Filtrage particulaire par bloc parallèle

Rui MIN[1,2,3], Christelle GARNIER[2,3], François SEPTIER[4], John KLEIN[1]

[1]Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille

[2]IMT Nord Europe, Institut Mines-Télécom, Centre for Digital Systems, F-59000 Lille

[3]Univ. Lille, CNRS, Centrale Lille, Institut Mines-Télécom, UMR 9189 - CRIStAL, F-59000 Lille

[4]Université Bretagne Sud, LMBA, UMR CNRS 6205, F-56000 Vannes

rui.min@univ-lille.fr, christelle.garnier@imt-nord-europe.fr
francois.septier@univ-ubs.fr, john.klein@univ-lille.fr

Résumé – Le filtrage particulaire est une méthode efficace pour estimer la distribution a posteriori dans les modèles d'espace d'état non linéaires ou/et non gaussiens. Pour surmonter la malédiction de la dimension du PF, le filtre particulaire par bloc partitionne l'espace d'état et effectue les étapes de correction et de rééchantillonnage indépendamment sur chaque sous-espace. Ce partitionnement en blocs peut réduire significativement la variance de l'estimée de la densité de filtrage, mais il détruit la corrélation entre les sous-espaces.

Dans cet article, nous introduisons un schéma de parallélisation dans le BPF. Le principe consiste à distribuer l'ensemble des particules dans plusieurs BPFs en parallèle. Les résultats montrent l'intérêt de la parallélisation en terme de compromis biais-variance et que l'affectation de partitions différentes aux BPFs en parallèle permet d'obtenir de bien meilleures performances que l'utilisation d'une seule et même partition.

Abstract – Particle filtering (PF) is a powerful method to estimate the posterior distribution in nonlinear/ non Gaussian state space models. To overcome the curse of dimensionality of PF, the block PF (BPF) partitions the state space and runs correction and resampling steps separately on each subspace. Using a blocking step can significantly reduce the variance of the filtering distribution estimate, but it breaks correlation across subspaces.

In this paper, we introduce a parallelization scheme in the BPF. The scheme consists in dispatching the set of particles into M parallel BPFs. We show that the usual benefit of parallelization in terms of bias-variance trade-off remains valid and, most importantly, that assigning different partitions to the parallel filters leads to far better performance than naive parallelization using only one partition.

1 Introduction

Le filtre particulaire (PF : particle filter en anglais) [1] est largement utilisé pour estimer la loi de filtrage dans le cas de modèles d'espace d'état non linéaires ou/et non gaussiens. Le PF approche cette loi par un ensemble d'échantillons pondérés appelés particules. Cependant, il devient inefficace dans les espaces de grande dimension. Cette limitation, appelée malédiction de la dimension, vient du fait que la loi a posteriori devient singulière en grande dimension. Seule une faible partie des échantillons tirés selon la loi d'importance ont des poids significatifs par rapport aux observations. Pour éviter la dégénérescence des poids, il est nécessaire d'augmenter le nombre de particules de façon exponentielle avec la dimension [2, 3], ce qui est trop coûteux en puissance de calcul.

Un certain nombre de méthodes ont été proposées pour palier cette limitation, un état de l'art est disponible dans [4, 5]. Une classe de méthodes prometteuse est basée sur le principe de localisation. En partitionnant l'espace d'état en sous-espaces et sous des hypothèses d'indépendance, certaines étapes du filtre particulaire peuvent être réparties sur les sous- espaces de plus petite dimension. De telles méthodes locales ont été développées dans de nombreux articles [6, 7, 8, 9].

En particulier, le filtre particulaire par bloc (BPF : block particle filter en anglais) proposé par Rebeschini et Van Handel [6] approche la loi a posteriori conjointe par le produit des lois marginales sur chaque sous-espace, appelé bloc. La structure de l'algorithme "bootstrap" est conservée. Après l'étape de prédiction usuelle, une étape de partitionnement en blocs est insérée, puis les étapes de correction et de rééchantillonnage sont effectuées séparément sur chaque bloc. Un partage en blocs de petites tailles réduit la variance de l'estimée de la distribution de filtrage, mais en contrepartie la corrélation entre les blocs est négligée et un biais est introduit. Avec une partition en blocs appropriée, pour laquelle la réduction de variance l'emporte sur l'augmentation du biais, le BPF est beaucoup plus performant que le PF standard dans les problèmes de grande dimension.

Dans ce papier, nous cherchons à améliorer les performances du BPF en atténuant les pertes de corrélation dues à l'étape de division en blocs. Pour cela, nous proposons d'introduire un schéma de parallélisation. Le BPF parallèle (PBPF : parallel BPF) proposé consiste à exécuter en parallèle plusieurs BPFs indépendants et de coût plus faible, chacun utilisant une partition spécifique, puis à combiner les estimées fournies par

ces filtres. Pour un même nombre de particules, notre PBPF fournit de meilleures performances que le BPF. Cette amélioration s'explique par la parallélisation elle-même [10], mais aussi par l'attribution de différentes partitions de l'espace d'état aux BPFs en parallèle, ce qui permet de compenser les pertes de corrélation induites par chaque partition. De plus, le PBPF a l'avantage de réduire le temps de calcul pour un coût identique.

2 Filtre particulaire par bloc (BPF)

Soit $\mathbf{x}_t \in \mathbb{R}^{d_x}$ et $\mathbf{y}_t \in \mathbb{R}^{d_y}$ respectivement un état caché et un processus d'observation. Soit $p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right)$ la densité de probabilité a posteriori de $\mathbf{x}_t$, également appelée densité de filtrage. Nous considérons un modèle de Markov caché, donc :

$$p\left(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}\right) \propto p\left(\mathbf{x}_0\right) \prod_{n=1}^{t} p\left(\mathbf{x}_n|\mathbf{x}_{n-1}\right) p\left(\mathbf{y}_n|\mathbf{x}_n\right) \qquad (1)$$

Dans la plupart des modèles de grande dimension [11], une forte décroissance de la corrélation peut être observée entre les composantes du vecteur d'état qui sont "éloignées" les unes des autres. Rebeschini et Van Handel [6] exploitent cette dépendance locale dans le modèle et insèrent une étape de division en blocs entre les étapes de prédiction et de correction du filtre "bootstrap". L'algorithme, connu sous le nom de filtre particulaire par bloc (BPF), repose sur une partition de l'espace d'état en un ensemble de K blocs indépendants et disjoints. Le vecteur d'état $\mathbf{x}_t$ peut alors s'écrire comme la concaténation de sous-vecteurs $\mathbf{x}_{t,k} = \{x_t(n) : n \in B_k\}$ correspondant à chaque bloc k :

$$\mathbf{x}_t^T = \left[\mathbf{x}_{t,1}^T \mathbf{x}_{t,2}^T \dots \mathbf{x}_{t,K}^T\right] \qquad (2)$$

Les sous-ensembles d'indices $\{B_k\}_{k=1}^K$ vérifient : $B_k \cap B_{k'} = \emptyset$, $\forall k \neq k'$, et $\cup_{k=1}^K B_k = \{1, \dots, d_x\}$. Une telle partition en K blocs s'écrit $\mathcal{P}_K = \{B_1, B_2, \cdots, B_K\}$. Dans la suite, une composante du $k^{\text{ième}}$ bloc $\mathbf{x}_{t,k}$ sera notée $x_{t,k}(n)$.

Dans le BPF, l'étape de prédiction est effectuée classiquement tandis que les étapes de correction (calcul des poids) et de rééchantillonnage sont effectuées indépendamment sur chaque bloc. A l'aide des poids obtenus localement, le BPF calcule séparément K densités marginales $\hat{\pi}_{t,k}$. À la fin de chaque itération, la loi de filtrage $p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right)$ est approchée par le produit de ces densités marginales sur chacun des blocs :

$$\hat{\pi}_t = \bigotimes_{k=1}^{K} \hat{\pi}_{t,k} \underset{N_p \to \infty}{\longrightarrow} \pi_t \qquad (3)$$

où la distribution cible limite π_t est censée être une approximation biaisée (mais pertinente) de la loi de filtrage. Le biais tend vers 0 quand le nombre de blocs K tend vers 1 [6]. L'algorithme du BPF est résumé par l'Algo. 1.

Avec une partition en blocs appropriée, le BPF fournit une meilleure estimée de la loi de filtrage que le PF standard. D'une part, l'utilisation de blocs de petites tailles (avec peu de composantes) permet de réduire considérablement la variance de l'estimée. D'autre part, l'utilisation de blocs de grandes dimensions permet de limiter le biais. Ce biais est introduit dans l'estimée par l'étape de division en blocs, car pour un nombre de blocs $K > 1$, l'approximation (3) ne converge pas vers la loi de filtrage exacte, même si le nombre de particules tend vers l'infini. La performance du BPF dépend donc de la taille des blocs. En pratique, un compromis entre biais et variance est nécessaire.

Algorithm 1 Filtre particulaire par bloc avec une partition spécifique de l'espace d'état $\mathcal{P}_K$.

Entrées : Partition en K blocs : $\mathcal{P}_K = \{B_1, B_2, \cdots, B_K\}$, nombre de particules N_p, densités $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)$, $p\left(\mathbf{y}_t|\mathbf{x}_t\right)$ et $p\left(\mathbf{x}_0\right)$.

Initialisation : À $t = 0$, tirage des particules $\mathbf{x}_0^{(i)} \sim p\left(\mathbf{x}_0\right)$ et initialisation des poids $w_{0,k}^{(i)} = \frac{1}{N_p}$, $\forall i \in [N_p]$ et $\forall k \in [K]$.

Processus séquentiel : À $t \geq 1$ répéter

1. Étape de prédiction :

Tirage des particules $\mathbf{x}_t^{(i)} \sim p\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}\right), \forall i \in [N_p]$

2. Étape de division en blocs / correction :

for $k = 1 : K$ **do**

Calcul des poids par bloc

$w_{t,k}^{(i)} = \prod_{n \in B_k} \alpha_{t,n}\left(\mathbf{y}_t, x_{t,k}^{(i)}(n)\right), \forall i \in [N_p]$

Normalisation des poids par bloc

end for

3. Étape d'estimation :

Approximation de la loi de filtrage

$\hat{\pi}_t = \bigotimes_{k=1}^K \hat{\pi}_{t,k} = \bigotimes_{k=1}^K \sum_{i=1}^{N_p} w_{t,k}^{(i)} \delta_{\mathbf{x}_{t,k}^{(i)}}(\mathbf{x}_{t,k})$

4. Étape de rééchantillonnage :

for $k = 1 : K$ **do**

Rééchantillonnage $\{\mathbf{x}_{t,k}^{(i)}\}_{i=1}^{N_p} \sim \sum_{i=1}^{N_p} w_{t,k}^{(i)} \delta_{\mathbf{x}_{t,k}^{(i)}}(\mathbf{x}_{t,k})$

end for

Une analyse plus approfondie montre que la performance du BPF dépend aussi de la composition des blocs. Le partitionnement entraîne une perte de corrélation entre les blocs, surtout à leurs frontières, ce qui entraîne des discontinuités dans les erreurs. L'erreur est potentiellement plus importante pour les composantes du vecteur d'état situées aux frontières des blocs. Il n'existe pas de moyen simple permettant de déterminer a priori la taille optimale des blocs ni la partition en blocs optimale. Une solution pour assurer de façon certaine un gain de performance avec le BPF consiste à utiliser plusieurs partitions en parallèle.

3 Filtre particulaire par bloc parallèle (PBPF)

Nous proposons d'exécuter plusieurs BPFs indépendants de coût plus faible, chacun utilisant une partition différente du vecteur d'état en blocs, puis de moyenner les estimations fournies par ces filtres. Cette implémentation parallèle du BPF est appelée filtre particulaire par bloc parallèle (ou PBPF : parallel block particle filter en anglais).

Soit N_p le nombre total de particules. Pour effectuer M BPFs

en parallèle, l'ensemble des N_p particules est réparti en M sous-ensembles de N particules, de sorte que $MN = N_p$:

$$\left\{\mathbf{x}_t^{m,(i)}\right\}_{1\leq i\leq N} \quad \text{for } m = 1, \ldots, M$$

Ensuite, le BPF décrit dans l'Algo. 1 est exécuté de façon indépendante avec chaque sous-ensemble de particules. Le $m^{\text{ième}}$ filtre utilise la partition en K blocs notée $\mathcal{P}_K^m = \{B_1^m, B_2^m, \cdots, B_K^m\}$ et met à jour le $m^{\text{ième}}$ sous-ensemble de particules, qui s'écrit alors comme la concaténation de K sous-vecteurs correspondant aux K blocs :

$$\left\{{\mathbf{x}_t^{m,(i)}}^T = \left[{\mathbf{x}_{t,1}^{m,(i)}}^T {\mathbf{x}_{t,2}^{m,(i)}}^T \ldots {\mathbf{x}_{t,K}^{m,(i)}}^T\right]\right\}_{1\leq i\leq N}$$

À chaque instant t, chaque BPF fournit une estimation de la loi de filtrage : $\hat{\pi}_t^m = \bigotimes_{k=1}^K \sum_{i=1}^N w_{t,k}^{m,(i)} \delta_{\mathbf{x}_{t,k}^{m,(i)}}(\mathbf{x}_{t,k})$.

La méthode la plus simple pour agréger ces estimées consiste à faire la moyenne des M sorties indépendantes des filtres. Finalement le PBPF fournit l'approximation suivante de la loi de filtrage $p(\mathbf{x}_t|\mathbf{y}_{1:t})$:

$$\hat{\pi}_t = \frac{1}{M}\sum_{m=1}^{M} \hat{\pi}_t^m \tag{4}$$

Ce simple schéma de parallélisation a l'avantage de réduire le temps de calcul pour un nombre de particules donné. Des travaux ultérieurs pourraient envisager d'autres façons de combiner l'information en introduisant plus ou moins d'interaction entre les BPFs [10]. Grâce à la parallélisation, on peut s'attendre à une réduction de la variance de l'estimateur. En outre, l'attribution de différentes partitions aux BPFs en parallèle peut aider à compenser les erreurs aux frontières des blocs de chaque partition.

4 Résultats de simulation

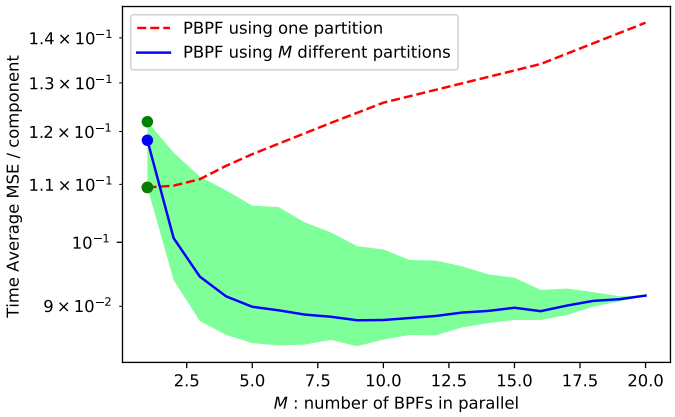
Cette section fournit une validation expérimentale du PBPF dans un espace d'état de grande dimension et montre l'intérêt de la parallélisation. On considère un modèle d'espace d'état linéaire et Gaussien :

$$\begin{aligned}\mathbf{x}_t &= \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_t \\ \mathbf{y}_t &= \mathbf{H}\mathbf{x}_t + \mathbf{v}_t\end{aligned} \tag{5}$$

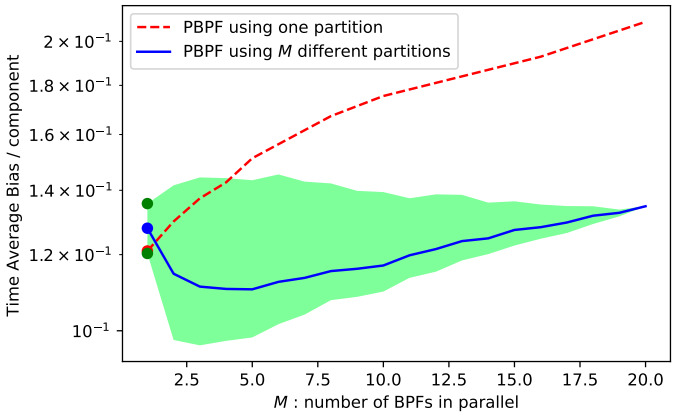
Les dimensions du modèle sont $d_x = d_y = 100$, $\mathbf{V}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{F} = \mathbf{H} = \mathbf{I}$ et $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Pour introduire de la corrélation dans la distribution a posteriori, on considère un bruit d'état corrélé. $\mathbf{W}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ où les éléments de la matrice de covariance s'écrivent $\Sigma_{ij} = \exp(-(i-j)^2/l)$ avec $l = 1000$. Les résultats sont obtenus à partir de 50 simulations indépendantes sur une durée de 100 échantillons temporels.

Nous étudions la stratégie classique qui consiste à partitionner le vecteur d'état en K blocs de même taille ℓ, de sorte que $\ell \times K = d_x$. Chaque bloc contient ℓ composantes consécutives de $\mathbf{x}_t$ (les composantes finales et initiales étant considérées comme telles). Il existe ℓ partitions distinctes de ce type.

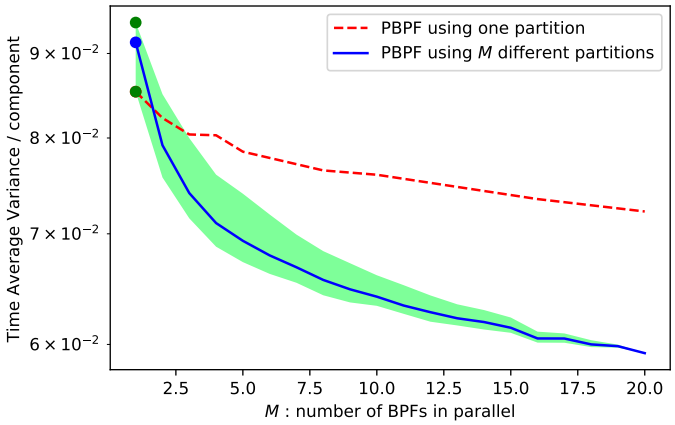
La figure 1 montre les performances moyennes par composante en termes d'erreur quadratique moyenne (ou MSE : mean squared error en anglais), de biais et de variance des estimées du vecteur d'état en fonction du nombre M de BPFs en parallèle pour $\ell = 20$. Le coût de calcul total est réparti entre les M BPFs, de sorte que $N_p = MN = 2000$ particules.



(a) Erreur quadratique moyenne (MSE)



(b) Biais



(c) Variance

FIGURE 1 – Performance du PBPF en fonction du nombre M de BPFs en parallèle pour $N_p = MN = 2000$ et $\ell = 20$.

Pour faire la distinction entre les effets de la parallélisation elle-même et de l'utilisation de différentes partitions en paral-

lèle, la figure représente la performance du PBPF dans deux cas : lorsque la même partition (celle qui permet d'obtenir les meilleures performances) est utilisée dans chaque BPF en parallèle (courbe rouge) et lorsque M partitions distinctes sont utilisées, une pour chaque BPF. Dans ce dernier cas, il faut choisir les M partitions distinctes parmi les ℓ partitions possibles. La performance moyenne obtenue sur toutes les combinaisons possibles est donnée par la courbe bleue. Pour un choix spécifique de M partitions, la performance se situe dans la zone verte délimitée par les performances obtenues avec la meilleure et la moins bonne combinaison. De même pour $M = 1$ (exécution d'un seul BPF utilisant les 2000 particules), les points sur la figure indiquent les performances obtenues avec la meilleure et la moins bonne partition ainsi que la performance moyenne obtenue à partir de toutes les partitions possibles.

La figure montre que le PBPF est plus performant qu'un seul BPF, uniquement si des partitions différentes sont affectées aux BPFs en parallèle. Le gain de performance n'est pas le résultat de la parallélisation mais le résultat de l'utilisation de M partitions distinctes en parallèle.

Pour une majorité de combinaisons de M partitions, le biais diminue pour les faibles valeurs de M (et plus grandes valeurs de N), puis augmente légèrement avec M lorsque M dépasse la valeur de 4 (et $N < 500$). L'utilisation de partitions distinctes avec suffisamment de particules permet de compenser le biais induit par l'étape de division en blocs de chaque BPF, ce qui met en évidence l'intérêt de l'approche parallèle proposée. La variance de l'estimateur fourni par le PBPF diminue lorsque M augmente. Lorsque les BPFs en parallèle utilisent la même partition, la réduction de la variance ne l'emporte pas sur l'augmentation du biais, ce qui conduit à des valeurs de MSE plus élevées.

L'amélioration de la précision de l'estimation peut également être observée pour d'autres valeurs de la taille de bloc ℓ. Le tableau Tab.1 résume les performances en terme de MSE obtenues pour $\ell = 2, 5, 10, 25$ lorsque la moins bonne combinaison de partitions est assignée aux différents BPFs en parallèle. Les résultats montrent que le PBPF utilisant M partitions distinctes ($M \geq 2$) est toujours plus performant qu'un seul BPF avec la meilleure partition, même dans le cas du pire choix pour les M partitions.

FiltreMSETaille de bloc ℓ	2	5	10	25
BPF (M = 1)	0.39	0.21	0.14	0.12
PBPF, M= 2	0.31	0.179	0.130	0.123
PBPF, M= 5		0.149	0.110	0.118
PBPF, M= 10			0.098	0.111

TABLE 1 – MSE en fonction de la taille de bloc ℓ pour $MN = 2000$.

5 Conclusion

Pour améliorer les performances du PF standard en grande dimension, le BPF partitionne l'espace d'état et exécute les étapes de correction et de rééchantillonnage séparément sur chaque sous-espace ou bloc. La variance de l'estimée de la loi de filtrage est ainsi réduite, mais elle s'accompagne d'une perte de corrélation entre les blocs. Afin d'augmenter le gain de performance du BPF, nous proposons une implémentation parallèle qui consiste à exécuter plusieurs BPFs, chacun utilisant une partition différente de l'espace d'état. Les simulations montrent que la stratégie de parallélisation proposée réduit davantage la variance de l'estimée et compense les pertes de corrélation induites par chaque partition. L'amélioration de la précision d'estimation est obtenue pour un coût de calcul identique.

Références

[1] A. Doucet, S. Godsill et C. Andrieu. *On sequential Monte Carlo sampling methods for Bayesian filtering*. Statistics and computing, vol.10, no.3, pp.197-208, 2000.

[2] C. Snyder, Chris, T. Bengtsson, P. Bickel et J. Anderson. *Obstacles to high-dimensional particle filtering*, Monthly Weather Review, vol.136, no.12, pp.4629-4640, 2008.

[3] P. Bickel, B. Li et T. Bengtsson. *Sharp failure rates for the bootstrap particle filter in high dimensions*, Institute of Mathematical Statistics Collections, pp.318–329, 2008.

[4] F. Septier, et G. Peters. *An overview of recent advances in Monte-Carlo methods for Bayesian filtering in high-dimensional spaces*, Theoretical Aspects of Spatial-Temporal Modeling, PP.31–61, 2015

[5] A. Farchi et M. Bocquet, *Comparison of local particle filters and new implementations*, Nonlinear Processes in Geophysics, vol.25, no.4, pp.765–807, 2018

[6] P. Rebeschini et R. Van Handel, *Can local particle filters beat the curse of dimensionality ?*, The Annals of Applied Probability, vol.25, no.5, pp.2809–2866, 2015

[7] P. Djuric, T. Lu et M. Bugallo, *Multiple particle filtering*, IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, vol.3, pp.1181-1184, 2007

[8] P. Closas et M. Bugallo, *Improving accuracy by iterated multiple particle filtering*, IEEE Signal Processing Letters, vol.19, no.8, pp.531–534,2012

[9] R. Min, C. Garnier, F. Septier et J. Klein, *Block Kalman Filter : an Asymptotic Block Particle Filter in the Linear Gaussian Case*, ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.5574–5578, 2021

[10] D. Crisan, J. Míguez et G. Ríos-Muñoz, *On the performance of parallelisation schemes for particle filtering*, EURASIP Journal on Advances in Signal Processing, vol.2018, no.1, pp.31, 2018

[11] H. Georgii, *Gibbs measures and phase transitions* , De Gruyter Studies in Mathematics ; 9. Walter de Gruyter & Co, 2011