modèle tel que $\{\mathbf{X}_n, \mathbf{Y}_n, R_n\}_{n \geq 0}$ est une chaîne de Markov [93] [95], c'est à dire un modèle vérifiant (4.1.52). Par conséquent, la forme générale de la loi du triplet joint $p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ s'écrit

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n}) = p(\mathbf{x}_0, \mathbf{y}_0, r_0) \prod_{i=1}^n p(\mathbf{x}_i, \mathbf{y}_i, r_i | \mathbf{x}_{i-1}, \mathbf{y}_{i-1}, r_{i-1}). \quad (4.1.54)$$

Notons quelques particularités des modèles de Markov triplet [93] [3] :

- le processus discret $\{R_n\}_{n \geq 0}$ n'est pas nécessairement une chaîne de Markov ;
- ni $\{\mathbf{X}_n\}_{n \geq 0}$, ni $\{\mathbf{X}_n, \mathbf{Y}_n\}_{n \geq 0}$ ne sont nécessairement markoviens. Par conséquent, les modèles de Markov triplet sont plus généraux que les modèles de Markov couple ;
- même conditionnellement à $\mathbf{R}_{0:n} = \mathbf{r}_{0:n}$, le processus $\{\mathbf{X}_n\}_{n \geq 0}$ n'est pas nécessairement markovien. Cependant, conditionnellement à $\mathbf{R}_{0:n} = \mathbf{r}_{0:n}$, le processus $\{\mathbf{X}_n, \mathbf{Y}_n\}_{n \geq 0}$ reste markovien.

Comme nous l'avons vu, le modèle JMSS est un modèle triplet particulier dans lequel de plus (4.1.53) est vérifiée. Notons qu'il est possible de construire d'autres modèles de Markov triplet, par exemple en considérant les modèles de Markov couple présentés dans le Paragraphe 4.1.4 et en introduisant un troisième processus $\{R_n\}_{n \geq 0}$, non nécessairement markovien, de telle façon que les transitions $p_{n|n-1}$ des différents modèles couple dépendent de r_{n-1} et de r_n .

4.1.7 Filtrage dans les modèles de Markov triplet

Nous présentons trois techniques de filtrage statistique pour calculer ou approcher un estimateur de Θ_n dans les modèles triplet [38] [39] [3] [95]. Nous montrons qu'il est possible d'étendre le FP ainsi que le FP marginalisé pour certaines classes de modèles triplet. Enfin, nous discutons de la possibilité du calcul exact de Θ_n avec une complexité linéaire en nombre d'observations est également faisable pour certains modèles triplet [95] [1]. Ce dernier point est particulièrement intéressant car rappelons que dans les modèles (4.0.2), même linéaires et gaussiens, le calcul exact de Θ_n n'est jamais faisable exactement. Cette technique de filtrage exact sera par ailleurs la pierre angulaire de la Section 4.3, qui proposera une alternative aux approximations classiques pour les modèles à sauts (4.0.3)-(4.0.4).

Filtrage particulaire

De façon générale, la loi de filtrage est propagée à partir de la formule

$$p(\mathbf{x}_n, r_n | \mathbf{y}_{0:n}) = \frac{\sum_{r_{n-1}} \int p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1}) p(\mathbf{x}_{n-1}, r_{n-1} | \mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1}}{\sum_{r_{n-1}} \int p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1}) p(\mathbf{x}_{n-1}, r_{n-1} | \mathbf{y}_{0:n-1}) d\mathbf{x}_{n-1} d\mathbf{y}_n}. \quad (4.1.55)$$

Par conséquent, le FP est identique à celui des modèles couple, si ce n'est que les particules sont tirées en dimension augmentée, selon une loi d'importance $q_n(\mathbf{x}_{0:n}, \mathbf{r}_{0:n})$; les poids associés aux particules $(\mathbf{x}_{0:n}^i, \mathbf{r}_{0:n}^i)$ s'écrivent ici

$$w_n^i = w_{n-1}^i \frac{p(\mathbf{x}_n^i, \mathbf{y}_n, r_n^i | \mathbf{x}_{n-1}^i, \mathbf{y}_{n-1}, r_{n-1}^i)}{q(\mathbf{x}_n^i, r_n^i | \mathbf{x}_{0:n-1}^i, \mathbf{r}_{0:n-1}^i)}, \quad \sum_{i=1}^N w_n^i = 1. \quad (4.1.56)$$

Filtrage particulière marginalisé

Le filtrage particulière marginalisé peut être utilisé pour certaines classes de modèles triplet [1]. Nous nous plaçons dans le cas où la transition du modèle triplet s'écrit

$$p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1}) = p(r_n | \mathbf{y}_{n-1}, r_{n-1}) p(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}, \mathbf{r}_{n-1:n}), \quad (4.1.57)$$

où

$$p(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}, \mathbf{r}_{n-1:n}) = \mathcal{N} \left(\boldsymbol{\xi}_n; \begin{bmatrix} \mathbf{F}_n^1(\mathbf{r}_{n-1:n}) & \mathbf{F}_n^2(\mathbf{r}_{n-1:n}) \\ \mathbf{H}_n^1(\mathbf{r}_{n-1:n}) & \mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \end{bmatrix} \boldsymbol{\xi}_{n-1}; \begin{bmatrix} \boldsymbol{\Sigma}_n^{11}(\mathbf{r}_{n-1:n}) & \boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n})^T \\ \boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n}) & \boldsymbol{\Sigma}_n^{22}(\mathbf{r}_{n-1:n}) \end{bmatrix} \right). \quad (4.1.58)$$

Notons que dans ce modèle, $\{R_n\}_{n \geq 0}$ n'est pas nécessairement une chaîne de Markov. Cependant, si $p(r_n | \mathbf{y}_{n-1}, r_{n-1}) = p(r_n | r_{n-1})$, $\mathbf{F}_n^1(\mathbf{r}_{n-1:n}) = \mathbf{F}_n(r_n)$, $\mathbf{F}_n^2(\mathbf{r}_{n-1:n}) = \mathbf{0}$, $\mathbf{H}_n^1(\mathbf{r}_{n-1:n}) = \mathbf{H}_n(r_n) \mathbf{F}_n(r_n)$, $\mathbf{H}_n^2(\mathbf{r}_{n-1:n}) = \mathbf{0}$, $\boldsymbol{\Sigma}_n^{11}(\mathbf{r}_{n-1:n}) = \mathbf{Q}_n(r_n)$, $\boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n}) = \mathbf{H}_n(r_n) \mathbf{Q}_n(r_n)$ et $\boldsymbol{\Sigma}_n^{22}(\mathbf{r}_{n-1:n}) = \mathbf{R}_n + \mathbf{H}_n(r_n) \mathbf{Q}_n(r_n) \mathbf{H}_n(r_n)^T$, alors ces modèles coïncident avec les modèles à sauts markoviens linéaires et gaussiens (4.0.3)-(4.0.4). Dans les modèles (4.1.57)-(4.1.58), le filtrage exact (au sens du calcul exact de Θ_n) n'est pas possible ; le cas échéant, cela signifierait qu'il est également faisable pour les modèles (4.0.3)-(4.0.4). Cependant, nous pouvons éviter partiellement le FP du paragraphe précédent, en partant de la factorisation générale

$$p(\mathbf{x}_n, \mathbf{r}_{0:n} | \mathbf{y}_{0:n}) = p(\mathbf{r}_{0:n} | \mathbf{y}_{0:n}) p(\mathbf{x}_n | \mathbf{y}_{0:n}, \mathbf{r}_{0:n}). \quad (4.1.59)$$

L'équation (4.1.59) suggère alors de combiner un FP pour approcher $p(\mathbf{r}_{0:n} | \mathbf{y}_{0:n})$ par un ensemble de N points pondérés $\{\mathbf{r}_{0:n}^i, w_n^i\}_{i=1}^N$, puis de calculer pour chaque tirage $\mathbf{r}_{0:n}^i$ la loi $p(\mathbf{x}_n | \mathbf{y}_{0:n}, \mathbf{r}_{0:n}^i)$, qui ici est une gaussienne dont les paramètres sont donnés par le FK pour les modèles PMC (voir Algorithme 4.1).

Filtrage exact

Un des intérêts des modèles triplet est que pour une certaine classe de modèles, le calcul exact de Θ_n est faisable. En particulier, nous rappelons le résultat [95] (que nous utilisons ici dans le cas gaussien [1]) qui montre que si la transition du triplet se factorise de la façon suivante :

$$p(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1}) = p(\mathbf{y}_n, r_n | \mathbf{y}_{n-1}, r_{n-1}) p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1:n}, \mathbf{r}_{n-1:n}), \quad (4.1.60)$$

(c'est à dire si $p(\mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1}) = p(\mathbf{y}_n, r_n | \mathbf{y}_{n-1}, r_{n-1})$) avec

$$p(\mathbf{x}_n | \boldsymbol{\xi}_{n-1}, \mathbf{y}_n, \mathbf{r}_{n-1:n}) = \mathcal{N}(\mathbf{x}_n; \mathbf{C}_n(\mathbf{r}_{n-1:n}) \boldsymbol{\xi}_{n-1} + \mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n}); \boldsymbol{\Sigma}_n^{\mathbf{x}}(\mathbf{r}_{n-1:n})), \quad (4.1.61)$$

alors Θ_n est calculable exactement, avec un coût linéaire en nombre d'observations. La proposition suivante, dont une preuve est donnée en Annexe C.3, détaille le calcul de

$$\Theta_n = \sum_{r_n} p(r_n | \mathbf{y}_{0:n}) \mathbb{E}(f(\mathbf{X}_n) | \mathbf{y}_{0:n}, r_n). \quad (4.1.62)$$

lorsque $f(\mathbf{x}_n) = \mathbf{x}_n$ ou $f(\mathbf{x}_n) = \mathbf{x}_n \mathbf{x}_n^T$.

Proposition 4.4. Soit un modèle triplet (4.1.54) vérifiant (4.1.60)-(4.1.61). Alors $p(r_n|\mathbf{y}_{0:n})$, $E(\mathbf{X}_n|\mathbf{y}_{0:n}, r_n)$ et $E(\mathbf{X}_n\mathbf{X}_n^T|\mathbf{y}_{0:n}, r_n)$ sont calculables récursivement :

$$p(r_n|\mathbf{y}_{0:n}) \propto \sum_{r_{n-1}} p(r_n|r_{n-1})p^2(\mathbf{y}_n|\mathbf{y}_{n-1}, \mathbf{r}_{n-1:n})p(r_{n-1}|\mathbf{y}_{0:n-1}), \quad (4.1.63)$$

$$E(\mathbf{X}_n|\mathbf{y}_{0:n}, r_n) = \sum_{r_{n-1}} p(r_{n-1}|r_n, \mathbf{y}_{0:n}) (\mathbf{C}_n(\mathbf{r}_{n-1:n})E(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, r_{n-1}) + \mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n})), \quad (4.1.64)$$

$$E(\mathbf{X}_n\mathbf{X}_n^T|\mathbf{y}_{0:n}, r_n) = \sum_{r_{n-1}} p(r_{n-1}|r_n, \mathbf{y}_{0:n}) (\Sigma_n^x(\mathbf{r}_{n-1:n}) + \Delta(\mathbf{r}_{n-1:n})) \quad (4.1.65)$$

$$p(r_{n-1}|r_n, \mathbf{y}_{0:n}) \propto p(r_n|r_{n-1})p(\mathbf{y}_n|\mathbf{y}_{n-1}, \mathbf{r}_{n-1:n})p^2(r_{n-1}|\mathbf{y}_{0:n-1}), \quad (4.1.66)$$

$$\begin{aligned} \Delta(\mathbf{r}_{n-1:n}) &= \mathbf{C}_n(\mathbf{r}_{n-1:n})E(\mathbf{x}_{n-1}\mathbf{x}_{n-1}^T|\mathbf{y}_{0:n-1}, r_{n-1})\mathbf{C}_n(\mathbf{r}_{n-1:n})^T \\ &+ \mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n})E(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, r_{n-1})^T\mathbf{C}_n(\mathbf{r}_{n-1:n})^T \\ &+ \mathbf{C}_n(\mathbf{r}_{n-1:n})E(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, r_{n-1})\mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n})^T \\ &+ \mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n})\mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n})^T. \end{aligned} \quad (4.1.67)$$

Notons que le coût impliqué dans le calcul de Θ_n ne croît pas exponentiellement au cours du temps et ne nécessite que le calcul de sommes sur r_{n-1} et r_n . L'objet de la Section 4.3 sera d'utiliser ce résultat pour construire un nouvel estimateur de Θ_n dans les modèles à sauts linéaires et gaussiens (4.0.3)-(4.0.4).

4.2 Filtrage multi-objets pour les chaînes de Markov couple

Les modèles couple peuvent être plus adaptés que les modèles HMC pour traiter des données aléatoires ; par conséquent, nous nous demandons comment utiliser ces modèles pour améliorer les méthodes de filtrage multi-objets. Jusqu'à présent, deux directions ont été exploitées pour améliorer les solutions basées sur la propagation du PHD.

Comme nous l'avons mentionné au Paragraphe 1.2.2, les équations du filtre PHD ne peuvent pas être implémentées exactement. Certains travaux se sont donc focalisés sur les problèmes d'implémentation du filtre, comme cela a été fait dans de nombreuses contributions (voir [121] [27] [92] [122] [26] [125]) et comme nous l'avons fait dans le Chapitre 3.

À l'opposé, d'autres contributions ont cherché à assouplir les hypothèses sous-jacentes à la dérivation du filtre PHD. Par exemple, le filtre PHD a été initialement proposé sous des hypothèses de Poisson (voir les hypothèses A.2-A.3 dans le Paragraphe 1.2.2). Mahler a alors assoupli ces hypothèses, au prix d'une complexité plus importante, et a proposé le filtre CPHD qui en plus du PHD, propage également la loi de probabilité du nombre d'objets [82]. Des implémentations pratiques de ce filtre ont été proposées par la suite [123], et des améliorations concernant les hypothèses de départ ont pu être apportées [84].

Dans cette section, nous nous focalisons sur le calcul du filtre PHD lorsque les hypothèses d'indépendances usuelles suivies par chaque état caché et son observation associée ne sont plus vérifiées. En effet, comme nous l'avons rappelé dans le Paragraphe 1.2.2, la plupart des algorithmes de filtrage multi-objets, en particulier le filtre bayésien multi-objets (1.2.7)-(1.2.8) ainsi que le filtre PHD (1.2.10)-(1.2.11), reposent sur l'hypothèse que chaque état caché et son observation associée suivent un modèle HMC. Cependant, comme nous l'avons vu au Paragraphe 4.1.4, ces modèles peuvent s'avérer inadaptés dans le cas où les données traitées suivent un modèle plus complexe [109]. Notre objectif est donc de propager le PHD dans le cas où les hypothèses des modèles HMC ne sont plus vérifiées. À cette fin, nous utilisons les modèles PMC présentés

dans le Paragraphe 4.1, après avoir proposé une expression du filtre bayésien multi-objets pour ces modèles. Bien sûr, notre filtre PHD pour les modèles PMC généralise le filtre PHD (1.2.10)-(1.2.11) dans la mesure où un HMC n'est qu'un PMC particulier (voir (4.1.2)-(4.1.3)).

Cette section est organisée de la manière suivante. Dans le Paragraphe 4.2.1, nous montrons que la propagation du PHD pour des modèles couple doit passer par une reformulation du problème de filtrage multi-objets. Ensuite, dans le Paragraphe 4.2.2, nous présentons les équations du filtre PHD pour les modèles PMC. Une implémentation par mélange de gaussiennes est alors proposée dans le Paragraphe 4.2.3. Enfin, dans le Paragraphe 4.2.4, l'amélioration des performances de notre filtre est illustrée sur des modèles de Markov couple possédant des propriétés physiques identiques à des modèles HMC (voir Proposition 4.1).

4.2.1 Les limites de l'approche classique

Nous supposons que chaque couple (état caché, observation associée) suit un modèle PMC (4.1.1), caractérisé par les transitions $\{p_{i|i-1}(\cdot)\}_{i \geq 1}$. Comme nous l'avons rappelé, les lois $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $p(\mathbf{z}|\mathbf{x}_n)$ existent dans un modèle PMC (elles sont données par (4.1.34)-(4.1.35)). Cependant, ces lois ne doivent pas être injectées à la place de $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $g_n(\mathbf{z}|\mathbf{x}_n)$ dans les équations du PHD (1.2.10)-(1.2.11) car cela supposerait implicitement que les propriétés statistiques **P.1** et **P.2** du Paragraphe 4.1 sont vérifiées, ce qui n'est pas nécessairement le cas. Notons qu'à ce stade, le calcul direct de $v_n(\mathbf{x})$, pour les modèles PMC, semble compliqué car \mathbf{X}_n n'est pas nécessairement markovien et la dynamique de chaque objet dépend de l'observation qui lui est associée par l'intermédiaire de la loi $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{n-1})$. Cependant, rappelons que dans un modèle couple, $\{\mathbf{X}_n, \mathbf{Y}_n\}_{n \geq 0}$ est une chaîne de Markov. Il est alors judicieux de propager une intensité jointe $v_n(\mathbf{x}, \mathbf{y})$ puis d'obtenir le PHD d'intérêt $v_n(\mathbf{x})$ à partir de $v_n(\mathbf{x}) = \int v_n(\mathbf{x}, \mathbf{y}) d\mathbf{y}$ (par un léger abus de notation, $v_n(\cdot)$ désigne à la fois une fonction de \mathbf{x} ou de (\mathbf{x}, \mathbf{y})).

L'intérêt de propager une fonction de (\mathbf{x}, \mathbf{y}) dans le contexte du filtrage multi-objets pour les modèles PMC peut être mis en avant à travers le scénario simple mono-objet suivant, dans lequel des non-détections peuvent survenir. Supposons qu'à l'instant $n - 1$, nous disposons de la loi de filtrage $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$. À l'instant n , \mathbf{Y}_n n'est pas observé (l'état caché \mathbf{X}_n n'est pas détecté) et nous souhaitons calculer, à l'instant $n + 1$, $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1, n+1})$ (\mathbf{X}_{n+1} est détecté). Posons $\mathbf{y}_{0:n-1, n+1} = (\mathbf{y}_{0:n-1}, \mathbf{y}_{n+1})$. Nous avons

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1, n+1}) \propto \int p_{n+1|n}(\xi_{n+1}|\xi_n)p(\xi_n|\mathbf{y}_{0:n-1})d\xi_n, \quad (4.2.1)$$

$$p(\xi_n|\mathbf{y}_{0:n-1}) = \int p_{n|n-1}(\xi_n|\xi_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1}. \quad (4.2.2)$$

Par conséquent, $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1, n+1})$ peut être calculée à partir de $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$ par l'intermédiaire de la loi jointe $p(\mathbf{x}_n, \mathbf{y}_n|\mathbf{y}_{0:n-1})$. Cela provient du fait que $p_{n+1|n}(\xi_{n+1}|\xi_n)$ dépend généralement de $\mathbf{Y}_n = \mathbf{y}_n$: même si \mathbf{Y}_n n'a pas été observé, un moyennage sur toutes les valeurs possibles prises par $\mathbf{Y}_n = \mathbf{y}_n$ est nécessaire pour construire $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1, n+1})$, d'où (4.2.1). À l'opposé, dans un modèle HMC, l'égalité (4.1.3) est vérifiée ; par conséquent $p_{n+1|n}(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}|\mathbf{x}_n, \mathbf{y}_n)$ ne dépend plus de \mathbf{y}_n ; le moyennage devient inutile, et en effet (4.2.1)-(4.2.2) deviennent

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1, n+1}) \propto g_{n+1}(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) \int f_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1})d\mathbf{x}_n, \quad (4.2.3)$$

$$p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) = \int f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})d\mathbf{x}_{n-1}. \quad (4.2.4)$$

Par conséquent, dans un HMC, $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n-1, n+1})$ peut être déduit de $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1})$ grâce au calcul

intermédiaire de $p(\mathbf{x}_n | \mathbf{y}_{0:n-1})$, et le calcul intermédiaire de $p(\mathbf{x}_n, \mathbf{y}_n | \mathbf{y}_{0:n-1})$ n'est plus nécessaire. C'est ce qui explique pourquoi la propagation de loi jointe dans les modèles HMC n'est pas nécessaire.

Ces difficultés sont d'autant plus importantes dans le contexte multi-objets, qu'à l'instant n , chaque objet $\mathbf{X}_{i,n}$ peut être détecté (ce qui signifie qu'il existe une réalisation $\mathbf{Y}_{i,n} = \mathbf{y}_{i,n}$ associée à $\mathbf{X}_{i,n}$), ou non, et certaines observations peuvent être des mesures de fausses alarmes. Nous cherchons alors à propager un filtre PHD joint (objet caché + observation) afin d'exploiter l'information apportée par les observations dans un modèle couple.

4.2.2 Un filtre PHD pour les modèles PMC

D'un point de vue statistique, si nous adaptons la formulation du problème de filtrage multi-objets en termes de RFS, notre problème consiste à calculer le PHD du RFS

$$\dot{\Xi}_n = \{(\mathbf{X}_{1,n}, \mathbf{Y}_{1,n}), \dots, (\mathbf{X}_{N(n),n}, \mathbf{Y}_{N(n),n})\}$$

conditionnellement à des réalisations des RFS $\Psi_{0:n} = \mathcal{Z}_{0:n}$, avec

$$\mathcal{Z}_n = \{\mathbf{z}_{1,n}, \dots, \mathbf{z}_{M(n),n}\}.$$

Par construction, l'évolution statistique d'un couple appartenant à $\dot{\Xi}_n$ ainsi que le lien statistique entre une mesure dans \mathcal{Z}_n et un couple dans $\dot{\Xi}_n$ sont donnés par les propriétés suivantes :

1. Comme chaque paire individuelle (objet + observation associée) suit un modèle PMC, $(\mathbf{X}_n, \mathbf{Y}_n)$ est une chaîne de Markov dont la transition est donnée par $p_{n|n-1}(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$. La loi $p_{n|n-1}$ inclut le modèle dynamique de l'objet, par l'intermédiaire de $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$, mais aussi celui du capteur par l'intermédiaire de $p(\mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{x}_n)$. Pour des raisons physiques, nous supposons que la probabilité de survie d'un couple ξ_n est donnée par $p_{s,n+1}(\mathbf{x}_n)$ et ne dépend que de \mathbf{x}_n ;
2. Étant donné une réalisation du couple $(\mathbf{X}_n, \mathbf{Y}_n) = (\mathbf{x}_n, \mathbf{y}_n) = \xi_n$, nous définissons la vraisemblance d'une mesure \mathbf{z} dans \mathcal{Z}_n par $\delta_{\mathbf{z}}(\mathbf{y}_n)$ où $\delta_{\mathbf{z}}(\mathbf{y}_n)$ désigne la fonction Dirac, c'est à dire $\int_{S_q} \delta_{\mathbf{z}}(\mathbf{y}_n) d\mathbf{y}_n = \mathbb{1}_{S_q}(\mathbf{z})$, pour $S_q \subset \mathbb{R}^q$ (voir par exemple [81, p. 693]). En effet, rappelons que \mathbf{Y}_n est la v.a. modélisant l'observation associée à \mathbf{X}_n ; pour une réalisation donnée \mathbf{y}_n , si $\mathbf{z} \neq \mathbf{y}_n$ nous pouvons prétendre que l'observation \mathbf{z} n'est pas produite par \mathbf{X}_n et que, conditionnellement à ξ_n , elle ne dépend pas des autres mesures passées. Nous supposons également que la probabilité de détection du couple ξ_n est donnée par $p_{d,n}(\mathbf{x}_n)$ et ne dépend donc que de \mathbf{x}_n .

D'après ces considérations statistiques, l'expression du filtre bayésien multi-objets s'écrit

$$p(\dot{\mathcal{X}}_n | \mathcal{Z}_{0:n-1}) = \int p_{n|n-1}(\dot{\mathcal{X}}_n | \dot{\mathcal{X}}_{n-1}) p(\dot{\mathcal{X}}_{n-1} | \mathcal{Z}_{0:n-1}) \delta \dot{\mathcal{X}}_{n-1}, \quad (4.2.5)$$

$$p(\dot{\mathcal{X}}_n | \mathcal{Z}_{0:n}) \propto p_n(\mathcal{Z}_n | \dot{\mathcal{X}}_n) p(\dot{\mathcal{X}}_n | \mathcal{Z}_{0:n-1}), \quad (4.2.6)$$

où $\delta \dot{\mathcal{X}}$ désigne l'intégrale multi-objets, $p_{n|n-1}(\dot{\mathcal{X}}_n | \dot{\mathcal{X}}_{n-1})$ et $p_n(\mathcal{Z}_n | \dot{\mathcal{X}}_n)$ sont les densités de transition et de vraisemblance multi-objets, qui peuvent être construites de manière identique aux densités multi-objets pour les modèles HMC (voir [81, eq. 13.43], [81, eq. 12.140] et [122]), si ce n'est que la transition d'un élément de $\dot{\mathcal{X}}_n$ est donné par $p_{n|n-1}(\xi_n | \xi_{n-1})$ et la vraisemblance d'une mesure \mathbf{z} de \mathcal{Z}_n par $\delta_{\mathbf{z}}(\mathbf{y}_n)$.

Afin de proposer un filtre PHD pour les modèles PMC, nous devons encore adapter les hypothèses **A.1-A.3** du Paragraphe 1.2.2 au contexte des modèles PMC. Les hypothèses **A.1-A.3** deviennent alors **A'.1-A'.3**.

L'hypothèse **A'.1** stipule que chaque paire (objet + observation) suit un modèle PMC et évolue indépendamment de chaque autre paire ; **A'.2** coïncide avec **A.2** ; et **A'.3** suppose que $p(\dot{X}_n | \mathcal{Z}_{0:n-1})$ est de Poisson (voir Remarque 1.1). Sous ces hypothèses, le PHD $v_n(\mathbf{x}, \mathbf{y})$ est propagé par une étape de prédiction et une étape de mise à jour et nous obtenons le filtre PMC-PHD :

$$v_{n|n-1}(\mathbf{x}, \mathbf{y}) = \int p_{s,n}(\mathbf{x}_{n-1}) p_{n|n-1}(\mathbf{x}, \mathbf{y} | \boldsymbol{\xi}_{n-1}) v_{n-1}(\boldsymbol{\xi}_{n-1}) d\boldsymbol{\xi}_{n-1} + \gamma_n(\mathbf{x}, \mathbf{y}), \quad (4.2.7)$$

$$v_n(\mathbf{x}, \mathbf{y}) = [1 - p_{d,n}(\mathbf{x})] v_{n|n-1}(\mathbf{x}, \mathbf{y}) + \sum_{\mathbf{z} \in \mathcal{Z}_n} \frac{p_{d,n}(\mathbf{x}) v_{n|n-1}(\mathbf{x}, \mathbf{z}) \delta_{\mathbf{z}}(\mathbf{y})}{\kappa_n(\mathbf{z}) + \int p_{d,n}(\mathbf{x}) v_{n|n-1}(\mathbf{x}, \mathbf{z}) d\mathbf{x}}. \quad (4.2.8)$$

Nous rappelons que l'estimation du nombre d'objets et de leur état caché est déduite du calcul de l'intégrale $v_n(\mathbf{x}) = \int v_n(\mathbf{x}, \mathbf{y}) d\mathbf{y}$. L'intégrale par rapport à \mathbf{y} peut se réduire à une somme selon la nature de v_n : le premier terme dans (4.2.8) est associé aux couples (\mathbf{x}, \mathbf{y}) non détectés, tandis que le second terme est associé aux couples détectés. Il n'est ainsi par surprenant que l'espérance du nombre de paires soit nulle dans les régions où $\mathbf{y} \neq \mathbf{z}$, pour tout $\mathbf{z} \in \mathcal{Z}_n$.

Remarque 4.2. Nous analysons les équations de propagation du PHD $v_n(\mathbf{x}) = \int v_n(\mathbf{x}, \mathbf{y}) d\mathbf{y}$ lorsque les paramètres du filtre PMC-PHD coïncident avec ceux du filtre PHD classique (que nous nommons ici HMC-PHD). Dans ce cas, pour des fonctions données $f_{n|n-1}(\mathbf{x} | \mathbf{x}_{n-1})$, $g_n(\mathbf{y} | \mathbf{x})$ et $\gamma_n(\mathbf{x})$ nous construisons

$$p_{n|n-1}(\mathbf{x}, \mathbf{y} | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) = f_{n|n-1}(\mathbf{x} | \mathbf{x}_{n-1}) g_n(\mathbf{y} | \mathbf{x}), \quad (4.2.9)$$

$$\gamma_n(\mathbf{x}, \mathbf{y}) = g_n(\mathbf{y} | \mathbf{x}) \tilde{\gamma}_n(\mathbf{x}) \quad (4.2.10)$$

(ce dernier choix est justifié par le fait que dans un HMC, la vraisemblance d'une mesure \mathbf{y} avec l'état caché \mathbf{x} est donnée par $g_n(\mathbf{y} | \mathbf{x})$). En écrivant

$$v_{n-1}(\mathbf{x}, \mathbf{y}) = \frac{v_{n-1}(\mathbf{x}, \mathbf{y})}{\int v_{n-1}(\mathbf{x}, \mathbf{y}) d\mathbf{y}} v_{n-1}(\mathbf{x}), \quad (4.2.11)$$

puis en appliquant (4.2.7)-(4.2.8), nous observons que $v_{n-1}(\mathbf{x})$ est propagé à partir des équations de propagation du PHD classique (voir (1.2.10)-(1.2.11)), ce qui signifie que le filtre PMC-PHD (marginalisé par rapport à \mathbf{y}) correspond au filtre PHD classique.

4.2.3 Une implémentation par mélanges de gaussiennes du filtre PMC-PHD

Dans ce paragraphe, nous nous focalisons sur les modèles PMC linéaires et gaussiens,

$$p(\boldsymbol{\xi}_0) = \mathcal{N}(\boldsymbol{\xi}_0; \tilde{\mathbf{m}}_0; \tilde{\mathbf{P}}_0), \quad (4.2.12)$$

$$p_{n|n-1}(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}) = \mathcal{N}(\boldsymbol{\xi}_n; \mathbf{B}_n \boldsymbol{\xi}_{n-1}; \boldsymbol{\Sigma}_n), \quad (4.2.13)$$

dont l'intérêt pratique a été illustré dans le Paragraphe 4.1.4. Nous supposons que l'intensité des naissances des couples $\gamma_n(\mathbf{x}, \mathbf{y})$ est un mélange gaussien :

$$\gamma_n(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{J_{\gamma_n}} w_{\gamma_n}^{(i)} \mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{\gamma_n}^{(i)}, \mathbf{P}_{\gamma_n}^{(i)}), \quad (4.2.14)$$

et que les probabilités de survie $p_{s,n}$ et de détection $p_{d,n}$ sont indépendantes de \mathbf{x} . Nous supposons également qu'à l'instant $n - 1$, le PHD v_{n-1} s'écrit

$$v_{n-1}(\mathbf{x}, \mathbf{y}) = v_{n-1}^1(\mathbf{x}, \mathbf{y}) + v_{n-1}^2(\mathbf{x}, \mathbf{y}), \quad (4.2.15)$$

où

$$v_{n-1}^1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{J_{n-1}^1} w_{n-1}^{1,(i)} \mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{n-1}^{1,(i)}, \mathbf{P}_{n-1}^{1,(i)}), \quad (4.2.16)$$

$$v_{n-1}^2(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{J_{n-1}^2} w_{n-1}^{2,(i)} \mathcal{N}(\mathbf{x}; \mathbf{m}_{n-1}^{2,(i)}, \mathbf{P}_{n-1}^{2,(i)}) \delta_{\mathbf{z}^{(i)}}(\mathbf{y}), \quad (4.2.17)$$

où $\mathbf{z}^{(i)}$ appartient à un ensemble donné $\mathcal{Z} = \{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(J_{n-1}^2)}\}$, avec d'éventuelles répétitions.

Étape de prédiction

En injectant (4.2.15), (4.2.16) et (4.2.17) dans (4.2.7), le PHD $v_{n|n-1}(\mathbf{x}, \mathbf{y})$ est un mélange gaussien,

$$v_{n|n-1}(\mathbf{x}, \mathbf{y}) = v_{n|n-1}^1(\mathbf{x}, \mathbf{y}) + v_{n|n-1}^2(\mathbf{x}, \mathbf{y}) + \gamma_n(\mathbf{x}, \mathbf{y}), \quad (4.2.18)$$

$$v_{n|n-1}^1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{J_{n-1}^1} w_{n|n-1}^{1,(i)} \mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{n|n-1}^{1,(i)}, \mathbf{P}_{n|n-1}^{1,(i)}), \quad (4.2.19)$$

$$v_{n|n-1}^2(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{J_{n-1}^2} w_{n|n-1}^{2,(i)} \mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{n|n-1}^{2,(i)}, \mathbf{P}_{n|n-1}^{2,(i)}), \quad (4.2.20)$$

où

$$w_{n|n-1}^{j,(i)} = p_{s,n} w_{n-1}^{j,(i)} \text{ pour } j = \{1, 2\}, \quad (4.2.21)$$

$$\mathbf{m}_{n|n-1}^{1,(i)} = \mathbf{B}_n \mathbf{m}_{n-1}^{1,(i)}, \quad \mathbf{m}_{n|n-1}^{2,(i)} = \mathbf{B}_n \begin{bmatrix} \mathbf{m}_{n-1}^{2,(i)} \\ \mathbf{z}^{(i)} \end{bmatrix}, \quad (4.2.22)$$

$$\mathbf{P}_{n|n-1}^{1,(i)} = \Sigma_n + \mathbf{B}_n \mathbf{P}_{n-1}^{1,(i)} \mathbf{B}_n^T, \quad (4.2.23)$$

$$\mathbf{P}_{n|n-1}^{2,(i)} = \Sigma_n + \begin{bmatrix} \mathbf{F}_n^1 \\ \mathbf{H}_n^1 \end{bmatrix} \mathbf{P}_{n-1}^{2,(i)} [(\mathbf{F}_n^1)^T (\mathbf{H}_n^1)^T]. \quad (4.2.24)$$

Ce résultat est basé sur le Lemme A.1 (voir Annexe A) appliqué à $\zeta = (\mathbf{x}, \mathbf{y})$, $\eta = (\mathbf{x}_{n-1}, \mathbf{y}_{n-1})$ pour le calcul de $v_{n|n-1}^1(\mathbf{x}, \mathbf{y})$ et à $\zeta = (\mathbf{x}, \mathbf{y})$, $\eta = \mathbf{x}_{n-1}$ pour celui de $v_{n|n-1}^2(\mathbf{x}, \mathbf{y})$. Enfin, en réarrangeant les termes du mélange (4.2.18) et en posant $J_{n|n-1} = J_{n-1}^1 + J_{n-1}^2 + J_{\gamma_n}$, $v_{n|n-1}$ peut se réécrire sous la forme

$$v_{n|n-1}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{J_{n|n-1}} w_{n|n-1}^{(i)} \mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{n|n-1}^{(i)}, \mathbf{P}_{n|n-1}^{(i)}), \quad (4.2.25)$$

où nous introduisons les notations suivantes :

$$\mathbf{m}_{n|n-1}^{(i)} = \begin{bmatrix} \mathbf{m}_{n|n-1}^{\mathbf{x},(i)} \\ \mathbf{m}_{n|n-1}^{\mathbf{y},(i)} \end{bmatrix}, \quad \mathbf{P}_{n|n-1}^{(i)} = \begin{bmatrix} \mathbf{P}_{n|n-1}^{\mathbf{x},(i)} & \mathbf{P}_{n|n-1}^{\mathbf{xy},(i)} \\ \mathbf{P}_{n|n-1}^{\mathbf{xy},(i)T} & \mathbf{P}_{n|n-1}^{\mathbf{y},(i)} \end{bmatrix}. \quad (4.2.26)$$

Étape de mise à jour

En injectant (4.2.25) dans (4.2.8) et en utilisant le Lemme A.2 (voir Annexe A), nous obtenons l'expression suivante pour $v_n(\mathbf{x}, \mathbf{y})$:

$$v_n(\mathbf{x}, \mathbf{y}) = v_n^1(\mathbf{x}, \mathbf{y}) + v_n^2(\mathbf{x}, \mathbf{y}), \quad (4.2.27)$$

où

$$v_n^1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{J_{n|n-1}} w_n^{1,(i)} \mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{n|n-1}^{(i)}; \mathbf{P}_{n|n-1}^{(i)}), \quad (4.2.28)$$

$$v_n^2(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{z} \in \mathcal{Z}_n} \sum_{i=1}^{J_{n|n-1}} w_n^{2,(i)}(\mathbf{z}) \mathcal{N}(\mathbf{x}; \mathbf{m}_n^{2,(i)}(\mathbf{z}); \mathbf{P}_n^{2,(i)}) \delta_{\mathbf{z}}(\mathbf{y}), \quad (4.2.29)$$

où

$$w_n^{1,(i)} = (1 - p_{d,n}) w_{n|n-1}^{(i)}, \quad (4.2.30)$$

$$w_n^{2,(i)}(\mathbf{z}) = \frac{p_{d,n} w_{n|n-1}^{(i)} q_n^{(i)}(\mathbf{z})}{\kappa_n(\mathbf{z}) + p_{d,n} \sum_{i=1}^{J_{n|n-1}} w_{n|n-1}^{(i)} q_n^{(i)}(\mathbf{z})}, \quad (4.2.31)$$

$$q_n^{(i)}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{m}_{n|n-1}^{\mathbf{y},(i)}; \mathbf{P}_{n|n-1}^{\mathbf{y},(i)}), \quad (4.2.32)$$

$$\mathbf{m}_n^{2,(i)}(\mathbf{z}) = \mathbf{m}_{n|n-1}^{\mathbf{x},(i)} + \mathbf{K}_n^{(i)}(\mathbf{z} - \mathbf{m}_{n|n-1}^{\mathbf{y},(i)}), \quad (4.2.33)$$

$$\mathbf{P}_n^{2,(i)} = \mathbf{P}_{n|n-1}^{\mathbf{x},(i)} - \mathbf{K}_n^{(i)} \mathbf{P}_{n|n-1}^{\mathbf{x}\mathbf{y},(i)T}, \quad (4.2.34)$$

$$\mathbf{K}_n^{(i)} = \mathbf{P}_{n|n-1}^{\mathbf{x}\mathbf{y},(i)} (\mathbf{P}_{n|n-1}^{\mathbf{y},(i)})^{-1}. \quad (4.2.35)$$

En réorganisant les termes du mélange (4.2.29) puis en posant $J_n^2 = J_{n|n-1} \times |\mathcal{Z}_n|$, $v_n^2(\mathbf{x}, \mathbf{y})$ peut se réécrire sous la forme

$$v_n^2(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{J_n^2} w_n^{2,(i)} \mathcal{N}(\mathbf{x}; \mathbf{m}_n^{2,(i)}; \mathbf{P}_n^{2,(i)}) \delta_{\mathbf{z}^{(i)}}(\mathbf{y}); \quad (4.2.36)$$

par conséquent, la forme du PHD v_n est la même que celle de v_{n-1} . Contrairement à l'implémentation classique par mélange de gaussiennes du PHD [121], v_n n'est pas un mélange gaussien au sens strict mais la somme d'un mélange gaussien pour les composantes correspondant aux non-détections et d'un mélange normal-discret pour les termes de détection. Cependant, $v_n(\mathbf{x}) = \int v_n(\mathbf{x}, \mathbf{y}) d\mathbf{y}$ est un mélange gaussien.

Remarque 4.3. Lorsque le modèle PMC considéré est aussi un modèle HMC et que l'hypothèse (4.2.10) sur $\gamma_n(\mathbf{x}, \mathbf{y})$ est valide, nous avons vu que le filtre PMC-PHD coïncidait avec le filtre PHD classique. Par conséquent, si les paramètres du modèle PMC correspondent avec celui du modèle HMC, c'est à dire si

$$\mathbf{B}_n = \begin{bmatrix} \mathbf{F}_n & \mathbf{0} \\ \mathbf{H}_n \mathbf{F}_n & \mathbf{0} \end{bmatrix}, \quad \Sigma_n = \begin{bmatrix} \mathbf{Q}_n & (\mathbf{H}_n \mathbf{Q}_n)^T \\ (\mathbf{H}_n \mathbf{Q}_n) & \mathbf{R}_n + \mathbf{H}_n \mathbf{Q}_n \mathbf{H}_n^T \end{bmatrix}, \quad (4.2.37)$$

et pour tout i , $1 \leq i \leq J_{\gamma_n}$,

$$\mathbf{m}_{\gamma_n}^{(i)} = \begin{bmatrix} \mathbf{m}_{\gamma_n}^{\mathbf{x},(i)} & \mathbf{H}_n \mathbf{m}_{\gamma_n}^{\mathbf{x},(i)} \end{bmatrix}^T, \quad \mathbf{P}_{\gamma_n}^{(i)} = \begin{bmatrix} \mathbf{P}_{\gamma_n}^{\mathbf{x},(i)} & (\mathbf{H}_n \mathbf{P}_{\gamma_n}^{\mathbf{x},(i)})^T \\ \mathbf{H}_n \mathbf{P}_{\gamma_n}^{\mathbf{x},(i)} & \mathbf{R}_n + \mathbf{H}_n \mathbf{P}_{\gamma_n}^{\mathbf{x},(i)} \mathbf{H}_n^T \end{bmatrix}, \quad (4.2.38)$$

(cf. Remarque 4.2) alors cette implémentation par mélange de gaussiennes du filtre PMC-PHD coïncide avec celle du filtre HMC-PHD [121], dans laquelle $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n)$, $g_n(\mathbf{y}_n | \mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n)$ et $\gamma_n(\mathbf{x}) = \sum_{i=1}^{J_{\gamma_n}} w_{\gamma_n}^{(i)} \mathcal{N}(\mathbf{x}; \mathbf{m}_{\gamma_n}^{(i)}; \mathbf{P}_{\gamma_n}^{(i)})$.

Comparé à l'implémentation par mélange de gaussiennes du filtre HMC-PHD, le coût supplémentaire induit par notre implémentation est négligeable car il est seulement dû aux matrices \mathbf{F}_n^2 et \mathbf{H}_n^2 qui, par exemple, ajoutent des termes supplémentaires dans (4.2.22). D'autre part, cette implémentation pour les modèles couple linéaires et gaussiens souffre des mêmes problèmes que l'implémentation classique : le nombre de composantes croît exponentiellement à cause de l'étape de mise à jour. Les techniques d'élagage et de limitation du nombre de gaussiennes sont bien connues et peuvent être utilisées ici : les composantes du mélange dont les poids $w_n^{1,(i)}$ et $w_n^{2,(i)}$ sont en dessous d'un seuil donné T_p sont supprimées ; nous pouvons également limiter le nombre de composantes dans $v_n^1(\mathbf{x}, \mathbf{y})$ et $v_n^2(\mathbf{x}, \mathbf{y})$, en gardant respectivement les J_{\max}^1 et J_{\max}^2 composantes de poids le plus élevé. Cependant, la nature de $v_n(\mathbf{x}, \mathbf{y})$ nous impose de modifier l'étape de fusion.

Étape de fusion

Afin de réduire le nombre de composantes d'un mélange gaussien, il est possible de fusionner les composantes proches les unes des autres [115] [121]. Néanmoins, dans notre cas, contrairement à l'étape de fusion classique proposée dans [121], nous ne devons pas fusionner les composantes de $v_n^1(\mathbf{x}, \mathbf{y})$, dues à des non-détections, avec celles de $v_n^2(\mathbf{x}, \mathbf{y})$, dues aux détections, car leur forme est différente : $v_n^1(\mathbf{x}, \mathbf{y})$ est un mélange gaussien tandis que $v_n^2(\mathbf{x}, \mathbf{y})$ est un mélange ND. Cet argument peut être illustré avec le scénario suivant. Supposons que $v_{n|n-1}(\mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{n|n-1}; \mathbf{P}_{n|n-1})$, $\kappa_n(\cdot) = 0$ et $\mathcal{Z}_n = \{\mathbf{z}\}$. Alors $v_n(\mathbf{x}, \mathbf{y}) = (1 - p_{d,n})\mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{n|n-1}; \mathbf{P}_{n|n-1}) + \mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_n(\mathbf{z}); \mathbf{P}_n)\delta_{\mathbf{z}}(\mathbf{y})$. Donc, si $\mathcal{N}(\mathbf{x}; \mathbf{m}_{n|n-1}^{\mathbf{x}}; \mathbf{P}_{n|n-1}^{\mathbf{x}})$ et $\mathcal{N}(\mathbf{x}; \mathbf{m}_n^{\mathbf{x}}; \mathbf{P}_n^{\mathbf{x}})$ sont assez proches, les deux gaussiennes sont fusionnées. L'état estimé est alors donné par $((1 - p_{d,n})\mathbf{m}_{n|n-1}^{\mathbf{x}} + \mathbf{m}_n^{\mathbf{x}}(\mathbf{z})) / (2 - p_{d,n})$. Cependant, nous sommes sûrs que la mesure \mathbf{z} est due à l'objet décrit par la gaussienne $\mathcal{N}(\mathbf{x}, \mathbf{y}; \mathbf{m}_{n|n-1}; \mathbf{P}_{k|k-1})$; il est par conséquent préférable d'estimer l'état caché de l'objet par $\mathbf{m}_n^{\mathbf{x}}(\mathbf{z})$.

Remarque 4.4. Dans le scénario décrit ci-dessus, un estimateur du nombre d'objets à l'instant n est $2 - p_{d,n}$, alors qu'un estimateur du nombre d'objets prédit est $\int v_{n|n-1}(\mathbf{x}, \mathbf{y})d\mathbf{x}d\mathbf{y} = 1$. De plus, une unique mesure est disponible à l'instant n (cette mesure ne peut pas être une mesure de fausse alarme car $\kappa_n = 0$). Par conséquent, un estimateur du nombre d'objet à l'instant n devrait être égal à 1 car cette mesure est inévitablement associée à l'état prédit $\mathbf{m}_{n|n-1}$. Ce problème est bien connu [46] et est dû à l'hypothèse **A.3** sur laquelle repose l'étape de mise à jour du filtre PHD [80]. Le filtre CPHD, qui ne repose plus sur cette hypothèse, corrige cette aberration.

Par conséquent, nous considérons deux étapes de fusion : une pour les composantes de $v_{n|n}^1(\mathbf{x}, \mathbf{y})$ et une pour celles de $v_{n|n}^2(\mathbf{x}, \mathbf{y})$. Plus précisément, pour $v_{n|n}^1(\mathbf{x}, \mathbf{y})$, nous fusionnons les gaussiennes (i) et (j) (de dimension $p + q$) si pour un seuil donné T_m , nous avons

$$(\mathbf{m}_n^{1,(i)} - \mathbf{m}_n^{1,(j)})^T (\mathbf{P}_n^{1,(i)})^{-1} (\mathbf{m}_n^{1,(i)} - \mathbf{m}_n^{1,(j)}) \leq T_m. \quad (4.2.39)$$

Enfin, deux composantes (i) et (j) de v_n^2 (de dimension p) sont fusionnées si $\mathbf{m}_n^{2,(i)}$, $\mathbf{m}_n^{2,(j)}$ et $\mathbf{P}_n^{2,(i)}$ vérifient (4.2.39) et $\mathbf{z}^{(i)} = \mathbf{z}^{(j)}$. En d'autres termes, nous fusionnons seulement les gaussiennes qui partagent la même observation \mathbf{z} .

Une fois les étapes d'élagage et de fusion exécutées, nous pouvons extraire les états cachés : une gaussienne dont le poids est au-dessus d'un certain seuil, typiquement 0.5, sera considérée comme un objet dont l'état est donné par la moyenne de la gaussienne.

4.2.4 Apport du PMC-PHD : simulations

Dans ce paragraphe, nous nous donnons des propriétés physiques utilisées dans des modèles HMC linéaires et gaussiens (4.1.7)-(4.1.8), puis nous considérons la classe des modèles couple associés donnée par la Proposition 4.1. Nous posons

$$\mathbf{F}_n = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{H}_n = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (4.2.40)$$

$$\mathbf{Q}_n = \begin{bmatrix} 100 & 1 & 0 & 0 \\ 1 & 10 & 0 & 0 \\ 0 & 0 & 100 & 1 \\ 0 & 0 & 1 & 10 \end{bmatrix}, \mathbf{R}_n = \begin{bmatrix} 10^2 & 0 \\ 0 & 10^2 \end{bmatrix}, \quad (4.2.41)$$

et nous générons des trajectoires suivant un modèle de la Proposition 4.1 où

$$\mathbf{F}_n^2 = \begin{bmatrix} a & 0 \\ 0 & 0 \\ 0 & b \\ 0 & 0 \end{bmatrix}, \mathbf{H}_n^2 = \begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix}. \quad (4.2.42)$$

Nous rappelons que dans ces modèles, les lois $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ de \mathbf{X}_n conditionnellement à $\mathbf{X}_{n-1} = \mathbf{x}_{n-1}$, et $p(\mathbf{y}_n|\mathbf{x}_n)$ de \mathbf{Y}_n conditionnellement à \mathbf{x}_n , coïncident respectivement avec $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n)$ et $g_n(\mathbf{y}_n|\mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n)$. Par conséquent, nous comparons un filtre PMC-PHD basé sur le modèle (4.1.43)-(4.1.48) à un filtre HMC-PHD classique basé sur les lois $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $g_n(\mathbf{y}_n|\mathbf{x}_n)$. Dans ce cas, les deux algorithmes partagent les mêmes propriétés physiques.

Nous considérons un scénario de 6 objets : les objets 1 et 2 naissent à $n = 1$, les objets 3 et 4 à $n = 20$ et les objets 5 et 6 à $n = 50$. Nous estimons les positions et vitesses des objets en coordonnées cartésiennes, $\mathbf{x}_n = [p_{x,n}, \dot{p}_{x,n}, p_{y,n}, \dot{p}_{y,n}]^T$.

Nous générons uniformément une moyenne de 20 mesures de fausses alarmes sur la région $V = [-2000, 2000] \times [-2000, 2000]$. L'intensité des mesures de fausses alarmes s'écrit $\kappa_n(\mathbf{z}) = 1.25 \times 10^{-6} \times \mathbb{1}_V(\mathbf{z})$. Nous fixons $p_{s,n} = 0.98$ et $p_{d,n} = 0.9$. Une réalisation d'un tel scénario est tracée sur la Figure 4.1.

Pour l'implémentation par mélange de gaussiennes du filtre HMC-PHD, l'intensité des naissances s'écrit

$$\gamma_n(\mathbf{x}) = 0.01 \sum_{i=1}^6 \mathcal{N}(\mathbf{x}; \mathbf{m}_{\gamma_n}^{(i)}; \mathbf{P}_{\gamma_n}),$$

où les moyennes $\mathbf{m}_{\gamma_n}^{(i)}$ correspondent aux zones où les objets sont susceptibles d'apparaître (par exemple $\mathbf{m}_{\gamma_n}^{(1)} = [200, 0, 500, 0]^T$) et $\mathbf{P}_{\gamma_n} = \text{diag}(1000, 400, 1000, 400)$; pour l'implémentation par mélange de gaussiennes du filtre PMC-PHD, nous choisissons

$$\gamma_n(\boldsymbol{\xi}) = 0.01 \sum_{i=1}^6 \mathcal{N}\left(\boldsymbol{\xi}; \begin{bmatrix} \mathbf{m}_{\gamma_n}^{(i)} \\ \mathbf{H}_n \mathbf{m}_{\gamma_n}^{(i)} \end{bmatrix}; \begin{bmatrix} \mathbf{P}_{\gamma_n} & (\mathbf{H}_n \mathbf{P}_{\gamma_n})^T \\ \mathbf{H}_n \mathbf{P}_{\gamma_n} & \mathbf{R}_n \mathbf{H}_n \mathbf{P}_{\gamma_n} \mathbf{H}_n^T \end{bmatrix}\right).$$

Les seuils d'élagage et de fusion sont respectivement $T_p = 10^{-5}$ et $T_m = 5$, nous gardons au plus $J_{\max} = 100$ gaussiennes pour l'implémentation HMC et $J_{\max}^1 = J_{\max}^2 = 50$ pour l'implémentation PMC. Une

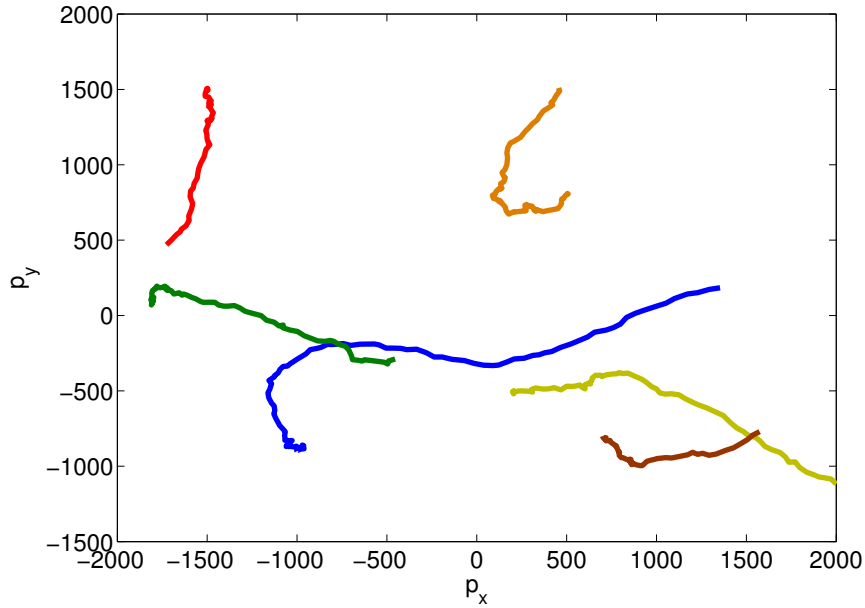


FIGURE 4.1 – Scénario de 6 objets évoluant suivant un modèle PMC linéaire et gaussien de la Proposition 4.1.

gaussienne est considérée comme un objet dès lors que son poids est supérieur à 0.5. D'autres valeurs de ces paramètres n'affectent pas les résultats.

Les algorithmes sont comparés en utilisant la distance OSPA (1.2.13) dans laquelle nous prenons $\bar{p} = 20$, $\bar{c} = 1$. Nous fixons $a = b = 0.7$ et $c = d = 0.1$ dans (4.2.42) et nous calculons à chaque instant la distance OSPA, moyennée sur 200 réalisations. Les résultats sont tracés sur la Figure 4.2. Nous observons que le filtre PMC-PHD est préférable au filtre PHD classique, bien que les deux algorithmes reposent sur les mêmes lois $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $p(\mathbf{y}_n|\mathbf{x}_n)$. Cependant, les hypothèses d'indépendances usuelles ne sont plus satisfaites, et les propriétés statistiques des modèles PMC ne sont pas prises en compte par l'implémentation reposant sur le modèle HMC. En effet, a , b , c et d sont différents de 0 ; par conséquent les propriétés **P.1-P.2** ne sont pas satisfaites. En particulier, la loi

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n^1 \mathbf{x}_{n-1} + \mathbf{F}_n^2 \mathbf{y}_{n-1}; \Sigma_n^{11}),$$

$$\Sigma_n^{11} = \begin{bmatrix} 49 & 1 & 0 & 0 \\ 1 & 10 & 0 & 0 \\ 0 & 0 & 49 & 1 \\ 0 & 0 & 1 & 10 \end{bmatrix} \quad (4.2.43)$$

est approchée, dans la formulation HMC, par $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n)$, où \mathbf{Q}_n est donné par (4.2.41). L'information apportée par l'observation \mathbf{y}_{n-1} n'est donc pas prise en compte dans l'implémentation classique : l'incertitude sur l'état \mathbf{x}_n , conditionnellement à \mathbf{x}_{n-1} et \mathbf{y}_{n-1} augmente. Une telle approximation a des conséquences sur les moyennes, les matrices de covariance et les poids du mélange, ce qui affecte les estimateurs des états cachés mais également l'estimateur du nombre d'objets. En effet, l'erreur de localisation (premier terme du membre de droite de (1.2.13)) moyennée sur le temps est approximativement de 3.14 pour l'implémentation PMC et de 5.68 pour l'implémentation classique ; l'erreur sur le nombre

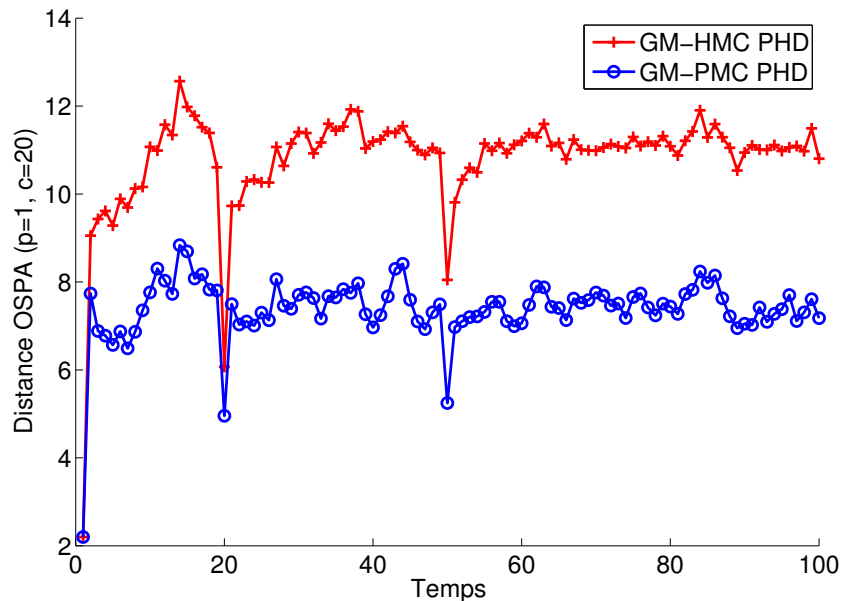


FIGURE 4.2 – Distance OSPA avec $p = 1$ et $c = 20$. Moyennée sur le temps, la distance OSPA est de 7.3837 pour le filtre PMC-PHD et de 10.9143 pour le filtre HMC-PHD.

d'objets estimés (second terme du membre de droite de (1.2.13)) moyennée sur le temps est de 4.23 pour l'implémentation PMC et de 5.12 pour l'implémentation classique.

La masse totale du PHD estimée par l'implémentation classique devient moins fiable au fur et à mesure du temps, comme le montre la Figure 4.3, sur laquelle nous avons tracé, à chaque instant, le rapport des écarts type de chaque estimateur de la masse totale.

Le temps de calcul est à peu près identique pour les deux implémentations (environ 0.65 seconde par itération), ce qui n'est pas surprenant puisque les deux filtres ne diffèrent que par les matrices \mathbf{F}_n^2 et \mathbf{H}_n^2 , nulles dans le cas HMC. Cette amélioration des performances requiert donc un coût de calcul supplémentaire négligeable.

Nous avons montré que l'implémentation classique du PHD n'est plus adaptée pour un tel scénario, bien que les deux implémentations partagent les lois $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $p(\mathbf{y}_n|\mathbf{x}_n)$. Dans le contexte des modèles HMC, une astuce pratique permettant de modéliser l'incertitude liée à la connaissance du modèle statistique consiste à augmenter la variance du bruit de processus. Pour le même scénario, nous implémentons donc le filtre PHD classique, en remplaçant \mathbf{Q} dans (4.2.41) par Σ_n^{11} dans (4.2.43), puis par $1.5\mathbf{Q}$. Les distances OSPA, moyennées sur le temps, sont maintenant égales à 10.9785 et 10.9092 ; cette astuce ne permet donc pas d'améliorer les performances de l'implémentation classique. Ceci est probablement dû au fait que dans ce nouveau modèle HMC, les propriétés physiques, données par $f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $g_n(\mathbf{y}_n|\mathbf{x}_n)$ ne coïncident plus avec les vraies propriétés, données par $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ et $p(\mathbf{y}_n|\mathbf{x}_n)$.

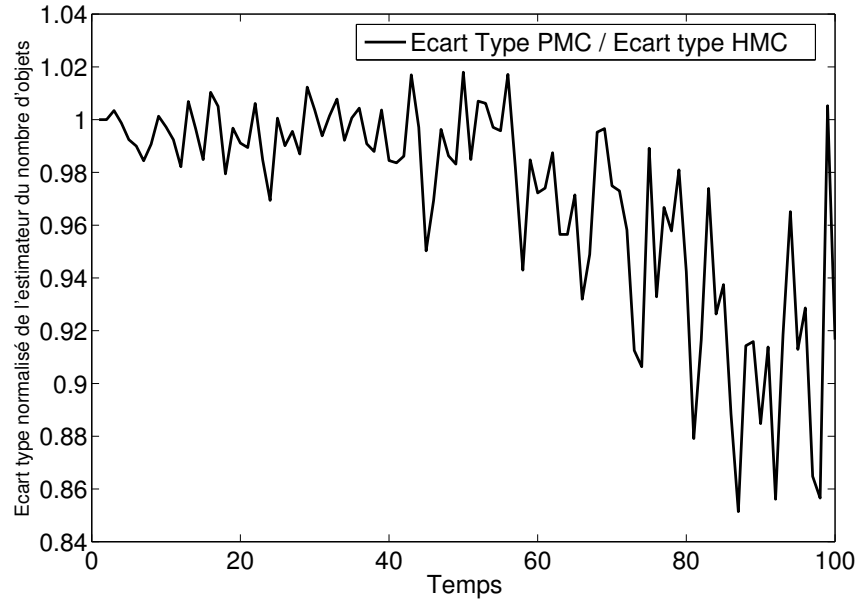


FIGURE 4.3 – Ecart type normalisé de l’estimateur du nombre d’objets associé au PMC-PHD par rapport à l’estimateur du nombre d’objets associé au HMC-PHD.

4.3 Une classe d’algorithmes de filtrage exact rapide pour des modèles de Markov à sauts

Dans cette section, nous nous plaçons dans le modèle à sauts linéaire et gaussien (4.0.3)-(4.0.4) et nous nous penchons sur le calcul de

$$\Theta_n = E(f(\mathbf{X}_n)|\mathbf{y}_{0:n}) = \int f(\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n})d\mathbf{x}_n = \sum_{r_n} p(r_n|\mathbf{y}_{0:n})E(\mathbf{X}_n|\mathbf{y}_{0:n}, r_n). \quad (4.3.1)$$

Comme nous l’avons rappelé, le calcul exact de Θ_n pour ces modèles est un problème NP-complet. Les solutions approchées incluent le FP, dont des améliorations ont été proposées tout au long des Chapitres 2 et 3, mais également des approximations numériques. Comme $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ est un mélange gaussien qui croît exponentiellement avec le temps, des approximations basées sur des techniques de fusion et d’élagage ont été étudiées [118] [115]. Une autre classe d’approximations numériques est donnée par les algorithmes de filtrage IMM [17] [85] [75]; dans cette classe d’approximations, plusieurs FK sont utilisés parallèlement (un FK pour chaque mode r_n); les estimateurs fournis par chaque filtre sont alors pondérés en fonction des paramètres du modèle et des observations disponibles.

Plutôt que de chercher à améliorer les approximations existantes afin d’estimer Θ_n , nous proposons dans cette section une approche différente. En effet, nous cherchons à approcher un modèle à sauts donné par un modèle triplet « assez proche », dans lequel le calcul de Θ_n est faisable exactement. En d’autres termes, plutôt que d’approcher l’estimateur dans le modèle originel, nous le calculons exactement et rapidement dans le modèle approché.

Nous devons à présent spécifier ce que nous entendons lorsque nous parlons de modèle triplet « assez proche » d’un modèle à sauts (4.0.3)-(4.0.4). Le succès des modèles à sauts provient en partie de leur capacité à modéliser les transitions de l’état caché \mathbf{X}_n en fonction du régime $R_n = r_n$ (par l’intermédiaire

de $f_{n|n-1}$), et les vraisemblances des observations \mathbf{Y}_n avec l'état caché \mathbf{X}_n , en fonction de $R_n = r_n$ (par l'intermédiaire de g_n). Cette capacité de modélisation ouvre un vaste champ d'applications [47][23]. D'autre part, ces propriétés physiques sont reliées aux propriétés statistiques suivantes,

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{r}_{0:n}) = f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}, r_n), \quad (4.3.2)$$

$$p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{r}_{0:n}) = g_n(\mathbf{y}_n|\mathbf{x}_n, r_n), \quad (4.3.3)$$

et sont donc intrinsèques au modèle (4.0.3)-(4.0.4). Par conséquent, nous allons rechercher une classe de modèles triplet dans laquelle les propriétés physiques, qui sont liées aux deux propriétés statistiques précédentes, restent valides mais dans laquelle le calcul exact de Θ_n est possible. En d'autres termes, à partir d'un JMSS donné (4.0.3)-(4.0.4), **dont nous noterons à présent la loi** $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$, nous cherchons une classe de modèles triplet décrit par $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ tels que

1. $p^2(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{r}_{0:i}) = f_{i|i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}, r_i)$;
2. $p^2(\mathbf{y}_i|\mathbf{x}_i, \mathbf{r}_{0:i}) = g_i(\mathbf{y}_i|\mathbf{x}_i, r_i)$; et
3. Θ_n peut être calculé exactement avec un coût linéaire en nombre d'observations.

Pour aboutir à cette classe de modèles couple, notre approche procède en plusieurs étapes. Dans un premier temps, nous considérons les modèles de Markov couple étudiés dans le Paragraphe 4.1.5, exemple 3. Nous avons vu que dans ces modèles, les propriétés physiques d'un modèle HMC linéaire et gaussien pouvaient être conservées (voir la Proposition 4.1). Dans ces modèles, nous réintroduisons le processus $\mathbf{R}_{0:n}$ afin d'obtenir une classe de modèles triplet qui conservent les propriétés physiques du modèle à sauts linéaire et gaussien $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$, ainsi que des propriétés statistiques très proches. Dans cette classe de modèles, en utilisant les résultats du Paragraphe 4.1.7, nous cherchons ceux pour lesquels le calcul exact de Θ_n est possible. Par conséquent, cette méthode d'approximation est avant tout valide pour une classe générale de modèles physiquement contraints, qui inclut le modèle à sauts linéaire et gaussien $p^1(\cdot)$. Nous discutons enfin de la proximité des modèles obtenus avec le modèle à sauts originel. Dans le Paragraphe 4.3.4, nous illustrons notre méthodologie pas à pas sur un exemple pratique, puis nous effectuons des simulations dans des modèles à sauts linéaires et gaussiens et en particulier, notre méthode est comparée aux techniques d'approximations classiques incluant le FP [44] et l'algorithme IMM [17].

4.3.1 Une classe particulière de modèles de Markov triplet

Dans un premier temps, nous recherchons une classe de modèles de Markov triplet $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$, vérifiant les mêmes propriétés physiques que le modèle à sauts classique $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$, mais dans lesquels les propriétés statistiques sont plus générales. Étant donné un triplet $(\mathbf{X}_{0:n}, \mathbf{Y}_{0:n}, \mathbf{R}_{0:n})$, nous cherchons donc à construire une loi de probabilité $p^2(\boldsymbol{\xi}_{0:n}, \mathbf{r}_{0:n})$, décrite par les transitions $p^2(\mathbf{x}_i, \mathbf{y}_i, r_i|\mathbf{x}_{i-1}, \mathbf{y}_{i-1}, r_{i-1})$ avec les contraintes ci-dessous, permettant de conserver les propriétés physiques associées à la loi $p^1(\boldsymbol{\xi}_{0:n}, \mathbf{r}_{0:n})$:

- $p^2(r_n|\mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1}) = p^1(r_n|r_{n-1})$; cette condition assure que le processus discret $\{R_n\}_{n \geq 0}$ est une chaîne de Markov dont les transitions sont données par $p^2(r_n|r_{n-1}) = p^1(r_n|r_{n-1})$;
- conditionnellement à $\mathbf{r}_{0:n}$, les propriétés physiques du modèle $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ sont conservées, c'est à dire $p^2(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{r}_{0:n}) = f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}, r_n)$ et $p^2(\mathbf{y}_n|\mathbf{x}_n, \mathbf{r}_{0:n}) = g_n(\mathbf{y}_n|\mathbf{x}_n, r_n)$.

Par extension de la preuve de la Proposition 4.1, le résultat suivant fournit une classe de modèles de Markov triplet vérifiant les contraintes ci-dessus.

Proposition 4.5. *Considérons le modèle à sauts linéaire et gaussien $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ décrit par*

$$p^1(\mathbf{x}_0|r_0) = \mathcal{N}(\mathbf{x}_0; \mathbf{m}_0(r_0); \mathbf{P}_0(r_0)), \quad (4.3.4)$$

$$f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}, r_n) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n(r_n)\mathbf{x}_{n-1}; \mathbf{Q}_n(r_n)) \text{ et} \quad (4.3.5)$$

$$g_n(\mathbf{y}_n|\mathbf{x}_n, r_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n(r_n)\mathbf{x}_n; \mathbf{R}_n(r_n)), \quad (4.3.6)$$

pour tout n . Alors les modèles triplet linéaires et gaussiens $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ vérifiant

$$p^2(r_n|r_{n-1}, \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) = p^1(r_n|r_{n-1}), \quad (4.3.7)$$

$$p^2(\mathbf{x}_0|r_0) = p^1(\mathbf{x}_0|r_0), \quad (4.3.8)$$

$$p^2(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{r}_{0:n}) = f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}, r_n), \quad (4.3.9)$$

$$p^2(\mathbf{y}_n|\mathbf{x}_n, \mathbf{r}_{0:n}) = g_n(\mathbf{y}_n|\mathbf{x}_n, r_n), \quad (4.3.10)$$

sont décrits par

$$p(\boldsymbol{\xi}_0|r_0) = \mathcal{N}\left(\boldsymbol{\xi}_0; \begin{bmatrix} \mathbf{m}_0(r_0) \\ \mathbf{H}_0(r_0)\mathbf{m}_0(r_0) \end{bmatrix}; \begin{bmatrix} \mathbf{P}_0(r_0) & (\mathbf{H}_0(r_0)\mathbf{P}_0(r_0))^T \\ \mathbf{H}_0(r_0)\mathbf{P}_0(r_0) & \mathbf{R}_0(r_0) + \mathbf{H}_0(r_0)\mathbf{P}_0(r_0)\mathbf{H}_0(r_0)^T \end{bmatrix}\right), \quad (4.3.11)$$

$$p^2(\boldsymbol{\xi}_n|\boldsymbol{\xi}_{n-1}, \mathbf{r}_{n-1:n}) = \mathcal{N}(\boldsymbol{\xi}_n; \mathbf{B}_n(\mathbf{r}_{n-1:n})\boldsymbol{\xi}_{n-1}; \boldsymbol{\Sigma}_n(\mathbf{r}_{n-1:n})), \quad (4.3.12)$$

où les matrices $\mathbf{B}_n(\mathbf{r}_{n-1:n})$ et $\boldsymbol{\Sigma}_n(\mathbf{r}_{n-1:n})$ sont définies par

$$\mathbf{B}_n(\mathbf{r}_{n-1:n}) = \begin{bmatrix} \mathbf{F}_n(r_n) - \mathbf{F}_n^2(\mathbf{r}_{n-1:n})\mathbf{H}_{n-1}(r_{n-1}) & \mathbf{F}_n^2(\mathbf{r}_{n-1:n}) \\ \mathbf{H}_n(r_n)\mathbf{F}_n(r_n) - \mathbf{H}_n^2(\mathbf{r}_{n-1:n})\mathbf{H}_{n-1}(r_{n-1}) & \mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \end{bmatrix}, \quad (4.3.13)$$

$$\boldsymbol{\Sigma}_n(\mathbf{r}_{n-1:n}) = \begin{bmatrix} \boldsymbol{\Sigma}_n^{11}(\mathbf{r}_{n-1:n}) & (\boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n}))^T \\ \boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n}) & \boldsymbol{\Sigma}_n^{22}(\mathbf{r}_{n-1:n}) \end{bmatrix}, \quad (4.3.14)$$

$$\boldsymbol{\Sigma}_n^{11}(\mathbf{r}_{n-1:n}) = \mathbf{Q}_n(r_n) - \mathbf{F}_n^2(\mathbf{r}_{n-1:n})\mathbf{R}_{n-1}(r_{n-1})(\mathbf{F}_n^2(\mathbf{r}_{n-1:n}))^T, \quad (4.3.15)$$

$$\boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n}) = \mathbf{H}_n(r_n)\mathbf{Q}_n(r_n) - \mathbf{H}_n^2(\mathbf{r}_{n-1:n})\mathbf{R}_{n-1}(r_{n-1})(\mathbf{F}_n^2(\mathbf{r}_{n-1:n}))^T, \quad (4.3.16)$$

$$\boldsymbol{\Sigma}_n^{22}(\mathbf{r}_{n-1:n}) = \mathbf{R}_n(r_n) - \mathbf{H}_n^2(\mathbf{r}_{n-1:n})\mathbf{R}_{n-1}(r_{n-1})(\mathbf{H}_n^2(\mathbf{r}_{n-1:n}))^T + \mathbf{H}_n(r_n)\mathbf{Q}_n(r_n)(\mathbf{H}_n(r_n))^T, \quad (4.3.17)$$

et où les matrices $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$ et $\mathbf{H}_n^2(\mathbf{r}_{n-1:n})$ peuvent être choisies librement à condition que $\boldsymbol{\Sigma}_n(\mathbf{r}_{n-1:n})$ soit une matrice de covariance définie positive, quel que soit $\mathbf{r}_{n-1:n}$.

Par extension directe des Propositions 4.1 à 4.3, la proposition suivante compare la classe de modèles triplet de la Proposition 4.5 au modèle à sauts linéaire et gaussien $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$.

Proposition 4.6. *Dans la classe de modèles de la Proposition 4.5, les lois $p^2(\boldsymbol{\xi}_n|\mathbf{x}_{n-1}, \mathbf{r}_{n-1:n})$ et $p^2(\mathbf{x}_{0:n}, \mathbf{r}_{0:n})$ sont indépendantes du choix de $\{\mathbf{F}_n^2(\mathbf{r}_{n-1:n}), \mathbf{H}_n^2(\mathbf{r}_{n-1:n})\}_{n \geq 1}$,*

$$p^2(\boldsymbol{\xi}_n|\mathbf{x}_{n-1}, \mathbf{r}_{n-1:n}) = p^1(\boldsymbol{\xi}_n|\mathbf{x}_{n-1}, \mathbf{r}_{n-1:n}) = f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}, r_n)g_n(\mathbf{y}_n|\mathbf{x}_n, r_n), \quad (4.3.18)$$

$$p^2(\mathbf{x}_{0:n}, \mathbf{r}_{0:n}) = p^1(\mathbf{x}_{0:n}, \mathbf{r}_{0:n}) = p^1(r_0) \prod_{i=1}^n p^1(r_i|r_{i-1})p^1(\mathbf{x}_0|\mathbf{r}_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i|\mathbf{x}_{i-1}, r_i). \quad (4.3.19)$$

Conditionnellement à $\mathbf{r}_{0:n}$, $\{\mathbf{X}_n\}_{n \geq 0}$ est donc une chaîne de Markov dont les transitions sont données par $\{f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}, r_n)\}_{n \geq 1}$. La différence entre $p^1(\cdot)$ et $p^2(\cdot)$ provient de la loi $p^2(\mathbf{y}_{0:n}|\mathbf{x}_{0:n}, \mathbf{r}_{0:n})$; nous avons donc

$$p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n}) = p^1(\mathbf{r}_{0:n})p^1(\mathbf{x}_{0:n}|\mathbf{r}_{0:n})p^2(\mathbf{y}_{0:n}|\mathbf{x}_{0:n}, \mathbf{r}_{0:n}), \quad (4.3.20)$$

$$p^2(\mathbf{y}_{0:n}|\mathbf{x}_{0:n}, \mathbf{r}_{0:n}) = p(\mathbf{y}_0|\mathbf{x}_{0:n}, \mathbf{r}_{0:n}) \prod_{i=1}^n p(\mathbf{y}_i|\mathbf{x}_{i-1:n}, \mathbf{y}_{i-1}, \mathbf{r}_{i-1:n}). \quad (4.3.21)$$

4.3.2 Une solution approchée du problème de filtrage

Nous nous intéressons maintenant au problème du filtrage, au sens du calcul de Θ_n , dans la classe de modèles de la Proposition 4.5. Bien sûr, le calcul exact de Θ_n n'est pas toujours possible dans cette classe de modèles qui englobe le modèle classique $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ ($\mathbf{F}_n^2(\mathbf{r}_{n-1:n}) = \mathbf{0}$, $\mathbf{H}_n^2(\mathbf{r}_{n-1:n}) = \mathbf{0}$). De façon générale, notons que la transition du triplet $\{\mathbf{X}_n, \mathbf{Y}_n, R_n\}$ markovien s'écrit ici

$$p^2(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1}) = p^1(r_n | r_{n-1}) p^2(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}, \mathbf{r}_{n-1:n}), \quad (4.3.22)$$

où $p^2(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}, \mathbf{r}_{n-1:n})$ est une gaussienne dont la moyenne est linéaire en $\boldsymbol{\xi}_{n-1}$ et est donnée par (4.3.12). Par conséquent, le FP marginalisé présenté dans le Paragraphe 4.1.7 peut s'appliquer pour cette classe de modèles.

Néanmoins, cette classe de modèles étant paramétrée par $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$ et $\mathbf{H}_n^2(\mathbf{r}_{n-1:n})$, il est légitime de se demander si un choix particulier de ces paramètres ne permet pas le calcul exact de Θ_n en se plaçant dans les conditions de la Proposition 4.4. Rappelons que cette proposition met en avant la forme des lois $p^2(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1:n}, \mathbf{r}_{n-1:n})$ et $p^2(\mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1})$. La forme de la loi $p^2(\boldsymbol{\xi}_n | \boldsymbol{\xi}_{n-1}, \mathbf{r}_{n-1:n})$ donnée par (4.3.12) assure, par conditionnement, que la loi $p^2(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1:n}, \mathbf{r}_{n-1:n})$ vérifie la condition (4.1.61). D'après la factorisation (4.3.22), la forme de la loi $p^2(\mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1})$ s'écrit ici

$$p^2(\mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1}) = p^1(r_n | r_{n-1}) p^2(\mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{r}_{n-1:n}). \quad (4.3.23)$$

Par conséquent, $p^2(\mathbf{x}_n, \mathbf{y}_n, r_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, r_{n-1})$ vérifie la condition (4.1.60) si et seulement si

$$p^2(\mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{r}_{n-1:n}) = p^2(\mathbf{y}_n | \mathbf{y}_{n-1}, \mathbf{r}_{n-1:n}). \quad (4.3.24)$$

Or dans les modèles de la Proposition 4.5, la loi $p^2(\mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{r}_{n-1:n})$ dépend de \mathbf{x}_{n-1} à travers sa moyenne qui s'écrit

$$\left(\mathbf{H}_n(r_n) \mathbf{F}_n(r_n) - \mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \mathbf{H}_{n-1}(r_{n-1}) \right) \mathbf{x}_{n-1} + \mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \mathbf{y}_{n-1}. \quad (4.3.25)$$

Par conséquent, nous avons le résultat suivant :

Proposition 4.7. *Soit un modèle triplet appartenant à la classe de modèles de la Proposition 4.5. S'il existe pour tout $\mathbf{r}_{n-1:n}$ une matrice $\mathbf{H}_n^2(\mathbf{r}_{n-1:n})$ vérifiant*

$$\mathbf{H}_n(r_n) \mathbf{F}_n(r_n) - \mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \mathbf{H}_{n-1}(r_{n-1}) = \mathbf{0}, \quad (4.3.26)$$

alors le calcul récursif et exact de Θ_n est faisable avec un coût linéaire en nombre d'observations et est donné par les équations (4.1.62)-(4.1.67) dans lesquelles

$$\mathbf{C}_n(\mathbf{r}_{n-1:n}) = \mathbf{F}_n(r_n) - \mathbf{F}_n^2(\mathbf{r}_{n-1:n}) \mathbf{H}_{n-1}(r_{n-1}), \quad (4.3.27)$$

$$\mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n}) = \mathbf{F}_n^2(\mathbf{r}_{n-1:n}) \mathbf{y}_n + (\boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n}))^T (\boldsymbol{\Sigma}_n^{22}(\mathbf{r}_{n-1:n}))^{-1} (\mathbf{y}_n - \mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \mathbf{y}_{n-1}), \quad (4.3.28)$$

$$\boldsymbol{\Sigma}_n^{\mathbf{x}}(\mathbf{r}_{n-1:n}) = \boldsymbol{\Sigma}_n^{11}(\mathbf{r}_{n-1:n}) - (\boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n}))^T (\boldsymbol{\Sigma}_n^{22}(\mathbf{r}_{n-1:n}))^{-1} \boldsymbol{\Sigma}_n^{21}(\mathbf{r}_{n-1:n}), \quad (4.3.29)$$

$$p^2(\mathbf{y}_n | \mathbf{y}_{n-1}, r_{n-1}) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \mathbf{y}_{n-1}; \boldsymbol{\Sigma}_n^{22}(\mathbf{r}_{n-1:n})). \quad (4.3.30)$$

Démonstration. Dans les modèles de la Proposition 4.5, le filtrage exact est possible si $p^2(\mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{r}_{n-1:n})$ est indépendant de \mathbf{x}_{n-1} , ce qui est équivalent à la condition (4.3.26). Enfin, par application des résultats sur les conditionnements de vecteurs gaussiens en Annexe A, nous avons

$$p^2(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1:n}, \mathbf{r}_{n-1:n}) = \mathcal{N}(\mathbf{x}_n; \mathbf{C}_n(r_{n-1:n})\mathbf{x}_{n-1} + \mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n}); \Sigma_n^{\mathbf{x}}(\mathbf{r}_{n-1:n}))$$

où $\mathbf{C}_n(r_{n-1:n})$, $\mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n})$ et $\Sigma_n^{\mathbf{x}}(\mathbf{r}_{n-1:n})$ sont données par (4.3.27)-(4.3.29). \square

Le résultat précédent suggère une nouvelle méthode d'approximation de Θ_n dans des modèles de Markov à sauts. En effet, la Proposition 4.5 fournit une classe de modèles triplet qui possèdent tous les mêmes propriétés physiques. Parmi les modèles de cette classe, nous cherchons ceux pour lesquels la condition (4.3.26) est vérifiée, c'est à dire ceux pour lesquels le calcul exact de Θ_n est possible. Ces modèles sont alors déduits de la Proposition 4.7. L'Algorithme 4.3, composé de quatre étapes, résume cette nouvelle méthode de calcul d'un moment Θ_n dans un modèles à sauts physiquement contraint.

Algorithme 4.3: Algorithme approché de filtrage exact

Entrées : $\mathbf{F}_n(r_n)$, $\mathbf{H}_n(r_n)$, $\mathbf{Q}_n(r_n)$ et $\mathbf{R}_n(r_n)$, $p(r_{n-1} | \mathbf{y}_{0:n-1})$, $E(\mathbf{X}_{n-1} | \mathbf{y}_{0:n-1}, r_{n-1})$ et $E(\mathbf{X}_{n-1} \mathbf{X}_{n-1}^T | \mathbf{y}_{0:n-1}, r_{n-1})$

Sorties : $p(r_n | \mathbf{y}_{0:n})$, $E(\mathbf{X}_n | \mathbf{y}_{0:n}, r_n)$, $E(\mathbf{X}_n \mathbf{X}_n^T | \mathbf{y}_{0:n}, r_n)$ et Θ_n

pour $r_n = 1, \dots, K$ **faire**

pour $r_{n-1} = 1, \dots, K$ **faire**

E.1 Dédire de la Proposition 4.5 la classe des modèles de Markov triplet qui conservent les propriétés physiques des modèles à sauts (4.0.3)-(4.0.4) et qui sont paramétrés par

$\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$, $\mathbf{H}_n^2(\mathbf{r}_{n-1:n})$;

E.2 Choisir $\mathbf{H}_n^2(\mathbf{r}_{n-1:n})$ tel que $\mathbf{H}_n(r_n)\mathbf{F}_n(r_n) - \mathbf{H}_n^2(\mathbf{r}_{n-1:n})\mathbf{H}_{n-1}(r_{n-1}) = \mathbf{0}$;

E.3 Dédire $\mathbf{C}_n(r_{n-1:n})$, $\mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n})$ et $\Sigma_n^{\mathbf{x}}(\mathbf{r}_{n-1:n})$ en utilisant (4.3.27)-(4.3.29);

fin

 Calculer $p(r_n | \mathbf{y}_{0:n})$, $E(\mathbf{X}_n | \mathbf{y}_{0:n}, r_n)$, $E(\mathbf{X}_n \mathbf{X}_n^T | \mathbf{y}_{0:n}, r_n)$ en utilisant (4.1.63)-(4.1.67);

fin

E.4 Dédire $\Theta_n = \sum_{r_n} p(r_n | \mathbf{y}_{0:n}) E(f(\mathbf{X}_n) | \mathbf{y}_{0:n}, r_n)$ avec $f(\mathbf{X}_n) = \mathbf{X}_n$ ou $f(\mathbf{X}_n) = \mathbf{X}_n \mathbf{X}_n^T$.

Notons que l'Algorithme 4.3 n'impose aucune condition ne porte sur le choix de $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$. Il est néanmoins évident que les performances de l'estimateur approché qui découle de cette méthode d'approximation dépendent du choix de $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$, ce dont nous discutons maintenant.

4.3.3 Discussion

Une des applications du résultat précédent concerne le problème d'estimation dans le modèle à sauts $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$. Plutôt que d'essayer d'améliorer les approximations d'un estimateur de Θ_n dans $p^1(\cdot)$, le paragraphe précédent suggère naturellement de commencer par approcher $p^1(\cdot)$ par $p^2(\cdot)$, puis de calculer Θ_n exactement et rapidement sous le modèle $p^2(\cdot)$.

Remarque 4.5. Comme nous l'avons dit précédemment, la classe d'approximations obtenues s'applique à une très grande variété de modèles à sauts qui sont caractérisés par la Proposition 4.5; cette technique d'approximation ne s'applique donc pas exclusivement aux JMSS linéaires et Gaussiens. Il est par ailleurs possible d'ajuster le paramètre $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$ afin de s'adapter aux données traitées qui suivent éventuellement un modèle différent du JMSS. En d'autres termes, si les données suivent un modèle statistique plus complexe

que le modèle à sauts $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ mais dans lequel les propriétés physiques sont identiques, notre méthode d'approximation reste utilisable. Ce point sera abordé dans nos simulations, au Paragraphe 4.3.4.

Nous supposons maintenant que les données traitées suivent le modèle à sauts $p^1(\cdot)$ défini par les lois (4.0.3)-(4.0.4) et nous nous demandons comment choisir $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$. Rappelons que le modèle à sauts classique $p^1(\cdot)$ est un modèle particulier de la classe des modèles de la Proposition 4.5, dans lequel $\mathbf{F}_n^2(\mathbf{r}_{n-1:n}) = \mathbf{0}$ et $\mathbf{H}_n^2(\mathbf{r}_{n-1:n}) = \mathbf{0}$. Néanmoins, cela ne doit pas être notre choix car dans la classe des modèles dans laquelle le filtrage exact est possible, $\mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \neq \mathbf{0}$. L'idée est donc d'ajuster le paramètre $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$ de façon à compenser l'approximation induite par le choix de $\mathbf{H}_n^2(\mathbf{r}_{n-1:n})$ dans (4.3.26).

Notre critère d'optimisation consiste à minimiser la DKL $D_{\text{KL}}(p^2(\xi_{0:n}, \mathbf{r}_{0:n}), p^1(\xi_{0:n}, \mathbf{r}_{0:n}))$, en tant que fonction de $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$, entre la loi $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ de la Proposition 4.7 (c'est à dire la classe des modèles de la Proposition 4.5, dans lesquels le filtrage exact est possible) et la loi $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$. Nous rappelons que la DKL entre deux lois $p(\mathbf{x}, \mathbf{r})$ et $q(\mathbf{x}, \mathbf{r})$ s'écrit

$$D_{\text{KL}}(p, q) = \sum_{\mathbf{r}} \int p(\mathbf{x}, \mathbf{r}) \log \left(\frac{p(\mathbf{x}, \mathbf{r})}{q(\mathbf{x}, \mathbf{r})} \right) d\mathbf{x}. \quad (4.3.31)$$

Nous avons la Proposition suivante, dont une démonstration est présentée en Annexe C.4.

Proposition 4.8. *Soit un modèle triplet de la Proposition 4.7, décrit par la loi $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ et paramétrée par $\{\mathbf{F}_n^2(\mathbf{r}_{n-1:n})\}_{n \geq 0}$. La DKL entre les lois $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ et $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ est minimum si et seulement si pour tout i , $1 \leq i \leq n$,*

$$p^2(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{y}_{i-1:i}, \mathbf{r}_{i-1:i}) = p^2(\mathbf{x}_i | \mathbf{x}_{i-1}, \mathbf{y}_i, \mathbf{r}_{i-1:i}) \quad (4.3.32)$$

c'est à dire si et seulement si

$$\mathbf{F}_i^{2,\text{opt}}(\mathbf{r}_{i-1:i}) = \mathbf{Q}_i(r_i) \mathbf{H}_i(r_i)^T \left[\mathbf{R}_i(r_i) + \mathbf{H}_i(r_i) \mathbf{Q}_i(r_i) \mathbf{H}_i(r_i)^T \right]^{-1} \mathbf{H}_i^2(\mathbf{r}_{i-1:i}). \quad (4.3.33)$$

Par conséquent, si le problème consiste à approcher au mieux le modèle (4.0.3)-(4.0.4), les paramètres $\{\mathbf{F}_i^{2,\text{opt}}(\mathbf{r}_{i-1:i})\}_{i=1}^n$ sont optimaux, lorsque les matrices $\{\mathbf{H}_i^2(\mathbf{r}_{i-1:i})\}_{i=1}^n$ vérifient (4.3.26), au sens de la minimisation de la DKL.

Remarque 4.6. Une autre approche visant à approximer le modèle $p^1(\cdot)$ a été proposée. En effet, l'article [37] se propose dans un premier temps de transformer le modèle $p^1(\cdot)$ en un modèle dans lequel la loi $p(\mathbf{x}_n, \mathbf{y}_n)$ est un mélange dont le nombre de composantes reste fixe au cours du temps, puis d'approcher ce dernier modèle par un modèle dans lequel le filtrage exact est possible.

4.3.4 Analyses de performances et simulations

Dans ce paragraphe, nous commençons par analyser les conséquences de l'approximation d'un modèle donné $p^1(\cdot)$ par un modèle donné $p^2(\cdot)$ vérifiant (4.3.26) et (4.3.33), en fonction des paramètres du modèle $p^1(\cdot)$. Ensuite, nous comparons notre technique de filtrage pour des modèles à sauts linéaires et gaussiens avec les algorithmes de FP et IMM sur des simulations. Enfin, nous nous focalisons sur des modèles plus généraux que les modèles JMSS (4.0.2) dans lesquels nous comparons les différentes techniques d'approximations. Pour chaque simulation, nous générons 200 jeux de données de longueur $T = 100$.

Illustration pas à pas

Nous décrivons ici notre méthodologie pas à pas pour un modèle à sauts dans lequel $\dim(\mathbf{X}_n) = \dim(\mathbf{Y}_n) = 1$ (voir par exemple [47][23]) :

$$f_{n|n-1}(x_n|x_{n-1}, r_n) = \mathcal{N}(x_n; a(r_n)x_{n-1}; Q(r_n)), \quad (4.3.34)$$

$$g_n(y_n|x_n, r_n) = \mathcal{N}(y_n; b(r_n)x_n; R(r_n)), \quad (4.3.35)$$

où $|a(r_n)| \leq 1$ et $\{R_n\}_{n \geq 0}$ est une chaîne de Markov dont les transitions sont données par $p^1(r_n|r_{n-1})$. Dans un premier temps, nous cherchons à comparer les modèles à sauts de la Proposition 4.7 vérifiant la condition (4.3.33) avec le modèle à sauts linéaire et gaussien correspondant $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}, \mathbf{r}_{0:n})$, lorsque les sauts sont connus (la loi des sauts est identique dans les deux modèles). Nous nous focalisons donc sur le modèle HMC sous-jacent décrit par

$$f_{n|n-1}(x_n|x_{n-1}) = \mathcal{N}(x_n; ax_{n-1}; Q), \quad (4.3.36)$$

$$g_n(y_n|x_n) = \mathcal{N}(y_n; bx_n; R), \quad (4.3.37)$$

où $|a| \leq 1$. Par ailleurs, les modèles de la Proposition 4.5 sont réduits aux modèles de la Proposition 4.1 lorsque les sauts $\mathbf{r}_{0:n}$ sont connus. Par conséquent, nous commençons par construire la classe de modèles de la Proposition 4.1. Ces modèles couple sont décrits par une transition paramétrée par $F_n^2 = c$ et $H_n^2 = d$,

$$p^2(\boldsymbol{\xi}_n|\boldsymbol{\xi}_{n-1}) = \mathcal{N}(\boldsymbol{\xi}_n; \begin{bmatrix} a - bc & c \\ ab - db & d \end{bmatrix} \boldsymbol{\xi}_{n-1}; \begin{bmatrix} Q - c^2R & bQ - cdR \\ bQ - cdR & R(1 - d^2) + b^2Q \end{bmatrix}). \quad (4.3.38)$$

Lorsque les sauts $\mathbf{r}_{0:n}$ sont fixés, la condition (4.3.26) devient

$$\mathbf{H}_n \mathbf{F}_n - \mathbf{H}_n^2 \mathbf{H}_{n-1} = \mathbf{0}; \quad (4.3.39)$$

nous cherchons donc d tel que $ab - db = 0$, c'est à dire $d = a$. D'après la condition (4.3.33) le paramètre c qui permet de minimiser la DKL entre $p^1(\boldsymbol{\xi}_{0:n})$ et $p^2(\boldsymbol{\xi}_{0:n})$ est donné par $c = \frac{abQ}{R+b^2Q}$; par conséquent, parmi toute la classe de modèles (4.3.38), nous choisissons celui dont la transition vérifie

$$p_{n|n-1}^2(\boldsymbol{\xi}_n|\boldsymbol{\xi}_{n-1}) = \mathcal{N} \left(\boldsymbol{\xi}_n; \begin{bmatrix} a - \frac{ab^2Q}{R+b^2Q} & \frac{abQ}{R+b^2Q} \\ 0 & a \end{bmatrix} \boldsymbol{\xi}_{n-1}; \begin{bmatrix} Q - \frac{a^2b^2Q^2R}{(R+b^2Q)^2} & bQ - \frac{a^2bQR}{R+b^2Q} \\ bQ - \frac{a^2bQR}{R+b^2Q} & R(1 - a^2) + b^2Q \end{bmatrix} \right). \quad (4.3.40)$$

Nous vérifions facilement que la matrice de covariance ci dessus est définie positive, quels que soient $-1 \leq a \leq 1$, $b, Q \geq 0$ et $R \geq 0$. Nous comparons maintenant la DKL entre $p_{n|n-1}^2(\boldsymbol{\xi}_n|\boldsymbol{\xi}_{n-1})$ et $p_{n|n-1}^1(\boldsymbol{\xi}_n|\boldsymbol{\xi}_{n-1})$ qui coïncide ici avec la DKL entre $p^2(y_n|y_{n-1})$ et $p^1(y_n|x_{n-1})$, d'après la preuve de la Proposition 4.8. En effet, le choix de d implique $p^2(y_n|x_{n-1}, y_{n-1}) = p^2(y_n|y_{n-1})$ (voir (4.3.24)) et le choix de c implique $p^2(x_n|x_{n-1}, y_{n-1}, y_n) = p^1(x_n|x_{n-1}, y_n)$ (voir (4.3.32)). Dans le modèle (4.3.36)-(4.3.37), $p^1(y_n|x_{n-1}) = \mathcal{N}(y_n; abx_{n-1}; b^2Q + R)$ et dans le modèle (4.3.40), $p^2(y_n|y_{n-1}) = \mathcal{N}(y_n; ay_{n-1}; R(1 - a^2) + b^2Q)$. Par conséquent, en utilisant les résultats classiques sur la DKL entre deux gaussiennes [105], nous avons

$$\text{DKL}(p^2(y_n|y_{n-1}), p^1(y_n|x_{n-1})) = 0.5 \left[-\frac{a^2R}{R + b^2Q} + \frac{a^2(y_{n-1} - bx_{n-1})^2}{R + b^2Q} - \ln\left(\frac{R + b^2Q - a^2R}{R + b^2Q}\right) \right], \quad (4.3.41)$$

qui dépend de y_{n-1} et x_{n-1} par l'intermédiaire de $(y_{n-1} - bx_{n-1})^2$. Cependant, dans ces modèles, $E((Y_{n-1} - bX_{n-1})^2) = R$, donc

$$E(D_{\text{KL}}(p^2(y_n|y_{n-1}), p^1(y_n|x_{n-1}))) = -0.5 \ln\left(1 - \frac{a^2(R/Q)}{R/Q + b^2}\right). \quad (4.3.42)$$

Interprétons ce résultat. $E(D_{\text{KL}}(p^2(y_n|y_{n-1}), p^1(y_n|x_{n-1})))$ est une fonction croissante du rapport R/Q . Par conséquent, lorsque R/Q est petit, c'est à dire lorsque le bruit de processus est important comparé au bruit d'observation, la DKL est faible. Le modèle (4.3.40) est donc proche du modèle originel (4.3.36)-(4.3.37) et l'estimation des données cachées à partir du modèle (4.3.40), bien qu'elles suivent (4.3.36)-(4.3.37), ne devrait pas avoir de conséquences importantes.

Pour le vérifier, nous générons des données suivant le modèle (4.3.36)-(4.3.37) où nous prenons $a = b = R = 1$. Nous estimons les états cachés à partir du modèle (4.3.40) en utilisant le FK pour les modèles de Markov couple (voir Algorithme 4.1) ainsi que le FK classique pour le modèle $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$. Bien sûr, cette dernière solution est ici optimale au sens de la minimisation de l'EQM car les données ont été générées suivant $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$. Nous notons $\hat{x}_{n,p,1}$ (resp. $\hat{x}_{n,p,2}$) l'estimateur basé sur le modèle $p^1(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$ (resp. basé sur le modèle $p^2(\mathbf{x}_{0:n}, \mathbf{y}_{0:n})$) pour la p -ème réalisation, à l'instant n . Pour chaque estimateur, nous calculons

$$\mathcal{J}^i = \frac{1}{T} \sum_{n=1}^T \left[\frac{1}{P} \sum_{p=1}^P (\hat{x}_{n,p,i} - x_{n,p})^2 \right] \quad (4.3.43)$$

où $x_{n,p}$ est le vrai état pour la p -ème réalisation, à l'instant n . Nous calculons également la DKL (4.3.42) entre $p_{n|n-1}^2(\xi_n|\xi_{n-1})$ et $p_{n|n-1}^1(\xi_n|\xi_{n-1})$. La DKL ainsi que l'écart relatif des EQM $(\mathcal{J}^2 - \mathcal{J}^1)/\mathcal{J}^1$ sont tracés en fonction de Q sur la Figure 4.4. L'EQM relative décroît lorsque $DKL(p_{n|n-1}^2, p_{n|n-1}^1)$ décroît, c'est à dire lorsque Q augmente. Il est intéressant de remarquer que les valeurs relatives de l'EQM sont inférieures à 0.10 lorsque $Q \geq 4$ et que pour de grandes valeurs de Q ($Q = 10$), elles sont proches de 0.03. Cela signifie qu'estimer les données cachées à partir du modèle (4.3.40) plutôt qu'à partir du modèle (4.3.36)-(4.3.37) n'a pas de conséquences importantes tant que Q n'est pas trop petit.

Simulations des modèles à sauts

Après avoir analysé les différences entre le modèle originel et le modèle approché, nous générons des données suivant des modèles à sauts et nous analysons les performances de notre solution approchée. Nous implémentons notre estimateur approché $\hat{x}_{n,p,1}$ du Paragraphe 4.3.2, un estimateur $\hat{x}_{k,p,2}$ basé sur le FP [44], un estimateur $\hat{x}_{n,p,3}$ basé sur l'algorithme IMM [17] et un estimateur $\hat{x}_{n,p,\text{kalm}}$ basé sur le FK qui utilise les vrais sauts et qui servira de référence. Pour le FP, nous utilisons $N = 100$ particules, tirées suivant la loi $p(r_n|r_{n-1})$. Nous calculons pour $P = 200$ réalisations l'EQM entre les différents estimateurs et l'estimateur du FK qui utilise les vrais sauts :

$$\text{EQM}^i(n) = \frac{1}{P} \sum_{p=1}^P (\hat{x}_{n,p,i} - \hat{x}_{n,p,\text{kalm}})^2. \quad (4.3.44)$$

Modèle à sauts réel : Nous reprenons le modèle (4.3.34)-(4.3.35) où $R_n \in \{1, 2, 3\}$, $a_n(r_n) = [1, -0.9, 0.9]$, $b = 1$, $Q(r_n) = [3, 10, 10]$ et $R = 1$. Les probabilités de transition sont données par $p^1(r_n|r_{n-1}) = 0.8$ si $r_n = r_{n-1}$ et $p^1(r_n|r_{n-1}) = 0.1$ si $r_n \neq r_{n-1}$. Une réalisation du scénario est tracée sur la Figure 4.5(a).

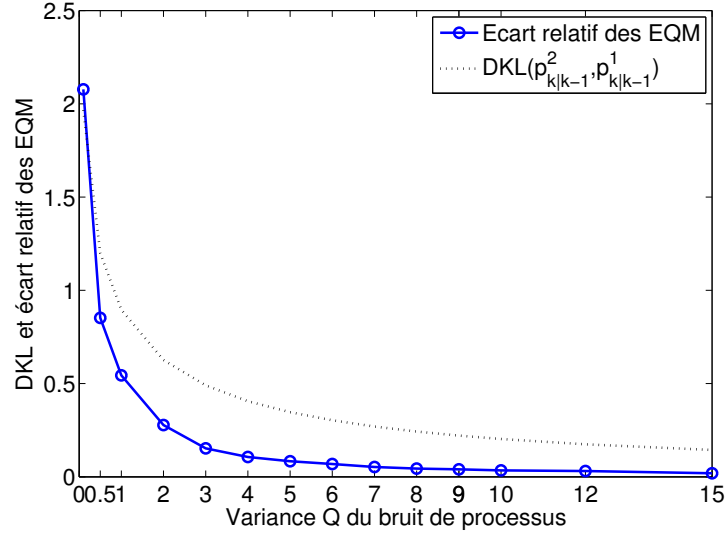


FIGURE 4.4 – EQM relative entre un FK basé sur (4.3.34)-(4.3.35) et un FK pour PMC basé sur (4.3.40) (cercles bleus) - DKL entre les transitions du modèle (4.3.34)-(4.3.35) et du modèle (4.3.40) (ligne noire en pointillé). Lorsque Q augmente, l'EQM relative et la DKL diminuent ; les estimateurs basés sur le modèle $p^2(\cdot)$ sont très proches des estimateurs optimaux basés sur $p^1(\cdot)$.

D'après (4.3.40), notre méthode d'approximation exacte est basée sur les modèles triplet dont la transition s'écrit

$$p^2(\boldsymbol{\xi}_n, r_n | \boldsymbol{\xi}_{n-1}, \mathbf{r}_{n-1}) = p^1(r_n | r_{n-1}) \mathcal{N}(\boldsymbol{\xi}_n; \mathbf{B}_n(\mathbf{r}_{n-1:n}) \boldsymbol{\xi}_{n-1}; \boldsymbol{\Sigma}(\mathbf{r}_{n-1:n})), \quad (4.3.45)$$

$$\mathbf{B}_n(\mathbf{r}_{n-1:n}) = \begin{bmatrix} a(r_n) - \frac{a(r_n)b^2Q(r_n)}{R+b^2Q(r_n)} & \frac{a(r_n)bQ(r_n)}{R(r_n)+b^2Q(r_n)} \\ 0 & a(r_n) \end{bmatrix}, \quad (4.3.46)$$

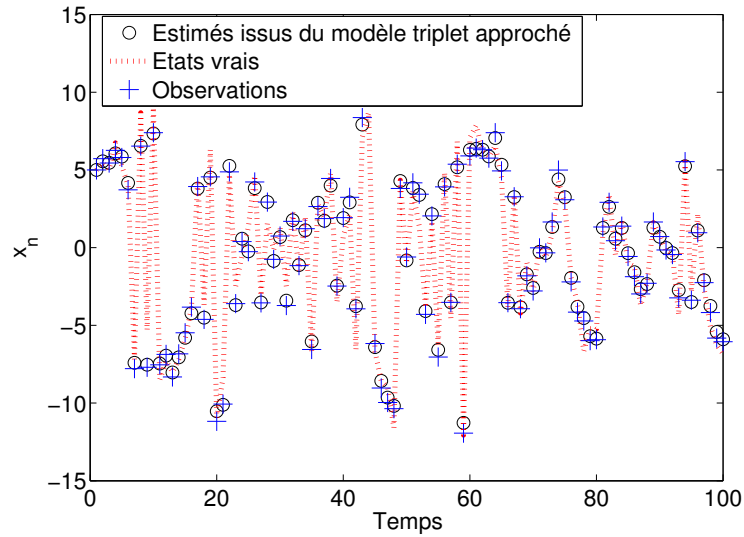
$$\boldsymbol{\Sigma}(\mathbf{r}_{n-1:n}) = \begin{bmatrix} Q(r_n) - \frac{a(r_n)^2b^2Q(r_n)^2R}{(R+b^2Q(r_n))^2} & bQ(r_n) - \frac{a(r_n)^2bQ(r_n)R}{R+b^2Q(r_n)} \\ bQ(r_n) - \frac{a(r_n)^2bQ(r_n)R}{R+b^2Q(r_n)} & R(1 - a(r_n)^2) + b^2Q(r_n) \end{bmatrix}. \quad (4.3.47)$$

Les EQM des différents estimateurs sont tracées sur la Figure 4.5(b) et sont normalisées par rapport à l'EQM de notre solution approchée. L'algorithme de filtrage exact dans $p^2(\cdot)$ donne de meilleurs résultats que l'algorithme IMM dans $p^1(\cdot)$ et améliore très légèrement le FP dans $p^1(\cdot)$. Même si notre solution n'est que légèrement préférable au FP, rappelons que les tirages probabilistes impliqués dans l'étape de propagation et de rééchantillonnage du FP requièrent un coût de calcul important. Nous avons donc calculé le rapport des temps moyens d'exécution entre notre solution et le FP. Même si les particules sont tirées suivant la transition $p(r_n | r_{n-1})$ (par conséquent, un seul FK par particule est nécessaire), cette solution est en moyenne 10 fois moins rapide que notre solution alternative approchée. Notre approche est donc beaucoup plus intéressante d'un point de vue calculatoire que le FP, pour un niveau de performances équivalent.

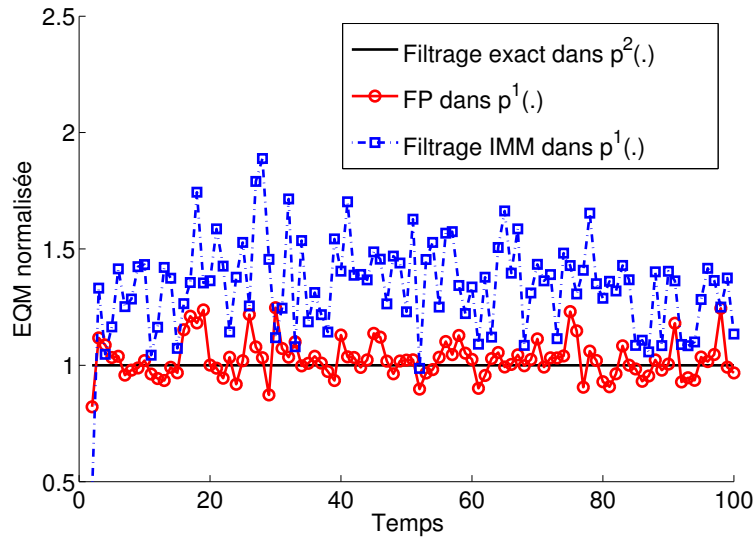
Suivi d'objet : Considérons maintenant un scénario de suivi d'objet manoeuvrant :

$$f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n(r_n) \mathbf{x}_{n-1}; \mathbf{Q}_n(r_n)), \quad (4.3.48)$$

$$g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) = \mathcal{N}(\mathbf{y}_n; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n), \quad (4.3.49)$$



(a)



(b)

FIGURE 4.5 – (a) -Exemple de réalisation du scénario correspondant au modèle (4.3.34)-(4.3.35) et restauration avec un modèle triplet de la Proposition 4.5 vérifiant (4.3.26) et (4.3.33). Les vrais états (points rouges) ; états estimés basé sur notre approximation (cercles noirs) et observations (croix bleues). (b) -EQM normalisée de notre estimateur (ligne noire), de l'estimateur basé sur le FP (cercles rouges) et de l'estimateur IMM (carrés bleus). Notre approche est préférable à l'IMM et légèrement préférable au FP.

où

$$\mathbf{F}_n(r_n) = \begin{bmatrix} 1 & \frac{\sin(\omega_r T_s)}{\omega_r} & 0 & -\frac{1-\cos(\omega_r T_s)}{\omega_r} \\ 0 & \cos(\omega_r T_s) & 0 & -\sin(\omega_r T_s) \\ 0 & \frac{1-\cos(\omega_r T_s)}{\omega_r} & 1 & \frac{\sin(\omega_r T_s)}{\omega_r} \\ 0 & \sin(\omega_r T_s) & 0 & \cos(\omega_r T_s) \end{bmatrix}, \mathbf{Q}_n(r_n) = \sigma_v^2(r_n) \begin{bmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} & 0 & 0 \\ \frac{T_s^2}{2} & T_s & 0 & 0 \\ 0 & 0 & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ 0 & 0 & \frac{T_s^2}{2} & T_s \end{bmatrix},$$

$\mathbf{H}_n = \mathbf{I}_4$ et $\mathbf{R}_n = 4\mathbf{I}_4$. Nous prenons $T_s = 2$, $R_n \in \{1, 2, 3\}$ représente le comportement dynamique de l'objet (ligne droite, virage à gauche, virage à droite). Nous choisissons $w_r = [0, 6\pi/180, -6\pi/180]$, $\sigma_v(r_n) = [7, 10, 10]$, $p^1(r_n|r_{n-1}) = 0.8$ si $r_n = r_{n-1}$ et $p^1(r_n|r_{n-1}) = 0.1$ si $r_n \neq r_{n-1}$. Une réalisation de ce scénario est tracée sur la Figure 4.6(a).

Les paramètres de notre modèle triplet utilisé pour appliquer le filtrage exact reposent sur les modèles de la Proposition 4.5, où $\mathbf{H}_n^2(\mathbf{r}_{n-1:n})$ vérifie (4.3.26) (donc $\mathbf{H}_n^2(\mathbf{r}_{n-1:n}) = \mathbf{F}_n(r_n)$) et $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$ vérifie (4.3.33). Les EQM normalisées sont tracées sur la Figure 4.6(b). Notre solution approchée améliore l'algorithme IMM et présente des résultats similaires au FP. Cependant, le temps d'exécution de notre algorithme est une nouvelle fois dix fois moins élevé que celui du FP.

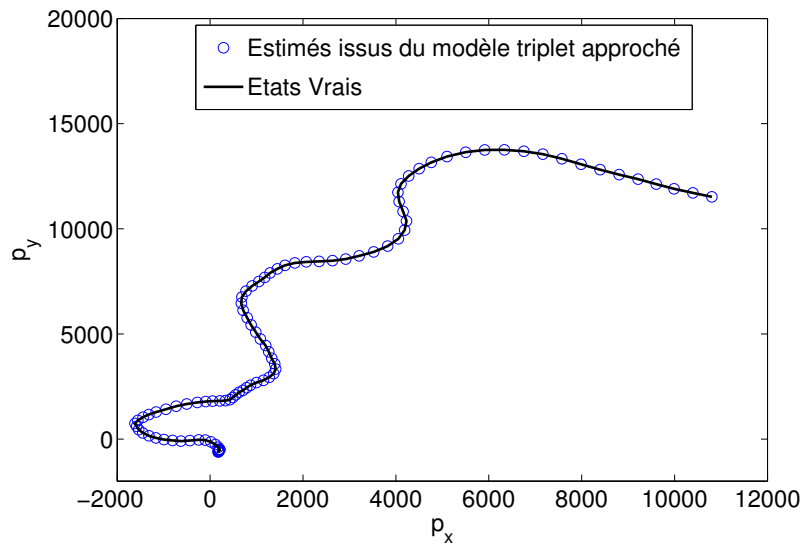
Enfin, dans toutes ces simulations, nous avons considéré le cas le moins favorable qui soit car les données ont en effet été générées suivant un JMSS linéaire et gaussien. Cependant, les données peuvent suivre un modèle plus général d'un point de vue statistique, tout en gardant les mêmes propriétés physiques, comme les modèles de la Proposition 4.5. L'utilisation du FP classique et de l'algorithme IMM supposent que les données vérifient les propriétés des modèles à sauts classiques. Par ailleurs, notre solution est utilisable pour une large classe de modèles puisque $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$ peut être choisi arbitrairement et s'adapter en fonction des données traitées. Générons alors des données suivant un modèle de la Proposition 4.5 avec $\mathbf{F}_n^{2,\text{vraie}}(\mathbf{r}_{n-1:n}) = 0.7\mathbf{F}_n(r_n)$ et $\mathbf{H}_n^{2,\text{vraie}}(\mathbf{r}_{n-1:n}) = 0.9\mathbf{H}_n(r_n)$. Nous implémentons les estimateurs en utilisant les mêmes algorithmes de FP et IMM que ci-dessus puis nous implémentons notre solution avec $\mathbf{F}_n^2(\mathbf{r}_{n-1:n}) = 0.8\mathbf{F}_n(r_n)$ et $\mathbf{H}_n^2(\mathbf{r}_{n-1:n})$ vérifiant (4.3.26), afin de pouvoir appliquer notre technique de filtrage exact. Remarquons que le choix $\mathbf{F}_n^2(\mathbf{r}_{n-1:n}) = \mathbf{F}_n^{2,\text{vraie}}(\mathbf{r}_{n-1:n})$ n'est pas nécessairement optimal car la condition (4.3.26) implique $\mathbf{H}_n^2(\mathbf{r}_{n-1:n}) \neq \mathbf{H}_n^{2,\text{vraie}}(\mathbf{r}_{n-1:n})$. La solution référence n'est plus le FK puisque les données ne suivent pas le modèle à sauts classique ; nous utilisons donc le FK de l'Algorithme 4.1, qui utilise la vraie valeur des sauts $\mathbf{r}_{0:n}$. Sur la Figure 4.7(a), nous traçons une réalisation de ce scénario. Comme nous le voyons, l'objet caché conserve les propriétés physiques du scénario précédent (trajectoire rectiligne, virage), bien que sa trajectoire ne soit plus générée suivant le modèle à sauts classique. Cependant, sur la Figure 4.7(b) nous traçons l'EQM normalisée et nous voyons que les solutions classiques ne sont plus du tout adaptées lorsque nous considérons des scénarios plus complexes.

En résumé, notre technique de filtrage exact peut donner des résultats aussi performants que les méthodes d'approximations particulières, mais à moindre coût, dans le cas le plus défavorable (c'est à dire le cas où les données suivent effectivement le modèle à sauts linéaire et gaussien (4.0.3)-(4.0.4)), mais améliore sensiblement les performances lorsque des modèles plus généraux sont considérés.

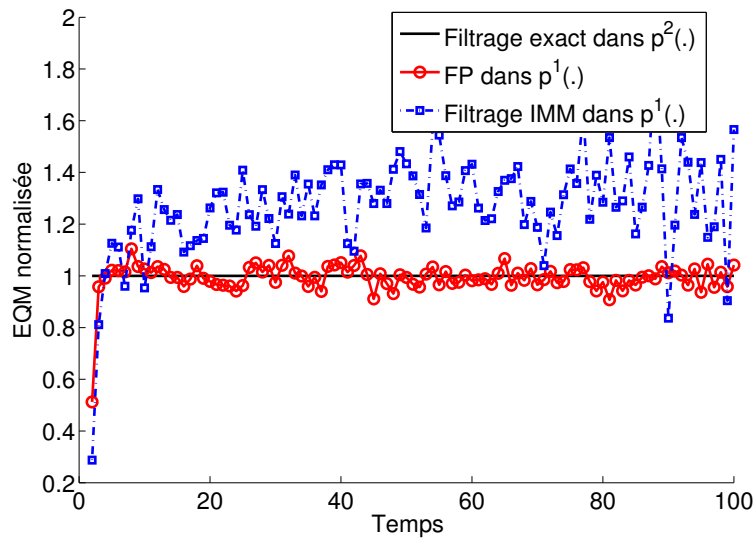
4.4 Conclusion

Dans ce chapitre, nous avons rappelé que les modèles de Markov cachés et les modèles de Markov à sauts markoviens pouvaient être respectivement étendus aux modèles de Markov couple et triplet.

Nous avons d'abord présenté l'intérêt statistique de ces modèles, puis nous avons discuté de la possibilité de calculer ou d'approcher l'estimateur bayésien optimal, au sens de l'EQM. Point important, nous avons

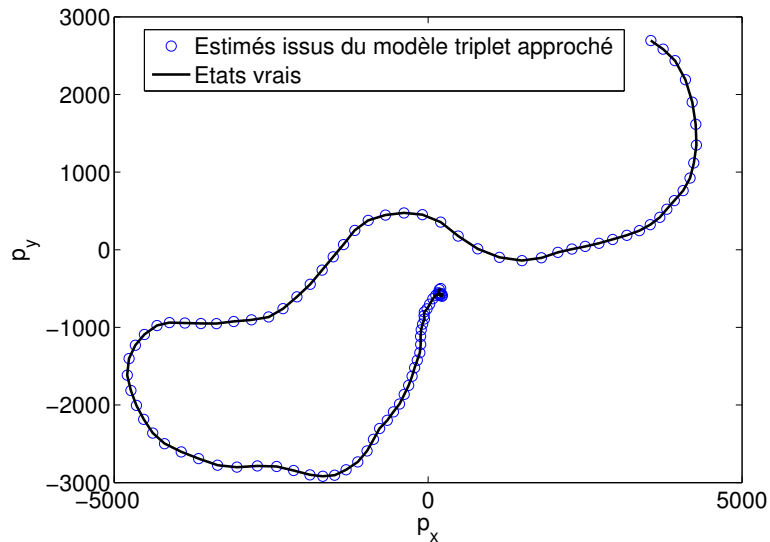


(a)

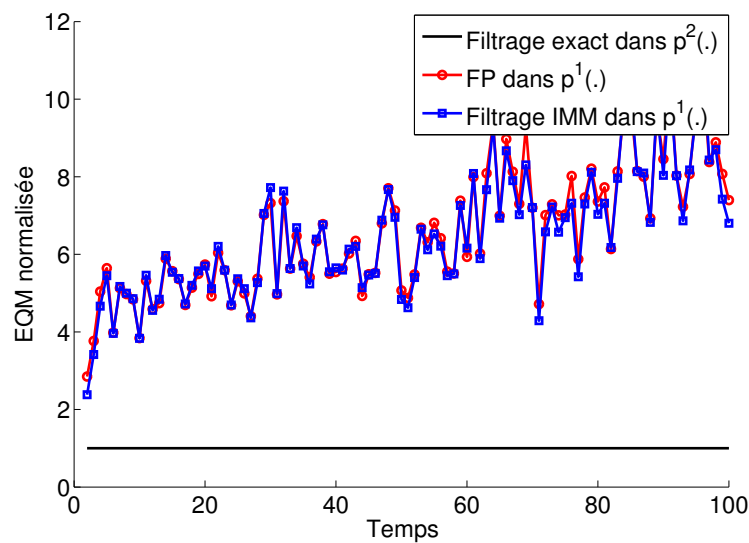


(b)

FIGURE 4.6 – (a) - Exemple d'un scénario d'objet manoeuvrant - Les données sont générées suivant le modèle (4.3.48)-(4.3.49). (b) - EQM normalisée de l'estimateur associé à notre algorithme (ligne noire), au FP (cercles rouges) et à l'IMM (carrés bleus). Notre solution est préférable à l'IMM et donne des résultats similaires au FP.



(a)



(b)

FIGURE 4.7 – (a) - Exemple d'un scénario d'objet manoeuvrant où les données sont générées suivant un modèle triplet de la Proposition 4.5 avec $\mathbf{F}_n^{2,vraie}(\mathbf{r}_{n-1:n}) = 0.7\mathbf{F}_n(r_n)$ et $\mathbf{H}_n^{2,vraie}(\mathbf{r}_{n-1:n}) = 0.9\mathbf{H}_n(r_n)$. Les propriétés physiques du scénario de la Figure 4.6(a) (virage à gauche, virage à droite, trajectoire rectiligne) sont conservées. (b) - EQM normalisée de l'estimateur associé à notre algorithme (ligne noire), au FP (cercles rouges) et à l'IMM (carrés bleus). Les solutions classiques ne sont plus adaptées pour de tels modèles, tandis que notre approximation reste utilisable. Cela est dû au fait qu'il est possible d'ajuster ici le paramètre $\mathbf{F}_n^2(\mathbf{r}_{n-1:n})$.

montré que le calcul exact de cet estimateur était possible dans une certaine classe de modèle triplet, qui dépendent, comme les modèles JMSS, d'un processus latent discret.

Nous nous sommes ensuite intéressés au problème de filtrage multi-objets pour les modèles couple. La formulation courante, en terme de RFS, est particulièrement difficile à utiliser dans ce contexte à cause des nouvelles dépendances entre les états cachés et les observations qui apparaissent dès lors que ces modèles sont considérés. Nous avons donc reformulé ce problème de filtrage de façon à prendre en compte les propriétés statistiques de ces modèles ; en particulier, la généralité des modèles PMC permet de considérer des modèles dans lesquels l'état caché de chaque objet ne suit pas une dynamique markovienne. Nous avons par ailleurs établi le filtre PHD pour les modèles couple et une implémentation par mélange de gaussiennes de ce filtre a été proposée. Sur des simulations, nous avons comparé notre filtre PHD pour les modèles PMC avec un modèle HMC partageant des propriétés physiques identiques, pour lequel nous avons implémenté le filtre PHD classique. Bien que les deux modèles partagent les mêmes propriétés physiques, notre implémentation, qui prend en compte les propriétés plus générales des modèles PMC, permet d'améliorer sensiblement les estimateurs du nombre d'objets et de chaque état caché.

Enfin, nous nous sommes focalisé sur le problème NP-complet du calcul de l'estimateur optimal, au sens de la minimisation de l'EQM, pour des modèles à sauts. Nous avons proposé une classe d'algorithmes de filtrage rapide consistant à approcher le modèle à sauts de départ par un modèle triplet dans lequel les propriétés physiques du modèle originel sont conservées, et dans lequel il est possible de calculer l'estimateur optimal, en un coût linéaire en nombre d'observations. Dans ce contexte, nous avons discuté de la pertinence de cette approximation, en fonction des paramètres du modèle de départ. Nous avons alors montré que cette approximation pouvait donner des résultats très proches d'un algorithme de FP dans le modèle originel, mais avec un coût de calcul beaucoup moins élevé. Enfin nous avons montré que notre technique pouvait améliorer nettement les techniques d'approximation existantes lorsqu'elle était employée dans une classe de modèles à sauts où les contraintes physiques sont vérifiées, mais dont les propriétés statistiques sont plus générales.

Conclusion et perspectives

Tout au long de cette thèse, nous avons abordé le problème de l'estimation bayésienne dans des modèles stochastiques qui fournissent des liens (probabilistes) entre les états cachés inconnus et les observations.

Dans un premier temps, nous nous sommes focalisés sur les méthodes SMC, qui sont d'intérêt dès lors que le modèle stochastique ne permet pas de calculer récursivement l'estimateur bayésien optimal. Dans ce cas, il est possible de propager une approximation discrète de la loi de filtrage, de laquelle est déduite une approximation de l'estimateur optimal. Les mécanismes de propagation, de pondération et de rééchantillonnage intervenant dans cette classe d'approximations ont été utilisés pour l'implémentation de boucles de propagation alternatives. Certains de ces algorithmes, qui ne diffèrent que par l'ordre des opérations qui les composent ont, de plus, fait l'objet d'une analyse statistique non asymptotique locale, ce qui nous a permis de comparer leurs performances en fonction du modèle et du nombre effectif de particules et de proposer un algorithme hybride qui à tout instant choisit automatiquement entre deux algorithmes SMC concurrents.

L'importante part que constituent les techniques de réduction de variance pour les méthodes SMC a été abordée sous différents points de vue. Tout d'abord, l'utilisation classique de ces techniques, c'est à dire dans le cas où la dimension du vecteur d'état est importante, a été étendue au contexte du filtrage multi-objets. Les hypothèses nécessaires à l'implémentation marginalisée du filtre PHD ont été discutées, puis l'apport de cette implémentation a été validée sur plusieurs scénarios pratiques. D'autre part, l'application de ces méthodes à une partition temporelle du vecteur d'état a été proposée, pour les problèmes de filtrage mono- et multi-objet(s). Ces filtres marginalisés temporellement peuvent être calculés exactement dans plusieurs classes de modèles incluant les modèles semi-linéaires et gaussiens et les modèles à sauts linéaires (ou semi-linéaires) et gaussiens. Dans les autres cas, différentes approximations ont été proposées, selon le degré de non linéarité du modèle sous-jacent.

Enfin, l'utilisation de modèles alternatifs markoviens plus généraux a été étudiée. En particulier, la possibilité de prendre en compte des propriétés statistiques plus générales (propriétés d'indépendances non nécessairement vérifiées), nous a incité à nous pencher sur le problème d'estimation bayésienne dans ce contexte. Le problème du filtrage multi-objets a donc été abordé dans le cadre des modèles de Markov couple, et une implémentation du filtre PHD par mélange de gaussiennes a été proposée. Par ailleurs, l'exploitation des propriétés plus générales des modèles triplet nous a conduit à proposer des algorithmes de filtrage exact dans des modèles dépendant de variables latentes, dans lesquels les propriétés physiques sont fixées. L'intérêt de ces algorithmes est double. D'une part, ils peuvent approcher l'estimateur bayésien optimal (au sens de l'EQM) dans des modèles JMSS linéaires et gaussiens et fournissent une alternative rapide aux méthodes SMC grâce à leur complexité linéaire en nombre d'observations ; d'autre part, ces algorithmes ne sont pas restreints à l'unique classe des modèles JMSS linéaires et gaussiens, mais peuvent être utilisés dans des modèles physiquement contraints mais dans lesquels les propriétés d'indépendance sont plus générales.

Les techniques de filtrage proposées dans le dernier chapitre pourraient par ailleurs ouvrir la voie à de nouvelles méthodes d'estimation bayésienne.

Tout d'abord, pour l'approximation d'estimateurs dans les modèles à sauts linéaires et gaussiens, on pourrait construire une méthode alternative à celle proposée dans la Section 4.3 ; plutôt que de commencer par garder les propriétés physiques de ces modèles, on pourrait d'abord imposer la condition permettant le calcul exact et rapide de l'estimateur bayésien optimal puis rechercher, parmi cette classe de modèles, celui qui est le plus proche d'un modèle à sauts markoviens linéaire et gaussien donné, en rajoutant éventuellement certaines propriétés d'intérêt.

De plus, on pourra réfléchir à l'utilisation de ces modèles pour le problème du filtrage exact dans les modèles HMC non linéaires et/ou non gaussiens. En effet, ces modèles pourraient être approchés par des modèles triplet dans lesquels le filtrage exact est faisable, et dans lesquels les paramètres seraient estimés par une méthode d'apprentissage, de type Expectation-Maximization [35] [89]. Ainsi, le modèle originel serait approché par un modèle dans lequel le calcul exact de l'estimateur bayésien optimal est faisable et rapide.

Remarquons enfin que la propagation exacte du PHD, même dans le cas linéaire et gaussien, se heurte au problème de la croissance exponentielle d'un mélange. Des solutions similaires à celle de l'approximation du calcul de l'estimateur dans des modèles de Markov cachés à sauts pourraient être envisagées, c'est à dire qu'on pourrait s'intéresser à la construction de modèles probabilistes dans lesquels un estimateur du nombre d'objets et des états cachés pourrait être calculé récursivement sans approximation.

Conditionnement dans les vecteurs aléatoires gaussiens

Sommaire

A.1 Résultat 1	157
A.2 Résultat 2	157

Dans cette annexe sont rappelés deux résultats, utilisés dans les différents chapitre de ce travail, sur les densités de probabilités gaussiennes (voir par exemple [98]).

A.1 Résultat 1

Lemme A.1. Soient $\zeta \in \mathbb{R}^p$, $\eta \in \mathbb{R}^q$, \mathbf{Q} (resp. \mathbf{P}) une matrice définie positive de taille $p \times p$ (resp. $q \times q$) (les autres vecteurs et matrices sont de dimensions appropriées), alors

$$\int \mathcal{N}(\zeta; \mathbf{F}\eta + \mathbf{d}; \mathbf{Q})\mathcal{N}(\eta; \mathbf{m}; \mathbf{P})d\eta = \mathcal{N}(\zeta; \mathbf{F}\mathbf{m} + \mathbf{d}; \mathbf{Q} + \mathbf{F}\mathbf{P}\mathbf{F}^T), \quad (\text{A.1.1})$$

A.2 Résultat 2

Lemme A.2. Soient $\zeta \in \mathbb{R}^p$, $\eta \in \mathbb{R}^q$, \mathbf{P}^ζ (resp. \mathbf{P}^η) une matrice définie positive de taille $p \times p$ (resp. $q \times q$) et $\mathbf{P}^{\zeta, \eta}$ une matrice de taille $p \times q$. Supposons que $p(\zeta, \eta)$ est une gaussienne multi-dimensionnelle,

$$p(\zeta, \eta) = \mathcal{N}\left(\zeta, \eta; \begin{bmatrix} \mathbf{m}^\zeta \\ \mathbf{m}^\eta \end{bmatrix}; \begin{bmatrix} \mathbf{P}^\zeta & \mathbf{P}^{\zeta, \eta} \\ \mathbf{P}^{\zeta, \eta T} & \mathbf{P}^\eta \end{bmatrix}\right). \quad (\text{A.2.1})$$

Alors

$$p(\zeta, \eta) = \mathcal{N}(\eta; \mathbf{m}^\eta; \mathbf{P}^\eta)\mathcal{N}(\zeta; \tilde{\mathbf{m}}^\zeta(\eta); \tilde{\mathbf{P}}^\zeta), \quad (\text{A.2.2})$$

$$\tilde{\mathbf{m}}^\zeta(\eta) = \mathbf{m}^\zeta + \mathbf{P}^{\zeta, \eta}(\mathbf{P}^\eta)^{-1}(\eta - \mathbf{m}^\eta), \quad (\text{A.2.3})$$

$$\tilde{\mathbf{P}}^\zeta = \mathbf{P}^\zeta - \mathbf{P}^{\zeta, \eta}(\mathbf{P}^\eta)^{-1}\mathbf{P}^{\zeta, \eta T}. \quad (\text{A.2.4})$$

Preuves des résultats du Chapitre 2

Sommaire

B.1	Preuve de la Proposition 2.1	159
B.2	Preuve de la Proposition 2.2	160
B.3	Preuve de la Proposition 2.3	160

B.1 Preuve de la Proposition 2.1

Les étapes discrète et continue de la méthode peuvent être réécrites ainsi :

$$\tilde{\mathbf{X}}^{2,1:N} \sim p(\tilde{\mathbf{x}}^{2,1:N}) = \prod_{j=1}^N p_j(\tilde{x}^{2,j}); \quad (\text{B.1.1})$$

$$\mathbf{X}^{2,1:N'} \sim P(\mathbf{X}^{2,1:N'} = \mathbf{x}^{2,1:N'} | \tilde{\mathbf{X}}^{2,1:N} = \tilde{\mathbf{x}}^{2,1:N}) = \prod_{j=1}^{N'} \left(\sum_{k=1}^N \alpha_k \delta_{\tilde{x}^{2,k}}(\{x^{2,j}\}) \right). \quad (\text{B.1.2})$$

Soit $X^{2,j}$ (pour $1 \leq j \leq N$) un point produit par la méthode de composition inverse. Soit $\mathbb{1}_A(x) = 1$ si $x \in A$ et 0 sinon. Pour un Borélien $A \in \mathcal{B}(\mathbb{R})$ soit $\mu(A) = P(X^{2,j} \in A)$. Alors

$$\begin{aligned} \mu(A) &= \int_{\mathbb{R}^N} \left[\sum_{i=1}^N \alpha_i \delta_{\tilde{x}^{2,i}}(A) \right] \prod_{j=1}^N p_j(\tilde{x}^{2,j}) d\tilde{\mathbf{x}}^{2,1:N} = \sum_{i=1}^N \alpha_i \int_{\mathbb{R}^N} \delta_{\tilde{x}^{2,i}}(A) \prod_{j=1}^N p_j(\tilde{x}^{2,j}) d\tilde{\mathbf{x}}^{2,1:N} \\ &= \sum_{i=1}^N \alpha_i \int_{\mathbb{R}} \mathbb{1}_A(\tilde{x}^{2,i}) p_i(\tilde{x}^{2,i}) d\tilde{x}^{2,i} = \int_A \underbrace{\sum_{i=1}^N \alpha_i p_i(x)}_{\pi(x)} dx, \end{aligned}$$

donc μ a comme densité de probabilité $\pi(\cdot)$ par rapport à la mesure de Lebesgue. □

B.2 Preuve de la Proposition 2.2

Soient $B(x) =]-\infty, x]$, $C(dx) = [x^1, x^1 + dx[$ et $\mu_N(\cdot|x, dx) = \mathbb{P}(X^2 \in B(x), X^1 \in C(dx))$. Nous avons

$$\begin{aligned} \mu_N(x, dx) &= \int_{\mathbb{R}^N} \prod_{i=1}^N p_i(\tilde{x}^{2,i}) \left(\sum_{k=1}^N \alpha_k \delta_{\tilde{x}^{2,k}}(B(x)) \right) \left(\sum_{j=1}^N \alpha_j \delta_{\tilde{x}^{2,j}}(C(dx)) \right) d\tilde{x}^{2,1} \dots d\tilde{x}^{2,N} \\ &= \int_{\mathbb{R}^N} \prod_{i=1}^N p_i(\tilde{x}^{2,i}) \left(\sum_{p=1}^N \alpha_p^2 \delta_{\tilde{x}^{2,p}, \tilde{x}^{2,p}}(B(x) \times C(dx)) \right) d\tilde{x}^{2,1} \dots d\tilde{x}^{2,N} + \\ &\quad \int_{\mathbb{R}^N} \prod_{q=1}^N p_q(\tilde{x}^{2,q}) \left(\sum_{k=1}^N \sum_{\substack{j=1 \\ j \neq k}}^N \alpha_k \alpha_j \delta_{\tilde{x}^k, \tilde{x}^{2,j}}(B(x) \times C(dx)) \right) d\tilde{x}^{2,1} \dots d\tilde{x}^{2,N} \\ &= \left[\sum_{p=1}^N \alpha_p^2 \int_{D(x, dx)} p_p(\tilde{x}^{2,p}) d\tilde{x}^{2,p} \right] \mathbb{1}_{x > x^1} + \sum_{k=1}^N \sum_{\substack{j=1 \\ j \neq k}}^N \alpha_k \alpha_j \int_{B(x)} p_k(\tilde{x}^{2,k}) d\tilde{x}^{2,k} \int_{C(dx)} p_j(\tilde{x}^{2,j}) d\tilde{x}^{2,j} \end{aligned}$$

avec $D(x, dx) = [x^1, \min(x^1 + dx, x)[$. Un développement en série de Taylor au premier ordre de $\mu_N(x, dx)$ donne

$$\mu_N(x, dx) = \left[\left(\sum_{p=1}^N \alpha_p^2 p_p(x^1) \right) \mathbb{1}_{x > x^1} + \sum_{k=1}^N \sum_{\substack{j=1 \\ j \neq k}}^N \alpha_k \alpha_j p_j(x^{2,1}) \int_{B(x)} p_k(\tilde{x}^{2,k}) d\tilde{x}^{2,k} \right] dx + o(dx)$$

avec $\frac{o(dx)}{dx} \rightarrow 0$ lorsque $dx \rightarrow 0$. Nous avons également

$$\mu_D(dx) \stackrel{\text{def}}{=} \mathbb{P}(X^1 \in C(dx)) = \pi(x^1) dx + o'(dx)$$

avec $\frac{o'(dx)}{dx} \rightarrow 0$ lorsque $dx \rightarrow 0$. Enfin,

$$\mathbb{P}(X^2 < x | X^1 = x^1) = \lim_{dx \rightarrow 0} \frac{\mu_N(x, dx)}{\mu_D(dx)} = \alpha \mathbb{1}_{x > x^1} + (1 - \alpha) \int_{B(x)} \tilde{\pi}(x) dx$$

où α et $\tilde{\pi}(\cdot)$ sont donnés respectivement par (2.4.3) et (2.4.4). □

B.3 Preuve de la Proposition 2.3

L'égalité des espérances est immédiate d'après la Proposition 2.2. (2.4.9) et (2.4.10) sont triviales. Calculons à présent

$$\text{var}(\hat{\Theta}^{2, N'}) = \frac{1}{N'^2} \sum_{i=1}^{N'} \text{var}(f(X^{2,i})) + \frac{1}{N'^2} \sum_{\substack{k, l=1 \\ k \neq l}}^{N'} \text{Cov}(f(X^{2,k}), f(X^{2,l})). \quad (\text{B.3.1})$$

D'après la Proposition 2.2, le premier terme dans (B.3.1) est égal à $\frac{1}{N'} \text{var}_\pi(f(X))$. Calculons alors le second terme. Nous avons d'abord

$$\mathbb{E}(f(X^{2,k})) | \tilde{\mathbf{x}}^{2,1:N} = \tilde{\Theta}^2(\tilde{\mathbf{x}}^{2,1:N}), \quad (\text{B.3.2})$$

B.3 Preuve de la Proposition 2.3

puis, d'après (B.3.2), $E(f(X^{2,k})f(X^{2,l})|\tilde{\mathbf{x}}^{2,1:N}) = (\tilde{\Theta}^2(\tilde{\mathbf{x}}^{2,1:N}))^2$, donc

$$E(f(X^{2,k})f(X^{2,l})) = E(E(f(X^{2,k})f(X^{2,l})|\tilde{\mathbf{X}}^{2,1:N})) = E((\tilde{\Theta}^2)^2),$$

pour tout $k, l, 1 \leq k, l, \leq N'$ avec $k \neq l$. En réutilisant (B.3.2), nous avons $\text{Cov}(f(X^{2,k}), f(X^{2,l})) = \text{var}(\tilde{\Theta}^2)$. En injectant dans (B.3.1) et en utilisant (2.4.10) nous obtenons (2.4.11). \square

Preuves des résultats du Chapitre 4

Sommaire

C.1	Preuve de la Proposition 4.1	163
C.2	Preuve des Propositions 4.2 à 4.3	164
C.3	Preuve de la Proposition 4.4 (filtrage exact dans les modèles de Markov triplet) . . .	165
C.4	Preuve de la Proposition 4.8	165

C.1 Preuve de la Proposition 4.1

Nous utilisons un raisonnement par récurrence. Supposons que pour tout i , $1 \leq i \leq n$, $p^2(\mathbf{x}_i|\mathbf{x}_{i-1}) = \mathcal{N}(\mathbf{x}_i; \mathbf{F}_i\mathbf{x}_{i-1}; \mathbf{Q}_i)$ et $p^2(\mathbf{y}_i|\mathbf{x}_i) = \mathcal{N}(\mathbf{y}_i; \mathbf{H}_i\mathbf{x}_i; \mathbf{R}_i)$. Alors la loi du couple $(\mathbf{X}_n, \mathbf{Y}_n)$ peut s'écrire

$$p^2(\boldsymbol{\xi}_n) = \mathcal{N}\left(\boldsymbol{\xi}_n; \begin{bmatrix} \mathbf{m}_n^x \\ \mathbf{H}_n \mathbf{m}_n^x \end{bmatrix}; \begin{bmatrix} \mathbf{P}_n^x & (\mathbf{H}_n \mathbf{P}_n^x)^T \\ \mathbf{H}_n \mathbf{P}_n^x & \mathbf{R}_n + \mathbf{H}_n \mathbf{P}_n^x \mathbf{H}_n^T \end{bmatrix}\right). \quad (\text{C.1.1})$$

Ensuite, d'après (4.1.14),

$$p^2(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n) = \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{F}_{n+1}^1 \mathbf{x}_n + \mathbf{F}_{n+1}^2 \mathbf{y}_n; \boldsymbol{\Sigma}_{n+1}^{11});$$

par conséquent, d'après le Lemme A.1 en Annexe A.1, la loi

$$p^2(\mathbf{x}_{n+1}|\mathbf{x}_n) = \int p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathbf{y}_n) p(\mathbf{y}_n|\mathbf{x}_n) d\mathbf{y}_n \quad (\text{C.1.2})$$

s'écrit en fonction de \mathbf{F}_{n+1}^1 , \mathbf{F}_{n+1}^2 , $\boldsymbol{\Sigma}_{n+1}^{11}$, \mathbf{H}_n et \mathbf{R}_n , et la condition (4.1.40) implique que

$$\mathbf{F}_{n+1}^1 + \mathbf{F}_{n+1}^2 \mathbf{H}_n = \mathbf{F}_{n+1}, \quad (\text{C.1.3})$$

$$\boldsymbol{\Sigma}_{n+1}^{11} + \mathbf{F}_{n+1}^2 \mathbf{R}_n (\mathbf{F}_{n+1}^2)^T = \mathbf{Q}_{n+1}. \quad (\text{C.1.4})$$

Ensuite, en utilisant à nouveau le Lemme A.1, nous calculons les paramètres de $p^2(\mathbf{x}_{n+1}, \mathbf{y}_{n+1})$,

$$p^2(\boldsymbol{\xi}_{n+1}) = \int p(\boldsymbol{\xi}_{n+1}|\boldsymbol{\xi}_n) p(\boldsymbol{\xi}_n) d\boldsymbol{\xi}_n, \quad (\text{C.1.5})$$

et nous déduisons du Lemme A.2 l'expression de $p^2(\mathbf{y}_{n+1}|\mathbf{x}_{n+1})$ en fonction des paramètres du modèle couple. La condition sur la moyenne dans (4.1.41) implique

$$\left[\mathbf{H}_{n+1}^1 - \mathbf{H}_{n+1} \mathbf{F}_{n+1} + \mathbf{H}_{n+1}^2 \mathbf{H}_n \right] \mathbf{m}_n^x = \mathbf{0}, \quad (\text{C.1.6})$$

qui doit être vérifiée pour tout \mathbf{m}_0 . D'après l'hypothèse de récurrence, $\mathbf{m}_n^x = \prod_{i=1}^n \mathbf{F}_i \mathbf{m}_0$. Une condition suffisante (qui devient nécessaire si les matrices $\{\mathbf{F}_i\}_{i \geq 0}$ sont inversibles, ce que nous supposons ici) est

$$\mathbf{H}_{n+1}^1 - \mathbf{H}_{n+1} \mathbf{F}_{n+1} + \mathbf{H}_{n+1}^2 \mathbf{H}_n = \mathbf{0}. \quad (\text{C.1.7})$$

En combinant la condition sur la variance dans (4.1.41) et (C.1.7), nous obtenons l'expression de Σ_{n+1}^{12} et Σ_{n+1}^{22} :

$$\Sigma_{n+1}^{21} = \mathbf{H}_{n+1} \mathbf{Q}_{n+1} - \mathbf{H}_{n+1}^2 \mathbf{R}_n (\mathbf{F}_{n+1}^2)^T, \quad (\text{C.1.8})$$

$$\Sigma_{n+1}^{22} = \mathbf{R}_{n+1} - \mathbf{H}_{n+1}^2 \mathbf{R}_n (\mathbf{H}_{n+1}^2)^T + \mathbf{H}_{n+1} \mathbf{Q}_{n+1} \mathbf{H}_{n+1}^T, \quad (\text{C.1.9})$$

ce qui conclut la démonstration. \square

Remarque C.1. Remarquons que si les matrices $\{\mathbf{F}_i\}_{i \geq 0}$ ne sont pas inversibles, le résultat reste valide mais il est alors possible de trouver d'autres modèles PMC qui vérifient les contraintes (4.1.36)-(4.1.38) et qui n'appartiennent pas à la classe de modèles de la Proposition 4.1.

C.2 Preuve des Propositions 4.2 à 4.3

Nous commençons avec (4.1.49). Soit $p_{n|n-1}^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1})$ la transition d'un modèle PMC de la Proposition 4.1. Nous avons

$$p^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}) = \int \underbrace{p^2(\mathbf{y}_{n-1} | \mathbf{x}_{n-1})}_{g_{n-1}(\mathbf{y}_{n-1} | \mathbf{x}_{n-1})} p_{n|n-1}^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{y}_{n-1}, \mathbf{x}_{n-1}) d\mathbf{y}_{n-1}.$$

De plus, $g_{n-1}(\mathbf{y}_{n-1} | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{y}_{n-1}; \mathbf{H}_{n-1} \mathbf{x}_{n-1}; \mathbf{R}_{n-1})$ et $p_{n|n-1}^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{y}_{n-1}, \mathbf{x}_{n-1})$ est une gaussienne dont les paramètres sont donnés par (4.1.44)-(4.1.48). En utilisant le Lemme A.1, nous obtenons (4.1.49). Nous démontrons à présent (4.1.50), par récurrence. Supposons que

$$p^2(\mathbf{y}_{n-1} | \mathbf{x}_{0:n-1}) = p^2(\mathbf{y}_{n-1} | \mathbf{x}_{n-1}) = g_{n-1}(\mathbf{y}_{n-1} | \mathbf{x}_{n-1}) \quad (\text{C.2.1})$$

((C.2.1) est vrai à l'instant $n = 1$). Puisque $(\mathbf{X}_n, \mathbf{Y}_n)$ est une chaîne de Markov couple, nous avons

$$p^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{0:n-1}) \stackrel{\text{PMC}}{=} \int p_{n|n-1}^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) p^2(\mathbf{y}_{n-1} | \mathbf{x}_{0:n-1}) d\mathbf{y}_{n-1} \quad (\text{C.2.2})$$

$$\stackrel{(\text{C.2.1})}{=} p^2(\mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}) \quad (\text{C.2.2})$$

$$\stackrel{(4.1.49)}{=} f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) g_n(\mathbf{y}_n | \mathbf{x}_n). \quad (\text{C.2.3})$$

De (C.2.2), nous obtenons

$$p^2(\mathbf{x}_n | \mathbf{x}_{0:n-1}) = f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}), \quad (\text{C.2.4})$$

et par conséquent $p^2(\mathbf{y}_n | \mathbf{x}_{0:n}) = g_n(\mathbf{y}_n | \mathbf{x}_n)$, qui n'est autre que (C.2.1) à l'instant n , et qui prouve donc (4.1.50). Puisque (C.2.1) est vérifiée, alors (C.2.4) l'est aussi, d'où (4.1.51). Il reste donc à prouver (4.1.4). Puisque $\{(\mathbf{X}_n, \mathbf{Y}_n)\}_{n \geq 0}$ est une chaîne de Markov,

$$\begin{aligned} p^2(\mathbf{y}_i | \mathbf{y}_{0:i-1}, \mathbf{x}_{0:n}) &= \frac{p^2(\mathbf{y}_{0:i}, \mathbf{x}_{0:n})}{\int p^2(\mathbf{y}_{0:i}, \mathbf{x}_{0:n}) d\mathbf{y}_i} \\ &= \frac{p^2(\mathbf{x}_{i:n}, \mathbf{y}_i | \mathbf{x}_{i-1}, \mathbf{y}_{i-1}) p^2(\mathbf{x}_{0:i-1}, \mathbf{y}_{0:i-1})}{\int p^2(\mathbf{x}_{i:n}, \mathbf{y}_i | \mathbf{x}_{i-1}, \mathbf{y}_{i-1}) p^2(\mathbf{x}_{0:i-1}, \mathbf{y}_{0:i-1}) d\mathbf{y}_i} \\ &= p^2(\mathbf{y}_i | \mathbf{y}_{i-1}, \mathbf{x}_{i-1:n}), \end{aligned}$$

d'où (4.1.4). \square

C.3 Preuve de la Proposition 4.4 (filtrage exact dans les modèles de Markov triplet)

Ce résultat est une application de la technique de filtrage exact présentée dans [95]. Nous considérons un modèle triplet qui vérifie (4.1.60)-(4.1.61). L'objectif est de calculer $p(r_n|\mathbf{y}_{0:n})$ à partir de $p(r_{n-1}|\mathbf{y}_{0:n-1})$ et $E(\mathbf{X}_n|\mathbf{y}_{0:n}, r_n)$ à partir de $E(\mathbf{X}_{n-1}|\mathbf{y}_{0:n-1}, r_{n-1})$. Dans ces modèles particuliers, (\mathbf{Y}_n, R_n) est markovien,

$$p(\mathbf{y}_n, r_n|\mathbf{y}_{0:n-1}, r_{n-1}) = p(\mathbf{y}_n, r_n|\mathbf{y}_{n-1}, r_{n-1}), \quad (\text{C.3.1})$$

$$= p(r_n|r_{n-1})p(\mathbf{y}_n|\mathbf{y}_{n-1}, \mathbf{r}_{n-1:n}). \quad (\text{C.3.2})$$

Par conséquent,

$$p(r_n|\mathbf{y}_{0:n}) \propto \sum_{r_{n-1}} p(r_{n-1}|\mathbf{y}_{0:n-1})p(\mathbf{y}_n, r_n|\mathbf{y}_{n-1}, r_{n-1}). \quad (\text{C.3.3})$$

Ensuite,

$$E(\mathbf{X}_n|\mathbf{y}_{0:n}, r_n) = \sum_{r_{n-1}} p(r_{n-1}|\mathbf{y}_{0:n}, r_n) \times \int \left[\underbrace{\int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{0:n}, \mathbf{r}_{n-1:n}) d\mathbf{x}_n}_{E(\mathbf{X}_n|\mathbf{x}_{n-1}, \mathbf{y}_{0:n}, \mathbf{r}_{n-1:n})} \right] p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n}, \mathbf{r}_{n-1:n}) d\mathbf{x}_{n-1}. \quad (\text{C.3.4})$$

Calculons donc les différentes quantités impliquées dans (C.3.4). D'après (C.3.1), nous avons

$$p(r_{n-1}|\mathbf{y}_{0:n}, r_n) \propto p(r_{n-1}|\mathbf{y}_{0:n-1})p(\mathbf{y}_n, r_n|\mathbf{y}_{n-1}, r_{n-1}). \quad (\text{C.3.5})$$

Puisque $\{\mathbf{X}_n, \mathbf{Y}_n, R_n\}_{n \geq 0}$ est une chaîne de Markov,

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{0:n}, \mathbf{r}_{n-1:n}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{n-1}, \mathbf{y}_n, \mathbf{r}_{n-1:n}); \quad (\text{C.3.6})$$

nous déduisons alors de (4.1.61)

$$E(\mathbf{X}_n|\mathbf{x}_{n-1}, \mathbf{y}_{0:n}, \mathbf{r}_{n-1:n}) = \mathbf{C}_n(\mathbf{r}_{n-1:n})\mathbf{x}_{n-1} + \mathbf{D}_n(\mathbf{r}_{n-1:n}, \mathbf{y}_{n-1:n}). \quad (\text{C.3.7})$$

Ensuite, dans ce modèle triplet particulier

$$p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n}, \mathbf{r}_{n-1:n}) = p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1}, r_{n-1}). \quad (\text{C.3.8})$$

En injectant alors (C.3.5), (C.3.7) et (C.3.8) dans (C.3.4), nous obtenons (4.1.64). La preuve pour le calcul de $E(\mathbf{X}_n \mathbf{X}_n^T|\mathbf{y}_{0:n}, r_n)$ est similaire. \square

C.4 Preuve de la Proposition 4.8

Considérons les modèles de la Proposition 4.5 vérifiant (4.3.26). $D_{\text{KL}}(p^2(\boldsymbol{\xi}_{0:n}, \mathbf{r}_{0:n}), p^1(\boldsymbol{\xi}_{0:n}, \mathbf{r}_{0:n}))$ s'écrit

$$D_{\text{KL}}(p^2(\boldsymbol{\xi}_{0:n}, \mathbf{r}_{0:n}), p^1(\boldsymbol{\xi}_{0:n}, \mathbf{r}_{0:n})) = \sum_{\mathbf{r}_{0:n}} p^1(\mathbf{r}_{0:n}) D_{\text{KL}}(p^2(\boldsymbol{\xi}_{0:n}|\mathbf{r}_{0:n}), p^1(\boldsymbol{\xi}_{0:n}|\mathbf{r}_{0:n})) \quad (\text{C.4.1})$$

(la loi des sauts est identique dans les deux modèles). D'après la Proposition 4.6, $p^1(\mathbf{r}_{0:n}) = p^2(\mathbf{r}_{0:n})$ et ne dépend donc pas de $\{\mathbf{F}_n^2(\mathbf{r}_{n-1:n})\}_{n \geq 1}$; par conséquent, nous étudions $D_{\text{KL}}(p^2(\boldsymbol{\xi}_{0:n}|\mathbf{r}_{0:n}), p^1(\boldsymbol{\xi}_{0:n}|\mathbf{r}_{0:n}))$, qui peut se réécrire, en utilisant les propriétés markoviennes de $\boldsymbol{\xi}_{0:n}$ conditionnellement à $\mathbf{r}_{0:n}$,

$$D_{\text{KL}}(p^2(\boldsymbol{\xi}_{0:n}|\mathbf{r}_{0:n}), p^1(\boldsymbol{\xi}_{0:n}|\mathbf{r}_{0:n})) = \sum_{j=1}^n \int p^2(\boldsymbol{\xi}_{j-1}|\mathbf{r}_{0:j-1}) D_{\text{KL}}(p^2(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j}), p^1(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j})) d\boldsymbol{\xi}_{j-1}, \quad (\text{C.4.2})$$

où, d'après les Propositions 4.5 et 4.6, $p^2(\boldsymbol{\xi}_{j-1}|\mathbf{r}_{0:j-1}) = p^1(\boldsymbol{\xi}_{j-1}|\mathbf{r}_{0:j-1})$, et ne dépend donc pas des matrices $\mathbf{F}_j^2(\mathbf{r}_{j-1:j})$. Nous minimisons donc pour tout j , $1 \leq j \leq n$

$$D_{\text{KL}}(p^2(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j}), p^1(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j})).$$

Par conditionnement,

$$p^1(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j}) = p^1(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}_j, r_j) p^1(\mathbf{y}_j|\mathbf{x}_{j-1}, r_j), \quad (\text{C.4.3})$$

$$p^2(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j}) = p^2(\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j, \mathbf{r}_{j-1:j}) p^2(\mathbf{y}_j|\mathbf{y}_{j-1}, \mathbf{r}_{j-1:j}) \quad (\text{C.4.4})$$

(rappelons que $p^2(\cdot)$ vérifie (4.3.26)) où

$$p^2(\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j, \mathbf{r}_{j-1:j}) = \mathcal{N}(\mathbf{x}_j; \mathbf{m}_j^{\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j}; \mathbf{P}_j^{\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j}), \quad (\text{C.4.5})$$

$$\begin{aligned} \mathbf{m}_j^{\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j} &= (\mathbf{F}_j(r_j) - \mathbf{F}_j^2(\mathbf{r}_{j-1:j})\mathbf{H}_{j-1}(r_{j-1}))\mathbf{x}_{j-1} + \mathbf{F}_j^2(\mathbf{r}_{j-1:j})\mathbf{y}_{j-1} \\ &\quad + (\boldsymbol{\Sigma}_j^{21}(\mathbf{r}_{j-1:j}))^T (\boldsymbol{\Sigma}_j^{22}(\mathbf{r}_{j-1:j}))^{-1} (\mathbf{y}_j - \mathbf{H}_j^2(\mathbf{r}_{j-1:j})\mathbf{y}_{j-1}), \end{aligned} \quad (\text{C.4.6})$$

$$\mathbf{P}_j^{\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j} = \boldsymbol{\Sigma}_j^{11}(\mathbf{r}_{j-1:j}) - (\boldsymbol{\Sigma}_j^{21}(\mathbf{r}_{j-1:j}))^T (\boldsymbol{\Sigma}_j^{22}(\mathbf{r}_{j-1:j}))^{-1} \boldsymbol{\Sigma}_j^{21}(\mathbf{r}_{j-1:j}), \quad (\text{C.4.7})$$

où $\boldsymbol{\Sigma}_j^{11}(\mathbf{r}_{j-1:j})$, $\boldsymbol{\Sigma}_j^{21}(\mathbf{r}_{j-1:j})$ et $\boldsymbol{\Sigma}_j^{22}(\mathbf{r}_{j-1:j})$ sont définies par (4.3.15)-(4.3.17).

La DKL entre les lois $p^2(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j})$ et $p^1(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j})$ s'écrit

$$D_{\text{KL}}(p^2(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j}), p^1(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j})) = \int p^2(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j}) \log \left(\frac{p^2(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j})}{p^1(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j})} \right) d\boldsymbol{\xi}_j. \quad (\text{C.4.8})$$

Le terme (C.4.8) se réécrit par ailleurs

$$D_{\text{KL}}(p^2(\mathbf{y}_j|\mathbf{y}_{j-1}, \mathbf{r}_{j-1:j}), p^1(\mathbf{y}_j|\mathbf{x}_{j-1}, \mathbf{r}_{j-1:j})) + \int p^2(\mathbf{y}_j|\mathbf{y}_{j-1}, \mathbf{r}_{j-1:j}) \times D_{\text{KL}}(p^2(\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j, \mathbf{r}_{j-1:j}), p^1(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}_j, \mathbf{r}_{j-1:j})) d\mathbf{y}_j$$

qui est minimum lorsque $p^2(\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j, \mathbf{r}_{j-1:j}) = p^1(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}_j, r_j)$. D'après la Proposition 4.6, nous avons $p^2(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}_j, \mathbf{r}_{j-1:j}) = p^1(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}_j, \mathbf{r}_{j-1:j}) = p^1(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{y}_j, r_j)$; par conséquent

$$D_{\text{KL}}(p^2(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j}), p^1(\boldsymbol{\xi}_j|\boldsymbol{\xi}_{j-1}, \mathbf{r}_{j-1:j}))$$

est minimum lorsque $p^2(\mathbf{x}_j|\boldsymbol{\xi}_{j-1}, \mathbf{y}_j, \mathbf{r}_{j-1:j})$ ne dépend pas de \mathbf{y}_{j-1} . À partir de (C.4.6), cette condition est vérifiée pour

$$\mathbf{F}_j^2(\mathbf{r}_{j-1:j}) = \mathbf{Q}_j(r_j)(\mathbf{H}_j(r_j))^T \left[\mathbf{R}_j(r_j) + \mathbf{H}_j(r_j)\mathbf{Q}_j(r_j)(\mathbf{H}_j(r_j))^T \right]^{-1} \mathbf{H}_j^2(\mathbf{r}_{j-1:j}). \quad (\text{C.4.9})$$

□

Un pisteur robuste basé sur une approche CPHD

Sommaire

D.1	L'implémentation par mélange de gaussiennes du CPHD	167
D.2	Un pisteur robuste basé sur une approche CPHD	169
D.3	Simulations	171

Dans les différents chapitres de ce travail, nous nous sommes focalisés sur le problème du filtrage multi-objets, qui permet d'estimer le nombre d'objets et leur état caché. Cependant, le filtre PHD ne permet pas de différencier l'historique des trajectoires des objets suivis. Dans cette annexe, nous nous proposons de construire un pisteur robuste déduit du filtre CPHD, qui permet d'améliorer l'estimation du nombre d'objets et des états cachés [82]. Pour des enjeux de clarté, nous supposons que le modèle suivi par chaque état caché et son observation associée est un modèle HMC linéaire et gaussien,

$$f_{n|n-1}(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n), \quad (\text{D.0.1})$$

$$g_n(\mathbf{z}|\mathbf{x}_n) = \mathcal{N}(\mathbf{z}; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n). \quad (\text{D.0.2})$$

Nous supposons également que les probabilités de survie $p_{s,n}$ et de détection $p_{d,n}$ ne dépendent pas du vecteur d'état \mathbf{x} .

Dans un premier temps, nous rappelons brièvement l'implémentation par mélange de gaussiennes du filtre CPHD, puis à partir de cette implémentation nous construisons un pisteur permettant de différencier l'historique des trajectoires de chaque objet.

D.1 L'implémentation par mélange de gaussiennes du CPHD

L'assouplissement des hypothèses de Poisson **A.1** et **A.2** (voir Paragraphe 1.2.2) n'empêche pas la propagation du PHD mais nécessite de connaître la loi de probabilité $p_{\gamma,n}(k)$ du nombre d'objets naissants, et celle du nombre de mesures de fausses alarmes, $p_{\kappa,n}(k)$. Nous supposons que le PHD à l'instant $n - 1$ et l'intensité des naissances à l'instant n sont des mélanges gaussiens,

$$v_{n-1}(\mathbf{x}) = \sum_{i=1}^{J_{n-1}} w_{n-1}^{(i)} \mathcal{N}(\mathbf{x}, \mathbf{m}_{n-1}^{(i)}, \mathbf{P}_{n-1}^{(i)}), \quad (\text{D.1.1})$$

$$\gamma_n(\mathbf{x}) = \sum_{i=1}^{J_{\gamma_n}} w_{n-1}^{(i)} \mathcal{N}(\mathbf{x}, \mathbf{m}_{\gamma_n}^{(i)}, \mathbf{P}_{\gamma_n}^{(i)}). \quad (\text{D.1.2})$$

L'assouplissement des hypothèses de Poisson modifie les équations de propagation du PHD, qui reposent à présent sur la loi de probabilité $p_{n-1}(k)$ du nombre d'objets, qu'il faut donc également propager. Pour les

modèles linéaires et gaussiens, la propagation du PHD repose également sur une étape de prédiction et sur une étape de mise à jour rappelées ci-après [123].

D.1.1 Prédiction CPHD

L'équation de prédiction du PHD reste inchangée par rapport à l'équation de prédiction du filtre PHD (1.2.10) car les hypothèses de Poisson n'interviennent que dans l'étape de mise à jour du filtre PHD [80]. En posant $C_j^l = l!/(j!((l-j)!))$, on obtient [82] [123]

$$p_{n|n-1}(k) = \sum_{j=0}^k p_{\gamma,n}(k-j) \sum_{l=j}^{\infty} C_j^l p_{k-1}(l) p_{s,n} (1-p_{s,n})^{l-j}, \quad (\text{D.1.3})$$

$$v_{n|n-1}(\mathbf{x}) = \sum_{i=1}^{J_{n-1}} p_{s,n} w_{n-1}^i \mathcal{N}(\mathbf{x}; \mathbf{m}_{n|n-1}^{(i)}; \mathbf{P}_{n|n-1}^{(i)}) + \gamma_n(\mathbf{x}). \quad (\text{D.1.4})$$

D.1.2 Mise à jour CPHD

Repartons de (D.1.3) et du mélange de gaussiennes (D.1.4) que nous réécrivons sous la forme compacte

$$v_{n|n-1}(\mathbf{x}) = \sum_{i=1}^{J_{n|n-1}} w_{n|n-1}^i \mathcal{N}(\mathbf{x}; \mathbf{m}_{n|n-1}^{(i)}; \mathbf{P}_{n|n-1}^{(i)}); \quad (\text{D.1.5})$$

alors

$$p_n(k) = \frac{\Psi_n^0[w_{n|n-1}, \mathcal{Z}_n] p_{n|n-1}(k)}{\langle \Psi_n^0[w_{n|n-1}, \mathcal{Z}_n], p_{n|n-1} \rangle}, \quad (\text{D.1.6})$$

$$v_n(\mathbf{x}) = (1-p_{d,n}) \frac{\langle \Psi_n^1[w_{n|n-1}, \mathcal{Z}_n], p_{n|n-1} \rangle}{\langle \Psi_n^0[w_{n|n-1}, \mathcal{Z}_n], p_{n|n-1} \rangle} v_{n|n-1}(\mathbf{x}) + \sum_{\mathbf{z} \in \mathcal{Z}_n} \sum_{j=1}^{J_{n|n-1}} w_n^{(j)}(\mathbf{z}) \mathcal{N}(\mathbf{x}; \mathbf{m}_n^j(\mathbf{z}); \mathbf{P}_n^{(j)}) \quad (\text{D.1.7})$$

où

$$\Psi_n^u[w, \mathcal{Z}](k) = \sum_{j=0}^{\min(|\mathcal{Z}|, k)} (|\mathcal{Z}| - j)! p_{\kappa,n} (|\mathcal{Z}| - j) P_{j+u}^n \frac{(1-p_{d,n})^{k-(j+u)}}{\langle 1, w \rangle^{j+u}} e_j(\Xi_n(w, \mathcal{Z})), \quad (\text{D.1.8})$$

$$\Xi_n(w, \mathcal{Z}) = \left\{ \frac{\langle 1, \kappa \rangle}{\kappa_n(\mathbf{z})} p_{d,n} w^T q_n(\mathbf{z}) : \mathbf{z} \in \mathcal{Z} \right\}, \quad (\text{D.1.9})$$

$$w_{n|n-1} = [w_{n|n-1}^{(1)}, \dots, w_{n|n-1}^{(J_{n|n-1})}]^T, \quad (\text{D.1.10})$$

$$q_n(\mathbf{z}) = [q_n^{(1)}(\mathbf{z}), \dots, q_n^{(J_{n|n-1})}(\mathbf{z})]^T, \quad (\text{D.1.11})$$

$$q_n^{(i)}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{H}_n \mathbf{m}_{n|n-1}^{(i)}, \mathbf{R}_n + \mathbf{H}_n \mathbf{P}_{n|n-1}^{(i)} \mathbf{H}_n^T), \quad (\text{D.1.12})$$

$$w_n^{(i)}(\mathbf{z}) = p_{d,n} w_{n|n-1}^{(i)} q_n^{(i)}(\mathbf{z}) \frac{\langle \Psi_n^1[w_{n|n-1}, \mathcal{Z}_n \setminus \{\mathbf{z}\}], p_{n|n-1} \rangle}{\langle \Psi_n^0[w_{n|n-1}, \mathcal{Z}_n], p_{n|n-1} \rangle} \times \frac{\langle 1, \kappa \rangle}{\kappa_n(\mathbf{z})} \quad (\text{D.1.13})$$

où $P_j^k = \frac{k!}{(k-j)!}$, $\langle \cdot, \cdot \rangle$ est le produit scalaire ($\langle f, g \rangle = \int f(\mathbf{x})g(\mathbf{x})d\mathbf{x}$ ou $\langle u, v \rangle = \sum_{n=-\infty}^{+\infty} u(n)v(n)$); enfin, pour $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_m\}$ les fonctions symétriques élémentaires $e_j(\mathcal{Z})$ sont définies par $e_0(\mathcal{Z}) = 1$, $e_1(\mathcal{Z}) = \mathbf{z}_1 + \mathbf{z}_2 + \dots + \mathbf{z}_m$, $e_2(\mathcal{Z}) = \mathbf{z}_1\mathbf{z}_2 + \mathbf{z}_3\mathbf{z}_4 + \dots$, $e_m(\mathcal{Z}) = \mathbf{z}_1\mathbf{z}_2\mathbf{z}_3 \dots \mathbf{z}_m$.

Remarque D.1. Contrairement au filtre PHD, la complexité de l'étape de mise à jour du filtre CPHD est cubique en nombre d'observations. Par conséquent, l'utilisation pratique du filtre est compromise lorsque le nombre d'observations est élevé. Notre pisteur sera associé à des techniques de « gating » (ou fenêtrage) permettant de réduire le nombre d'observations en entrée de cette étape de mise à jour [128].

D.2 Un pisteur robuste basé sur une approche CPHD

Dans le contexte de l'implémentation par mélange de gaussiennes du filtre PHD, une solution simple visant à différencier les trajectoires des différents objets consiste à attribuer aux différentes gaussiennes du mélange une étiquette ou un label [91]. Plus précisément, ce label est alors inchangé pendant les étapes de prédiction et de mise à jour et de nouveaux labels sont attribués aux gaussiennes du mélange $\gamma_n(\mathbf{x})$. Néanmoins, ce schéma n'est pas efficace lorsque les objets suivis ont tendance à être proches les uns des autres :

1. des objets différents peuvent être représentés par des gaussiennes dont le label est identique ; à l'opposé, un même objet peut être représenté par des gaussiennes dont le label est différent ;
2. les objets peuvent échanger leur label, et il n'est donc plus possible d'interpréter correctement le comportement des trajectoires des différents objets.

Afin d'améliorer la gestion des labels des gaussiennes, nous introduisons une méthode d'affectation couplée avec une méthode de « clustering » dont le but est d'améliorer les performances des estimateurs. Nous introduisons également une méthode de « gating » afin de diminuer la complexité de l'algorithme.

Plus précisément, un objet potentiel, appelé « piste » est représenté à l'instant $n - 1$ par une gaussienne labellisée, c'est à dire un vecteur d'état $m_{n-1}^{(i)}$, une matrice de covariance $P_{n-1}^{(i)}$, un poids $w_{n-1}^{(i)}$ et un label (unique) $\tau_{n-1}^{(i)}$. Après l'étape de prédiction, le PHD est composé de $J_{n|n-1}$ gaussiennes ; nous commençons par regrouper les gaussiennes d'une même région (clustering) afin d'appliquer, pour chaque région, une étape de mise à jour d'un filtre CPHD spécifique dont les paramètres dépendent de la région considérée. Pour cela, une région est caractérisée par un mélange gaussien, une intensité des mesures de fausses alarmes, et une loi de probabilité du nombre d'objets. Tous ces paramètres spécifiques sont nécessaires afin de pouvoir appliquer les équations de mise à jour du filtre CPHD (D.1.7)-(D.1.13).

À partir du mélange prédit $v_{n|n-1}(\mathbf{x})$, nous expliquons ci-dessous comment créer les différentes régions et comment déduire la loi de probabilité du nombre d'objets et l'intensité des mesures de fausses alarmes associées. Nous partons donc de l'étape de prédiction du CPHD,

$$v_{n|n-1}(\mathbf{x}) = \sum_{i=1}^{J_{n|n-1}} w_{n|n-1}^i \mathcal{N}(\mathbf{x}; \mathbf{m}_{n|n-1}^{(i)}; \mathbf{P}_{n|n-1}^{(i)}). \quad (\text{D.2.1})$$

D.2.1 Clustering

Cette étape consiste à regrouper les gaussiennes du mélange $v_{n|n-1}(\mathbf{x})$ appartenant à une même région. Étant donné un seuil T_c , nous évaluons alors la distance entre la i -ème gaussienne et les autres gaussiennes du mélange :

$$d_c = \left[\mathbf{m}_{n|n-1}^{(i)} - \mathbf{m}_{n|n-1}^{(j)} \right]^T \left[\mathbf{P}_{n|n-1}^{(i)} \right]^{-1} \left[\mathbf{m}_{n|n-1}^{(i)} - \mathbf{m}_{n|n-1}^{(j)} \right]. \quad (\text{D.2.2})$$

Si $d_c \leq T_c$, nous considérons alors que la i -ème et j -ème gaussienne appartiennent au même cluster. Nous recommençons cette étape jusqu'à ce que toutes les gaussiennes soient affectées à un cluster.

D.2.2 Gating

Si l'on souhaite appliquer une étape de mise à jour du filtre CPHD pour chaque cluster, il n'est pas envisageable de prendre en compte toutes les observations disponibles. Nous allons donc associer à chaque cluster un ensemble d'observations. On se donne alors un seuil T_g et on évalue pour toutes les observations disponibles \mathcal{Z}_n et pour toutes les gaussiennes du cluster considéré la distance

$$d_g(\mathbf{z}) = [\mathbf{z} - \mathbf{H}_n \mathbf{m}_{n|n-1}^{(i)}]^T [\mathbf{R}_n + \mathbf{H}_n \mathbf{P}_{n|n-1}^{(i)} \mathbf{H}_n^T]^{-1} [\mathbf{z} - \mathbf{H}_n \mathbf{m}_{n|n-1}^{(i)}] \quad (\text{D.2.3})$$

Si $d_g(\mathbf{z}) < T_g$, pour au moins une des gaussiennes du cluster considéré, alors nous considérons que la mesure \mathbf{z} appartient à ce cluster.

D.2.3 Loi de probabilité du nombre de cibles

Afin d'appliquer les équations (D.1.7)-(D.1.13) à chaque cluster, nous avons besoin de la loi de probabilité locale du nombre d'objets du cluster considéré. Nous rajoutons l'hypothèse que la densité multi-objets $p(\mathcal{X}_n | \mathcal{Z}_{0:n-1})$ est multi-Bernoulli [81], c'est à dire qu'il est possible de déduire la loi de probabilité du nombre d'objets à partir des poids des N_j gaussiennes du cluster considéré, $w_{n|n-1}^{(1)}, w_{n|n-1}^{(2)}, \dots, w_{n|n-1}^{(N_j)}$. Cette distribution de probabilité locale est alors donnée par $p_{n|n-1}(k)$ tel que $p_{n|n-1}(0) = \prod_{i=1}^{N_j} (1 - w_{n|n-1}^{(i)})$, $p_{n|n-1}(1) = \sum_{j=1}^{N_j} w_{n|n-1}^{(j)} \prod_{i \neq j} (1 - w_{n|n-1}^{(i)})$, $p_{n|n-1}(2) = \sum_{j=1}^{N_j} w_{n|n-1}^{(j)} \sum_{p \neq j} w_{n|n-1}^{(p)} \prod_{i \neq j \neq p} (1 - w_{n|n-1}^{(i)})$ \dots et $p(N_j) = \prod_{i=1}^{N_j} w_{n|n-1}^{(i)}$.

D.2.4 Intensité des mesures de fausses alarmes

Il nous reste à redéfinir l'intensité locale des mesures de fausses alarmes, qui est spécifique à chaque cluster. Pour chaque cluster, le volume de la région est approché par la somme des volumes de chaque gaussienne, c'est à dire

$$V_{new} = \sum_{i=1}^{N_j} V_{new}^{(i)}, \quad V_{new}^{(i)} = c_{\mathbf{z}} \sqrt{\det(\mathbf{R}_n + \mathbf{H}_n \mathbf{P}_{n|n-1}^{(i)} \mathbf{H}_n^T)} \gamma^{n_{\mathbf{z}}/2}, \quad (\text{D.2.4})$$

où $c_{\mathbf{z}}$ est le volume de l'hypersphère unité de dimension q (la dimension d'une observation) et γ un paramètre définissant la taille de la fenêtre [70]. Si la répartition des mesures de fausses alarmes est uniforme sur la région observée, son intensité est donnée par $\kappa_n(\mathbf{z}) = M_n u(\mathbf{z})$, où M_n est le nombre moyen de fausses alarmes, et $u(\mathbf{z})$ la densité de la loi uniforme sur la région observée, de volume V . Dans ce cas, pour chaque cluster, on a $\kappa_{new}(\mathbf{z}) = M_{new} u_{new}(\mathbf{z})$ où $M_{new} = \frac{M}{V} V_{new}$, et $u_{new}(\mathbf{z})$ représente la densité uniforme sur V_{new} . La quantité V_{new} dépend évidemment du cluster considéré.

D.2.5 Gestion des pistes

Pour chaque gaussienne d'un cluster donné, nous appliquons l'étape de mise à jour du filtre CPHD (D.1.7)-(D.1.13) dont les paramètres (intensité des mesures de fausses alarmes, loi de probabilité prédite du nombre d'objets et observations disponibles) sont donnés dans les Paragraphes D.2.1-D.2.4.

D'après (D.1.7)-(D.1.13), un poids est affecté à chaque couple gaussienne-observation (terme de non détection « θ » inclus). Si une observation n'appartient pas au cluster considéré, nous lui affectons un poids

	Piste prédite 1	Piste prédite 2	...	Piste prédite $J_{n n-1}$
\mathbf{z}_1	$w_n^{(1)}(\mathbf{z}_1)$	$w_n^{(2)}(\mathbf{z}_1)$...	$w_n^{(J_{n n-1})}(\mathbf{z}_1)$
\mathbf{z}_2	$w_n^{(1)}(\mathbf{z}_2)$	$w_n^{(2)}(\mathbf{z}_2)$...	$w_n^{(J_{n n-1})}(\mathbf{z}_2)$
\vdots	\vdots	\vdots	\vdots	\vdots
\mathbf{z}_m	$w_n^{(1)}(\mathbf{z}_m)$	$w_n^{(2)}(\mathbf{z}_m)$...	$w_n^{(J_{n n-1})}(\mathbf{z}_m)$
θ	$w_n^{(1)}(\theta)$	$w_n^{(2)}(\theta)$...	$w_n^{(J_{n n-1})}(\theta)$

TABLE D.1 – Matrice d’affectation : on cherche à maximiser la somme des poids

nul avec toutes les gaussiennes du cluster. Nous obtenons alors une matrice d’affectation, dont la forme est donnée par la Table D.1.

Afin d’éviter les problèmes multiples dus à l’introduction des labels, nous utilisons alors une méthode d’affectation 2-D dont l’objectif est d’affecter à chaque piste (ou gaussienne) une unique observation ; le critère utilisé pour réaliser cette affectation est la maximisation de la somme globale des poids. Par conséquent, nous utilisons l’algorithme de Munkres [88] [111], spécialement conçu pour ce critère d’optimisation, et qui assure qu’un objet est représenté par une unique gaussienne. Les moyennes et matrices de covariance de chaque gaussienne sont actualisées en utilisant les équations de mise à jour classiques avec l’observation affectée. Si le terme de non détection est affecté à une gaussienne prédite, sa moyenne et sa matrice de covariance restent identiques.

D.2.6 Initialisation des pistes

Les nouvelles pistes sont initialisées à partir de l’intensité des objets naissants γ_n : à chaque gaussienne du mélange γ_n dans (D.1.1) est affecté un nouveau label qui n’a pas été utilisé jusqu’à l’instant considéré. Si nous n’avons pas d’information a priori sur les zones d’apparition des objets naissants, il est possible d’initialiser une nouvelle piste autour de chaque mesure disponible comme cela a été suggéré dans [101].

D.3 Simulations

Notre approche locale (Clustering 2-D CPHD) est comparée à un filtre CPHD classique auquel on a ajouté la méthode d’affectation 2-D du Paragraphe D.2.5 ainsi qu’à un pisteuse CPHD basique (Filtre CPHD, avec labellisation des gaussiennes [91]), en utilisant un scénario de croisement de 5 cibles présenté sur la Figure D.1(a)-D.1(b). Chaque objet et son observation associée suivent un modèle linéaire et gaussien identique à celui utilisé lors de la simulation du Paragraphe 3.3.4, dans laquelle $\sigma_v = 1m^2/sec^3$ et $\sigma_x = \sigma_y = 10m$. Le nombre moyen de fausses alarmes est de 30. Le volume de la région observée est $V = 10^6$ et nous choisissons $T_c = 3$ et $T_g = 9$.

Un exemple de résultat de notre algorithme de pistage est présenté sur la Figure D.3. Afin d’étudier l’apport de notre méthode, nous réalisons 200 simulations et nous moyennons à chaque instant la distance OSPA définie dans (1.2.13) avec $\bar{p} = 2$ et $\bar{c} = 500$. Néanmoins, la distance OSPA permet d’évaluer les performances d’algorithmes de filtrage, et non de pistage, dans lesquels on prend également en compte le label affecté à chaque état estimé. Nous utilisons donc une généralisation de la distance OSPA, qui a été proposée dans [102] afin d’évaluer les performances des algorithmes de pistage. Cette généralisation repose

sur la redéfinition de $d^{\bar{c}}(\mathbf{x}, \hat{\mathbf{x}})$ dans (1.2.13), qui s'écrit à présent

$$d^{\bar{c}}(\mathbf{x}, \hat{\mathbf{x}}) = \min(\bar{c}, \|\mathbf{x} - \hat{\mathbf{x}}\| + \alpha \bar{\delta}(s, t)), \quad (\text{D.3.1})$$

où s et t sont respectivement les labels des objets dont les vecteurs d'état sont \mathbf{x} et $\hat{\mathbf{x}}$, $\bar{\delta}(s, t) = 1$ si $s \neq t$ et 0 sinon, et où α est un paramètre qui pénalise une erreur sur l'estimation du label de l'objet.

La Figure D.3 représente les performances de notre méthode par rapport à un pisteur CPHD classique en termes de distance OSPA, pour le filtrage (Fig.D.3(a)) et le pistage (Fig.D.3(b)). Notre approche permet une meilleure estimation du nombre d'objets et de leur vecteur d'état comme le montre la Figure D.3(a). Plus important, notre méthode locale améliore la robustesse du pistage comme le montre l'écart entre les trois algorithmes à la Figure D.3(b).

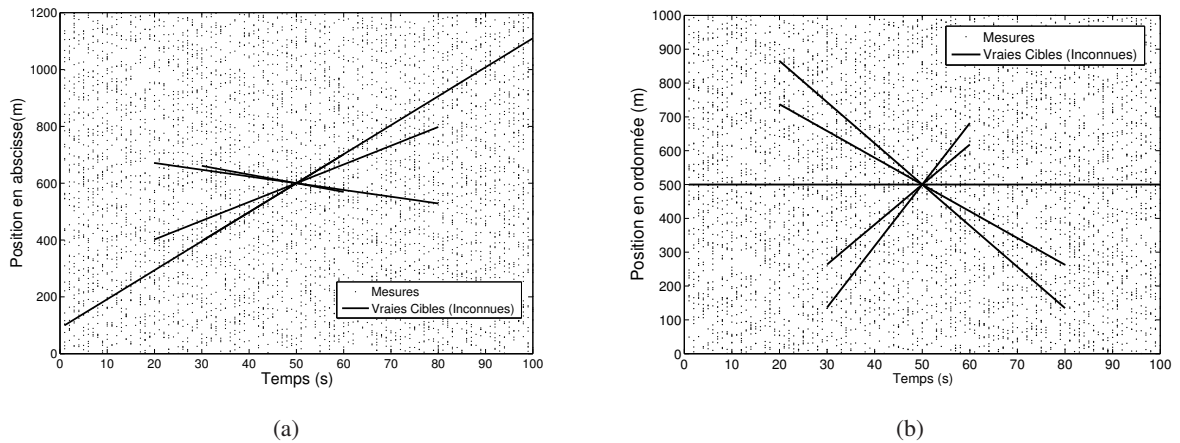


FIGURE D.1 – Croisement de 5 objets - Mesures et Positions des objets en abscisse (a) et en ordonnée (b)

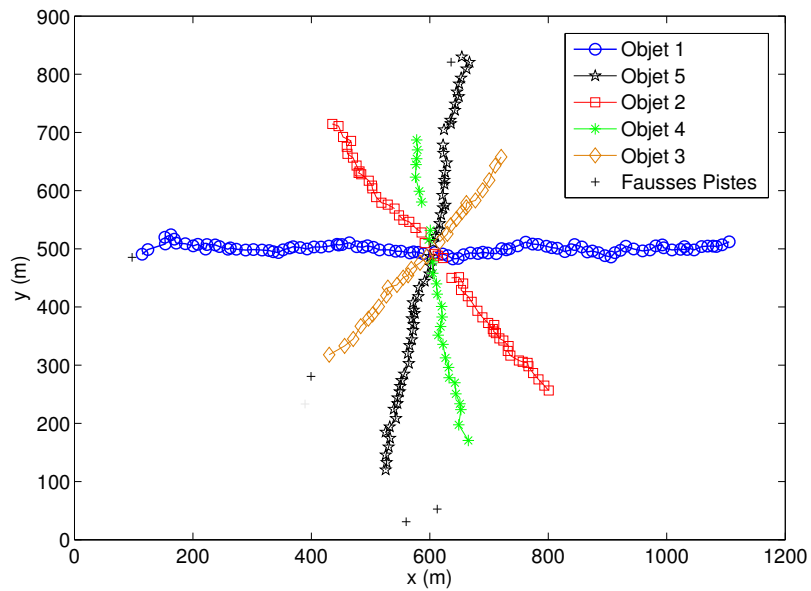
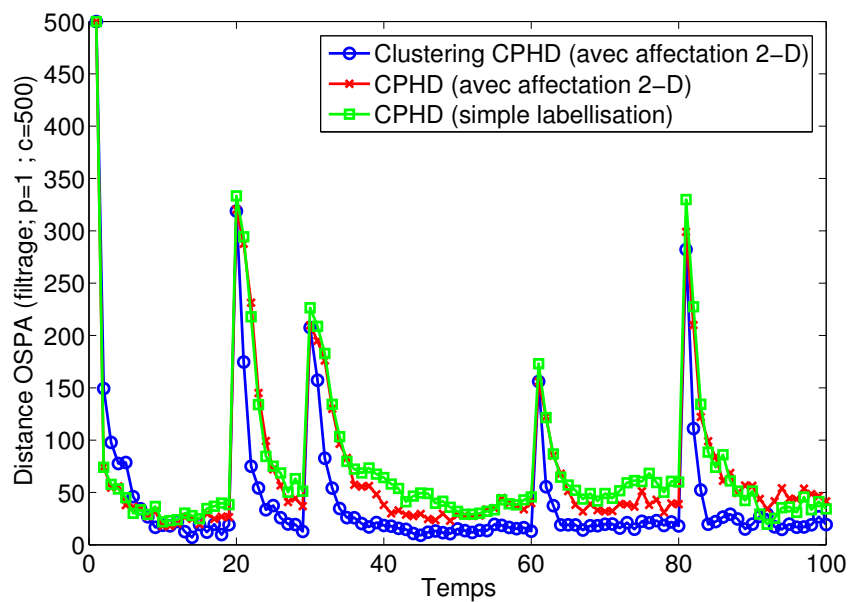
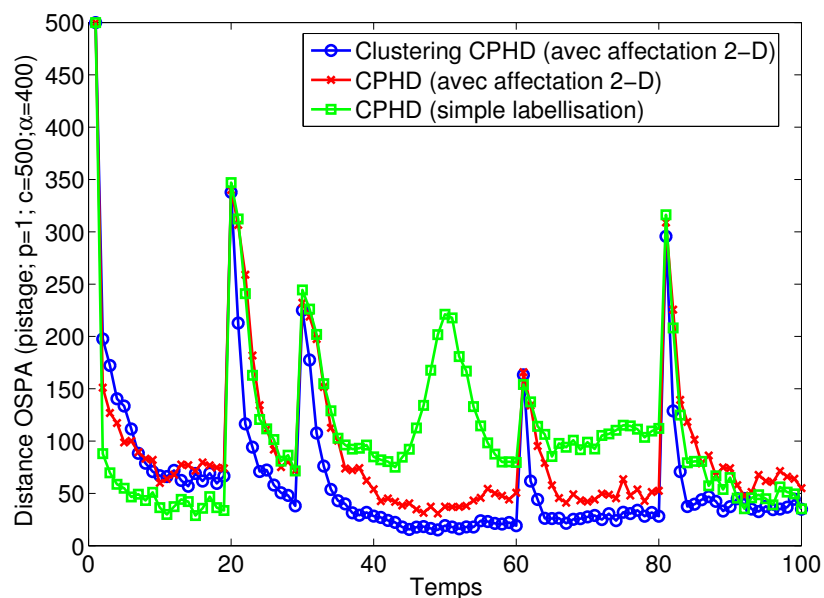


FIGURE D.2 – Croisement de 5 objets - Trajectoires pistées par notre méthode locale. Chaque trajectoire est représentée par une couleur et un symbole différents.



(a)



(b)

FIGURE D.3 – Distances OSPA pour le scénario de croisement de 5 objets - Filtrage (a) - Pistage (b). Notre méthode locale améliore non seulement les performances du filtrage comparé au filtre CPHD classique mais aussi la robustesse du pistage, surtout lorsque les objets sont proches autour de $n = 50$.

Bibliographie

- [1] N. Abbassi, D. Benboudjema, and W. Pieczynski. Kalman filtering approximations in triplet Markov Gaussian switching models. In *IEEE Workshop on Statistical Signal Processing*, Nice, France, June. 28-30 2011. 128, 129
- [2] G. Ackerson and K.S. Fu. On state estimation in switching environments. *IEEE Transactions on Automatic Control*, 4 :429–434, February 1970. 21
- [3] B. Ait-El-Fquih and F. Desbouvries. Kalman filtering in triplet Markov chains. *IEEE Transactions on Signal Processing*, 54(8) :2957–63, August 2006. 128
- [4] H. Akashia and H. Kumamoto. Random sampling approach to state estimation in switching environments. *Automatica*, 13(4) :429–434, July 1977. 21, 22
- [5] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, New Jersey, 1979. 20
- [6] C. Andrieu, M. Davy, and A. Doucet. Efficient particle filtering for jump Markov systems. *IEEE transactions on Signal Processing*, 51 :1762–1770, 2002. 49, 52, 107, 108
- [7] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B*, 72(3) :269–342, 2010. 98
- [8] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear / non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2) :174–188, February 2002. 35
- [9] S. Asmussen and P. W. Glynn. *Stochastic Simulation : Algorithms and Analysis*. Springer-Verlag, New York, NY, 2007. 77
- [10] B. Balakumar, A. Sinha, and T. Kirubarajan. PHD filtering for tracking an unknown number of sources using an array of sensors. In *Proceedings of SPIE*, volume 43–48, July 17-20 2005. 19
- [11] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987. 16, 23
- [12] Y Bar-Shalom and X. R. Li. *Multitarget-multisensor tracking : principles and techniques*. YBS, 1995. 19
- [13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley and sons, New-York, 2001. 16, 123
- [14] E. Bellone, J. P. Hughes, and P. Guttorp. A hidden Markov model for downscaling synoptic atmospheric patterns to precipitation amounts. *Climate research*, 15(1) :1–12, 2000. 19

-
- [15] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985. 20
- [16] S. Blackman. *Multiple-target tracking with radar applications*. Artech House radar library. Norwood, Mass. Artech House, 1986. 16, 23
- [17] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8) :780–783, 1988. 140, 141, 147
- [18] O. Cappé, S. J. Godsill, and É. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proc. of the IEEE*, 95(5) :899–924, May 2007. 21
- [19] O. Cappé, É. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer-Verlag, 2005. 19, 20, 35, 36, 54, 61
- [20] F. Caron, M. Davy, E. Duflos, and P. Vanheeghe. Particle filtering for multisensor data fusion with switching observation models : Application to land vehicle positioning. *IEEE Transactions on Signal Processing*, 55(6-1) :2703–2719, 2007. 21
- [21] F. Caron, A. Doucet, and R. Gottardo. On-line changepoint detection and parameter estimation with application to genomic data. *Statistics and Computing*, 22(2) :579–595, March 2012. 19
- [22] J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for non-linear problems. In *IEE Proceedings Radar Sonar Navigation*, pages 2–7, 2004. 36
- [23] R. Chen and J.S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society : Series B*, 62 :493–508, 2000. 17, 19, 21, 28, 78, 79, 141, 146
- [24] N. Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6) :2385–2411, 2004. 17, 22, 27, 28, 78
- [25] D. Clark and J. Bell. Bayesian multiple target tracking in forward scan sonar images using the PHD. *IEE Radar, Sonar and Navigation*, 152 :327–334, 2005. 19
- [26] D. Clark and J. Bell. Convergence results for the particle-PHD filter. *IEEE Transactions on Signal Processing*, 54, 2006. 26, 27, 81, 130
- [27] D. Clark and B.-N. Vo. Convergence Analysis of the Gaussian Mixture PHD Filter. *IEEE Transactions on Signal Processing*, 55(4) :1204–1212, 2007. 130
- [28] D. Clark, B.-T. Vo, and B.-N. Vo. Gaussian particle implementations of probability hypothesis density filters. In *IEEE Aerospace Conference*, 2007. 81
- [29] D. Clark, B.-T. Vo, B.-N. Vo, and S. Godsill. Gaussian mixture implementations of Probability Hypothesis Density filters for non-linear dynamical models. In *IET seminar on Target Tracking and Data Fusion*, pages 21–28, April 2008. 81
- [30] J. Cornebise. *Adaptive Sequential Monte Carlo Methods*. PhD thesis, Télécom ParisTech, 2010. 38, 49
- [31] J. Cornebise, É. Moulines, and J. Olsson. Adaptive methods for sequential importance sampling with application to state-space models. *Statistics and Computing*, 18(4) :461–480, 2008. 36, 38, 59
- [32] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3) :736–46, March 2002. 27
-

- [33] P. del Moral. *Feynman-Kac formulae. Genealogical and interacting particle systems with applications*. Probability and its applications. Springer, New-York, 2004. 15
- [34] P. Del Moral and J. Jacod. Interacting particle filtering with discrete observations. In A. Doucet, S. J. Godsill, and C. Andrieu, editors, *On sequential Monte Carlo sampling methods for Bayesian filtering*. Springer, 1999. 36
- [35] A. P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (B)*, 39(1) :1–38, 1977. 156
- [36] S. Derrode and W. Pieczynski. Signal and image segmentation using pairwise Markov chains. *IEEE Transactions on Signal Processing*, 52(9) :2477–89, 2004. 121
- [37] S. Derrode and W. Pieczynski. Exact fast computation of optimal filter in gaussian switching linear systems. *IEEE Signal Processing Letters*, 20(7) :701–704, 2013. 145
- [38] F. Desbouvries and W. Pieczynski. Modèles de Markov triplet et filtrage de Kalman. *Comptes Rendus de l'Académie des Sciences - Mathématiques*, 336(8) :667–670, 2003. in French. 128
- [39] F. Desbouvries and W. Pieczynski. Particle filtering in pairwise and triplet Markov chains. In *Proc. IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing*, Grado-Gorizia, Italy, June 8-11 2003. 121, 123, 128
- [40] R. Douc, O. Cappé, and É. Moulines. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia, September 2005. 37
- [41] R. Douc, É. Moulines, and J. Olsson. Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics*, 29(1) :1–28, 2009. 38, 47, 49, 98
- [42] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer Verlag, New York, 2001. 35
- [43] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10 :197–208, 2000. 15, 21, 35, 37, 39, 67, 69, 78, 87, 99
- [44] A. Doucet, N. J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3) :613–24, March 2001. 16, 17, 22, 28, 67, 78, 79, 105, 107, 108, 141, 147
- [45] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM) : Part I the essential algorithms. *IEEE Robotics and automation magazine*, 2 :2006, 2006. 19
- [46] O. Erdinc, P. Willett, and Y. Bar-Shalom. The bin-occupancy filter and its connection to the PHD filters. *IEEE Transactions on Signal Processing*, 57(11) :4232–4246, November 2009. 136
- [47] P. Fearnhead and P. Clifford. On-Line Inference for Hidden Markov Models via Particle Filters. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 65(4) :887–899, 2003. 19, 21, 141, 146
- [48] P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2) :447–464, 2010. 39
- [49] A. E. Gelfand and A. F. M. Smith. Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410) :398–409, 1990. 35

-
- [50] J.E. Gentle. *Random Number Generation and Monte Carlo Methods*. Springer, 2004. 77
- [51] J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6) :1317–1339, November 1989. 78
- [52] J. Geweke. Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrica*, 57(6) :1317–1339, 1989. 35
- [53] W.R. Gilks and C. Berzuini. Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, pages 127–146, 2001. 47, 52
- [54] I. R. Goodman, Ronald P. Mahler, and Hung T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, Norwell, MA, USA, 1997. 23, 24
- [55] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/ non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2) :107–113, April 1993. 15, 34, 37, 40, 41
- [56] J. Goutsias, R. Mahler, and H. Nguyen. *Random Sets Theory and Applications*. Springer- Verlag New York, 1997. 16, 23, 24
- [57] J. M. Hammersley and D. Handscomb. *Monte Carlo Methods*. Methuen & Co, 1965. 34
- [58] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, pages 97–109, 1970. 47, 49
- [59] Y.C. Ho and R. Lee. A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9 :333–339, 1964. 21
- [60] J. D. Hol, T. B. Schön, and F. Gustafsson. On resampling algorithms for particle filtering. In *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop (NSSPW)*, Cambridge, UK, September 2006. 37
- [61] Y. Huang and P.M. Djurić. A Hybride Importance Function for Particle Filtering. *IEEE Signal Processing Letters*, 11(3) :404–406, 2004. 38
- [62] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, San Diego, 1970. 19, 21, 123
- [63] A. M. Johansen and A. Doucet. A note on the auxiliary particle filter. *Statistics and Probability Letters*, 78(12) :1498–1504, September 2008. 38, 47, 54, 64, 76, 98
- [64] A.M. Johansen, S. Singh, A. Doucet, and B-N. Vo. Convergence of the SMC implementation of the PHD filter. *Methodology and Computing in Applied Probability*, 8(2) :265–291, 2006. 27
- [65] A.M. Johansen, N. Whiteley, and A. Doucet. Exact approximation of Rao-Blackwellised particle filters. In *16th IFAC Symposium on System Identification*, 11-12 July 2012. 79, 97, 101, 106
- [66] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422, March 2004. 21, 37, 87, 99
- [67] R. E. Kalman. A new approach to linear filtering and prediction problems. *J. Basic Eng., Trans. ASME, Series D*, 82(1) :35–45, 1960. 21
- [68] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *J. Basic Eng., Trans. ASME, Series D*, 83(3) :95–108, 1961. 21
- [69] C-J. Kim. Dynamic linear models with Markov switching. *J. of Econometrics*, 60 :1–22, 1994. 21
-

- [70] T. Kirubarajan and Y. Bar-Shalom. Probabilistic data association techniques for target tracking in clutter. In *Proceeding of IEEE*, volume 92 (3), pages 536–556, March 2004. 170
- [71] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1) :1–25, 1996. 36
- [72] A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89(425) :278–88, March 1994. 54, 59
- [73] P. Lanchantin and W. Pieczynski. Unsupervised restoration of hidden non stationary Markov chain using evidential priors. *accepted for publication, IEEE Transactions on Signal Processing*, 2004. 121
- [74] P. L'Ecuyer. Efficiency improvement and variance reduction. In *Winter Simulation Conference 1994*, pages 122–132, 1994. 107
- [75] X.R. Li and V.P. Jilkov. Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on In Aerospace and Electronic Systems*, 41(4) :1255–1321, 2005. 140
- [76] F. Lindsten, T. Schön, and J. Olsson. An explicit variance reduction expression for the Rao-Blackwellized particle filter. In *18th World Congress of the Int. Federation of Automatic Control (IFAC)*, August 2011. 28, 78
- [77] R. S. Lipster and A. N. Shiryaev. *Statistics of Random Processes, Vol. 2 : Applications*, chapter 13 : "Conditionally Gaussian Sequences : Filtering and Related Problems". Springer Verlag, Berlin, 2001. 123
- [78] J. S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6 :113–119, 1996. 59
- [79] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443) :1032–44, September 1998. 21, 36, 37
- [80] R. Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(no.4), October 2003. 17, 23, 25, 136, 168
- [81] R. Mahler. *Statistical Multisource Multitarget Information Fusion*. Artech House, 2007. 16, 24, 25, 132, 170
- [82] R. Mahler. A theory of PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems* , pages 1523 –1543, October 2007. 26, 130, 167, 168
- [83] R. Mahler. On multitarget jump-Markov filters. In *15th International Conference on Information Fusion*, Singapore, 9-12 July 2012. 80, 89
- [84] R. Mahler, B.-T. Vo, and B.-N. Vo. CPHD filtering with unknown clutter rate and detection profile. *IEEE Transactions on Signal Processing*, 59(8) :3497–3513, 2011. 130
- [85] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking : a survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1) :103–123, 1998. 16, 22, 140
- [86] M.R. Morelande and A. Skvortsov. Radiation field estimation using a Gaussian mixture. In *Proceedings of the International Conference on Information Fusion*, Seattle, USA, 2009. 88
- [87] J. Mullane, Ba-Ngu Vo, M. D. Adams, and Ba-Tuong Vo. A random-finite-set approach to Bayesian SLAM. *IEEE Transactions on Robotics*, 27(2) :268–282, April 2011. 19

-
- [88] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1) :32–38, March 1957. 171
- [89] V. Nemesin and S. Derrode. Robust blind pairwise Kalman algorithms using QR decompositions. *IEEE Transactions on Signal Processing*, 61, 2013. 156
- [90] J. Olsson and T. Ryden. Rao-blackwellization of particle Markov chain Monte Carlo methods using forward filtering backward sampling. *IEEE Transactions on Signal Processing*, 59(10) :4606–4619, October 2011. 39
- [91] K. Panta, B.-N. Vo, and D. Clark. An efficient track management scheme for the Gaussian Mixture Probability Hypothesis Density (GM-PHD) tracker. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1003–1016, July 2009. 169, 171
- [92] A. Pasha, B.-N. Vo, H.D. Tuan, and W-K. Ma. A Gaussian Mixture PHD Filter for Jump Markov System Models. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3) :919–936, 2009. 80, 89, 130
- [93] W. Pieczynski. Chaînes de Markov triplet. *Comptes Rendus de l’Académie des Sciences - Mathématiques*, 335 :275–278, 2002. in French. 17, 128
- [94] W. Pieczynski. Pairwise Markov chains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5) :634–39, May 2003. 17, 121
- [95] W. Pieczynski. Exact filtering in conditionally markov switching hidden linear models. *Comptes Rendus Mathématique*, 349(9-10) :587–590, 2011. 18, 128, 129, 165
- [96] W. Pieczynski and F. Desbouvries. Kalman filtering using pairwise Gaussian models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 03)*, pages 57–60, Hong-Kong, 2003. 123
- [97] M. K. Pitt and N. Shephard. Filtering via simulation : Auxiliary particle filter. *Journal of the American Statistical Association*, 94(446) :590–99, June 1999. 15, 19, 21, 34, 38, 41, 42, 49, 98, 103
- [98] C.R. Rao. *Linear statistical inference and its applications*. Wiley series in probability and mathematical statistics. Wiley, New York, second edition, 1973. 157
- [99] B. D. Ripley. *Stochastic simulation*. Wiley, 1987. 40, 55
- [100] B. Ristic, D. Clark, and B.-N. Vo. Improved SMC implementation of the PHD filter. In *Proceeding of the 13th International Conference on Information Fusion*, 2010. 113, 114, 115
- [101] B. Ristic, D. Clark, and B-N. Vo. Adaptive target birth intensity for PHD and CPHD filters. *IEEE Transactions on Aerospace and Electronic Systems*, 48 :1656–1668, April 2012. 171
- [102] B. Ristic, B-N. Vo, D. Clark, and B-T. Vo. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Transactions on Signal Processing*, 59 :3452–3457, July 2011. 171
- [103] C. P. Robert. *The Bayesian choice : a decision-theoretic motivation*. Springer-Verlag, 1994. 20
- [104] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. 35
- [105] S.J. Robert and W.D. Penny. Variational Bayes for generalized autoregressive models. Technical report, Oxford University, 2002. 146
- [106] J.A. Roecker and G.L. Phillis. Suboptimal joint probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, 29 :510–517, April 1993. 16, 23
-

- [107] D. B. Rubin. Using the SIR algorithm to simulate posterior distributions. In M. H. Bernardo, K. M. Degroot, D. V. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics III*. Oxford University Press, Oxford, 1988. [35](#), [40](#)
- [108] J. S. and R. Chen. Blind deconvolution via sequential imputation. *Journal of the American Statistical Association*, 90(430) :567–76, June 1995. [54](#), [59](#), [61](#), [64](#)
- [109] S. Saha and F. Gustafsson. Particle filtering with dependent noise processes. *IEEE Transactions on Signal Processing*, 60(9) :4497–4508, September 2012. [123](#), [124](#), [130](#)
- [110] S. Saha, P. K. Manda, Y. Boers, H. Driessen, and A. Bagchi. Gaussian proposal density using moment matching in SMC methods. *Statistics and Computing*, 19(2) :203–208, 2009. [37](#), [49](#), [67](#), [99](#)
- [111] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 2009. [16](#), [23](#), [171](#)
- [112] T. Schön, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear nonlinear state-space models. *IEEE Trans. on Signal Processing*, 53 :2279–2289, 2005. [78](#)
- [113] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8) :3447–3457, August 2008. [26](#)
- [114] A. F. M. Smith and A. E. Gelfand. Bayesian statistics without tears : a sampling-resampling perspective. *The American Statistician*, 46(2) :84–87, 1992. [35](#)
- [115] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7(4) :465–479, 1971. [136](#), [140](#)
- [116] S. Särkkä, A. Vehtari, and J. Lampinen. Rao-blackwellized particle filter for multiple target tracking. *Information Fusion Journal*, 8, 2007. [79](#)
- [117] C. Tianshi, T.B. Schon, H. Ohlsson, and L. Ljung. Decentralized Particle Filter With Arbitrary State Decomposition. *IEEE Transactions on Signal Processing*, 59(2) :465–478, February 2011. [79](#), [97](#), [101](#)
- [118] J. K. Tugnait. Adaptive estimation and identification for discrete systems with Markov jump parameters. *IEEE Transactions on Automatic Control*, 27(5) :1054–65, October 1982. [16](#), [21](#), [22](#), [140](#)
- [119] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. *Advances in Neural Information Processing Systems*, 19(2) :567–76, June 2000. [37](#), [52](#), [67](#)
- [120] M. Vihola. Rao-blackwellised particle filtering in random set multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 2006. [80](#)
- [121] B.-N. Vo and W.K Ma. The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE transactions Signal Processing*, 54 :4091 –4104, November 2006. [26](#), [81](#), [85](#), [114](#), [130](#), [135](#), [136](#)
- [122] B.-N. Vo, S. Singh, and A. Doucet. Sequential monte carlo methods for multi-target filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41, 2005. [25](#), [26](#), [81](#), [86](#), [91](#), [113](#), [114](#), [130](#), [132](#)
- [123] B.-T. Vo, B.-N. Vo, and A. Cantoni. The Cardinalized Probability Hypothesis Density filter for linear gaussian multi- target models. *IEEE Transactions on Signal Processing*, pages 3553–3367, July 2007. [130](#), [168](#)

- [124] N. Whiteley and A. M. Johansen. Recent developments in auxiliary particle filtering. In Barber, Cemgil, and Chiappa, editors, *Inference and Learning in Dynamic Models*. Cambridge University Press, 2010. 38, 39, 47
- [125] N. Whiteley, S. Singh, and S. Godsill. Auxiliary particle implementation of the probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, to appear. 81, 82, 87, 92, 130
- [126] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4 :65–85, 1993. 36
- [127] V. Zaritskii, V. Svetnik, and L. Shimelevich. Monte Carlo technique in problems of optimal data processing. *Automation and remote control*, pages 95–103, 1975. 21, 37
- [128] H. Zhang, Z. Jing, and S. Hu. Gaussian mixture CPHD filter with gating technique. *Signal Processing*, 89(8) :1521–1530, 2009. 169