

# Filtrage particulaire\*

François LEGLAND<sup>1</sup>

<sup>1</sup> IRISA / INRIA

Campus de Beaulieu, 35042 RENNES Cédex, France

legland@irisa.fr

**Abstract** – Particle filtering is becoming increasingly used in various areas, including localization, navigation, tracking, computer vision, mobile robotics, digital communications, etc. An overview of the currently existing algorithms is presented here, within the framework of hidden Markov models.

## 1 Introduction

Le filtrage particulaire, développé à l'origine [22] dans la communauté « poursuite », connaît actuellement un fort développement dans de nombreux domaines relevant des sciences et techniques de l'information et de la communication, ou des sciences de l'ingénieur :

- localisation, navigation, poursuite [23],
- poursuite multi-cibles [25, 26],
- vision [28], [36, 37],
- robotique [19],
- traitement du signal audio [18, 41],
- communications numériques [4], [8, 42, 43], [40].

Cette liste de références est bien sûr loin d'être exhaustive, en particulier dans le domaine de la poursuite.

Il s'agit d'une méthode de simulation séquentielle, de type Monte Carlo, c'est-à-dire d'une méthode dans laquelle

- des particules explorent l'espace d'état, en évoluant de manière indépendante comme le processus sous-jacent,
- et interagissent sous l'effet d'un mécanisme de sélection, qui concentre automatiquement les particules (i.e. la puissance de calcul disponible) dans les régions d'intérêt de l'espace d'état.

En toute généralité, le processus sous-jacent est une chaîne de Markov, ou un processus de Markov à temps continu, dont l'espace d'état peut être fini, continu (euclidien), hybride (fini / continu), variable dans le temps, trajectorien, etc. la seule condition étant qu'il soit possible de le *simuler* facilement. Dans le cas du filtrage particulaire, le mécanisme de sélection est assuré par la fonction de vraisemblance : à chaque pas de temps, une particule a d'autant plus de chance de se reproduire à la génération suivante qu'elle est cohérente avec l'observation courante. Ces méthodes sont très facile à mettre en œuvre, puisqu'il suffit de savoir simuler de manière indépendante des trajectoires du processus sous-jacent, et se situent à l'interface des algorithmes génétiques et des systèmes de particules en interaction.

Pour les aspects mathématiques, le cours [3] est une très bonne introduction, et le long article [12] présente une synthèse

des très nombreux résultats obtenus par les auteurs. Les applications du filtrage particulaire sont maintenant nombreuses : on en trouvera quelques exemples dans l'ouvrage collectif [15] et dans le numéro spécial [1], où on trouvera aussi un très bon tutoriel [2]. Enfin, on trouvera dans [31] des applications des méthodes de Monte Carlo séquentielles dans d'autres domaines, en dehors du traitement du signal et des images : biologie, dynamique moléculaire, etc.

## 2 Bootstrap filter

De nombreux algorithmes de filtrage particulaire ont été proposés et étudiés dans la littérature depuis une dizaine d'années. On commence par décrire l'algorithme de base, le premier historiquement à contenir une étape de redistribution, introduit sous le nom de filtre particulaire avec interaction [13], *bootstrap filter* [22], *Monte Carlo filter* [29] ou algorithme *condensation* (conditional density propagation) [28].

### 2.1 Modèle d'état non-linéaire non-gaussien

On considère le problème simple suivant, où il s'agit d'estimer l'état  $X_k$  d'un système (par exemple la position et la vitesse d'un mobile), au vu d'observations  $(Y_0, \dots, Y_k)$  (par exemple des observations d'angles et / ou de distance par rapport à une ou plusieurs stations) recueillies dans un bruit blanc additif, pas nécessairement gaussien. En général, les observations seules ne permettent d'estimer l'état caché, et il est nécessaire de disposer d'un modèle *a priori*, suffisamment réaliste, pour décrire l'évolution de l'état entre deux instants d'observation. Par exemple

$$\begin{aligned} X_k &= F_k(X_{k-1}, W_k) \\ Y_k &= h_k(X_k) + V_k \end{aligned} \tag{1}$$

où l'incertitude sur l'état initial est décrite par la distribution de probabilité  $X_0 \sim \mu_0(dx)$ , où  $\{W_k\}$  et  $\{V_k\}$  sont des bruits blancs indépendants, indépendants de la condition initiale  $X_0$ , c'est-à-dire des suites de v.a. indépendantes, pas nécessairement gaussiennes, décrites par les distributions de probabilité

---

\*Ce travail a partiellement bénéficié du soutien du CNRS, dans le cadre du projet *MathSTIC Chaînes de Markov Cachées et Filtrage Particulaire*, et de l'action spécifique *Méthodes Particulières* (AS 67) du département STIC.

$W_k \sim p_k(dw)$  et  $V_k \sim q_k(v) dv$ . Sous ces hypothèses : (i) les états cachés  $\{X_k\}$  suivent une dynamique markovienne (conditionnellement aux états passés  $(X_0, \dots, X_{k-1})$  l'état présent.  $X_k$  ne dépend que de l'état précédent  $X_{k-1}$ ), et (ii) conditionnellement aux états cachés  $(X_0, \dots, X_k)$ , les observations sont indépendantes, et la distribution de l'observation  $Y_k$  ne dépend que de l'état  $X_k$  au même instant :  $Y_k \mid X_k = x \sim q_k(y - h(x)) dy$ , et on désigne par

$$\Psi_k(x) = q_k(Y_k - h_k(x))$$

la fonction de *vraisemblance* associée, qui quantifie l'adéquation de chaque état  $x$  par rapport à l'observation courante  $Y_k$  : cette fonction dépend de l'écart  $Y_k - h_k(x)$  entre l'observation  $h_k(x)$  à laquelle on s'attend si le système occupait l'état  $x$ , et l'observation réelle  $Y_k$ .

## 2.2 Algorithme

L'idée est de représenter la distribution de probabilité conditionnelle  $\mu_k(dx)$  de l'état caché  $X_k$  sachant les observations  $(Y_0, \dots, Y_k)$  à l'aide d'un système de particules caractérisé par les positions  $(\xi_k^1, \dots, \xi_k^N)$  et les poids  $(w_k^1, \dots, w_k^N)$ , soumis au mécanisme d'évolution suivant

**mutation** entre deux observations, les particules explorent l'espace d'état de façon indépendante, en imitant le comportement suivi par l'état caché, c'est-à-dire en suivant le modèle *a priori*,

**pondération** lorsqu'une nouvelle observation est disponible, l'adéquation de chaque particule avec cette observation est évaluée grâce à la fonction de vraisemblance : chaque particule est alors affectée d'un poids proportionnel à la valeur calculée,

**sélection** les particules sont éliminées ou multipliées, en fonction de leur poids, c'est-à-dire que les particules auront d'autant plus de descendants à la génération suivante que leur vraisemblance est grande.

Pour mettre en œuvre l'algorithme décrit dans le Tableau 1, il suffit de savoir

- *simuler* une v.a. selon la distribution initiale, et en tout point  $x$  de l'espace d'état
- *simuler* une v.a. selon la transition  $X_k \mid X_{k-1} = x$ , c'est-à-dire selon la dynamique du modèle *a priori*,
- *calculer* à une constante multiplicative près la fonction de vraisemblance : il faut pour cela que la probabilité d'émission  $Y_k \mid X_k = x$  ait une densité par rapport à une mesure ne dépendant pas de l'état  $x$ , et qu'on dispose d'une expression explicite pour cette densité.

On verra dans la suite de l'article qu'il est possible de modifier cet algorithme sur plusieurs points

- dans l'étape de sélection, il est possible de réduire la variance introduite par l'étape de ré-échantillonnage, en utilisant d'autres algorithmes de redistribution des particules, voir Annexe B,
- il est recommandé de ne pas redistribuer les particules à chaque instant, mais seulement quand un déséquilibre trop important est constaté dans la répartition de leurs poids, voir Annexe C,
- dans l'étape de mutation, il est possible de laisser les particules explorer l'espace d'état selon une autre transition

À l'instant initial  $k = 0$  :

**initialisation** indépendamment pour  $i = 1, \dots, N$  :

- générer  $\xi_0^i \sim \mu_0(dx)$ .

**pondération** pour  $i = 1, \dots, N$  :

- poser  $w_0^i \propto q_0(Y_0 - h_0(\xi_0^i))$ .

**sélection** indépendamment pour  $i = 1, \dots, N$  :

- générer  $\hat{\xi}_0^i \sim \sum_{j=1}^N w_0^j \delta_{\xi_0^j}$ ,

À tout instant  $k \geq 1$  :

**mutation** indépendamment pour  $i = 1, \dots, N$  :

- générer  $W_k^i \sim p_k(dw)$ ,
- poser  $\xi_k^i = F_k(\hat{\xi}_{k-1}^i, W_k^i)$ .

**pondération** pour  $i = 1, \dots, N$  :

- poser  $w_k^i \propto q_k(Y_k - h_k(\xi_k^i))$ .

**sélection** indépendamment pour  $i = 1, \dots, N$  :

- générer  $\hat{\xi}_k^i \sim \sum_{j=1}^N w_k^j \delta_{\xi_k^j}$ ,

TAB. 1 – Bootstrap filter

que celle correspondant à la dynamique du modèle *a priori*, quitte à pondérer (ou rejeter) les v.a. simulées, de façon à compenser cette erreur, voir Section 4.

L'étape de sélection, au cours de laquelle les particules sont éliminées ou multipliées, est cruciale pour l'efficacité de l'algorithme. Si cette étape est omise, tout se passe comme si  $N$  trajectoires indépendantes  $\xi_{0:n}^i = (\xi_0^i, \dots, \xi_n^i)$  pour  $i = 1, \dots, N$ , étaient simulées une fois pour toutes, en *aveugle*, avant même que les observations soient disponibles. Il n'y a aucune raison pour que, parmi ces  $N$  trajectoires données à l'avance, il s'en trouve même une seule qui soit proche de la *vraie* trajectoire. En pratique, on constate d'ailleurs que très rapidement une trajectoire concentre tout le poids, les autres trajectoires ayant un poids quasi-nul et ne contribuant pas à l'estimation de l'état caché (jusqu'à ce qu'éventuellement une autre trajectoire, ayant fini par acquérir un poids cumulé supérieur, concentre à son tour tout le poids). La solution à ce problème de *dégénérescence des poids* consiste précisément à redistribuer les particules, de façon à utiliser au mieux les ressources disponibles.

## 2.3 Points faibles et variantes

On peut remarquer que l'algorithme décrit dans cette section effectue la redistribution de manière *aveugle* : une population de particules est proposée à l'issue de l'étape de mutation, et la seule latitude laissée à l'algorithme est de valider ou d'invalider chacune des particules proposées, au vu de son adéquation à la nouvelle observation. Dans certains cas, par exemple si le bruit d'observation est additif et que le rapport signal-à-bruit est fort, il est peut-être déjà trop tard pour réagir : la fonction de vraisemblance étant très localisée, il peut arriver qu'aucune des particules proposées à l'issue de l'étape de mutation ne convienne. Une première solution consiste à introduire avant l'étape de sélection une étape de régularisation par noyau, e.g. *kernel filter* [27], filtre particulière pré-régularisé [33], qui

offre la possibilité de proposer des particules dans un voisinage des particules déjà proposées. Idéalement, il faudrait pouvoir forcer le mécanisme de mutation à proposer des particules dans des régions de forte vraisemblance, c'est-à-dire effectuer la redistribution de manière *guidée* par la nouvelle observation. Plusieurs méthodes ont été proposées : *prior editing* [22], *auxiliary particle filter* [39], correction progressive [33], *annealed importance sampling* [34], *bridging densities* [21].

À l'issue de l'étape de redistribution, les descendants d'une particule qui a été multipliée occupent toutes la même position. Si la transition  $X_k | X_{k-1} = x$  est faiblement bruitée, voire même déterministe, alors la capacité d'exploration de l'espace d'état est très diminuée. Pour résoudre ce problème de *dégénérescence des positions*, on peut ajouter, à l'issue de l'étape de sélection, une étape de *roughening* [22], qui consiste essentiellement à ajouter un bruit dans le modèle d'état, et qui a donné lieu au filtre particulaire post-régularisé [33], ou quelques itérations d'une méthode MCMC [20].

### 3 Modèles de Markov cachés

Les modèles de Markov cachés possèdent un très grand pouvoir de modélisation, et permettent de décrire très simplement des structures de dépendances temporelles complexes. En pratique, il suffit de spécifier les transitions  $X_k | X_{k-1} = x$  et les densités d'émission  $Y_k | X_k = x$ . Par rapport aux modèles linéaires gaussiens usuels, il est également très facile

- d'inclure dans la modélisation des contraintes sur l'état,
- de considérer des modèles de Markov à état fini, continu, hybride continu / fini, variable dans le temps, trajectoriel, etc.,

et les algorithmes associés, à base de systèmes de particules en interaction, sont très simples et intuitifs à développer et à mettre en œuvre.

#### 3.1 Modèle d'état

On suppose que la suite des états cachés  $\{X_k\}$  forme une chaîne de Markov sur un espace  $E$  (on peut même considérer le cas où cet espace dépend du temps, c'est-à-dire où  $X_k$  prend ses valeurs dans un espace  $E_k$ ), i.e. conditionnellement aux états passés  $(X_0, \dots, X_{k-1})$ , l'état présent  $X_k$  ne dépend que de l'état précédent  $X_{k-1}$ . On en déduit

$$\begin{aligned} & \mathbb{P}[X_0 \in dx_0, \dots, X_n \in dx_n] \\ &= \underbrace{\mathbb{P}[X_0 \in dx_0]}_{\mu_0(dx_0)} \prod_{k=1}^n \underbrace{\mathbb{P}[X_k \in dx_k | X_{k-1} = x_{k-1}]}_{Q_k(x_{k-1}, dx_k)}. \end{aligned}$$

Cette situation est complètement décrite par la donnée de la distribution initiale et de la caractéristique locale

$$\mathbb{P}[X_0 \in dx] = \mu_0(dx),$$

$$\mathbb{P}[X_k \in dx' | X_{k-1} = x] = Q_k(x, dx').$$

En général, il n'existe pas d'expression explicite pour le noyau de transition  $Q_k(x, dx')$ , ou bien cette expression est si compliquée que *calculer* une intégrale telle que

$$\mu Q_k(dx') = \int_E \mu(dx) Q_k(x, dx'),$$

est impossible. Dans la plupart des cas pratiques, y compris dans l'exemple considéré en (1), la distribution de probabilité  $Q_k(x, dx')$  n'admet déjà pas de densité par rapport à la mesure de Lebesgue sur  $E$ .

En revanche, il est souvent très facile de *simuler* une v.a. selon la distribution de probabilité  $Q_k(x, dx')$  (c'est le cas dans l'exemple considéré en (1), pourvu toutefois qu'on sache générer une v.a. selon la distribution de probabilité  $p_k(dw)$ ). Dans toute la suite, on fera donc l'hypothèse que pour tout point de départ  $x \in E$ , il est facile de *simuler* une v.a. selon la distribution de probabilité  $Q_k(x, dx')$ .

#### 3.2 Canal sans mémoire

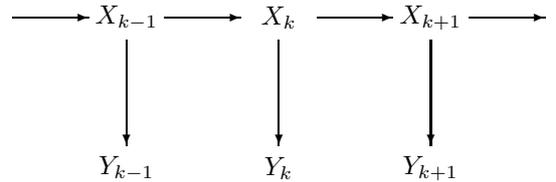
L'hypothèse usuellement faite dans les modèles de Markov cachés pour relier les observations aux états cachés, est de supposer que les observations sont indépendantes, sachant les états cachés, et que la probabilité d'émission de  $Y_k$  ne dépend que de  $X_k$ , i.e.

$$\begin{aligned} & \mathbb{P}[Y_0 \in dy_0, \dots, Y_n \in dy_n | X_0 = x_0, \dots, X_n = x_n] \\ &= \prod_{k=0}^n \mathbb{P}[Y_k \in dy_k | X_0 = x_0, \dots, X_n = x_n] \\ &= \prod_{k=0}^n \underbrace{\mathbb{P}[Y_k \in dy_k | X_k = x_k]}_{g_k(x_k, y_k) \lambda_k(dy_k)}. \end{aligned}$$

La fonction de vraisemblance (adéquation aux données) est alors définie par

$$\Psi_k(x) = g_k(x, Y_k). \quad (2)$$

Cette situation est complètement décrite par le diagramme de dépendance ci-dessous



et par la donnée de la distribution initiale et des caractéristiques locales

$$\mathbb{P}[X_0 \in dx] = \mu_0(dx),$$

$$\mathbb{P}[X_k \in dx' | X_{k-1} = x] = Q_k(x, dx'),$$

$$\mathbb{P}[Y_k \in dy' | X_k = x'] = g_k(x', y') \lambda_k(dy').$$

On peut vérifier que le modèle (1) rentre bien dans ce cadre. Dans tous les modèles où le bruit d'observation est additif, avec une densité connue explicitement, il est en principe facile de calculer la fonction de vraisemblance (2) mais ce calcul peut parfois être coûteux, par exemple dans les applications en vision. Si le bruit d'observation n'est pas additif, une solution possible consiste à considérer un modèle perturbé, incluant un petit bruit d'observation de densité connue [11]. Des modèles beaucoup plus généraux, où les observations et les états cachés forment conjointement une chaîne de Markov [11], voire même des modèles non-markoviens [10], peuvent également être considérés.

### 3.3 Filtre bayésien

L'objectif est l'estimation *récurrente* de l'état caché  $X_k$  au vu des observations  $(Y_0, \dots, Y_k)$ . On introduit à cet effet les distributions de probabilité conditionnelles

$$\mu_{k|k-1}(dx) = \mathbb{P}[X_k \in dx \mid Y_0, \dots, Y_{k-1}],$$

$$\mu_k(dx) = \mathbb{P}[X_k \in dx \mid Y_0, \dots, Y_k].$$

Dans un contexte linéaire gaussien, ces lois conditionnelles sont des lois gaussiennes, respectivement

$$\mathcal{N}(\widehat{X}_{k|k-1}, P_{k|k-1}) \quad \text{et} \quad \mathcal{N}(\widehat{X}_k, P_k),$$

et le filtre de Kalman donne des équations pour le calcul récursif des moyennes conditionnelles  $\widehat{X}_{k|k-1}$  et  $\widehat{X}_k$ , et des matrices de covariance d'erreur  $P_{k|k-1}$  et  $P_k$ . Dans le cas général considéré ici, la transition de  $\mu_{k-1}$  à  $\mu_k$  est décrite par les deux étapes suivantes

**Étape de prédiction** Entre l'instant  $(k-1)$  et l'instant  $k$  : on utilise

- l'information disponible  $\mu_{k-1}$  sur l'état caché  $X_{k-1}$ ,
- et le modèle *a priori*, i.e. le noyau de transition  $Q_k(x, dx')$ , pour prédire l'état caché  $X_k$ , d'où

$$\mu_{k|k-1}(dx') = \int_E \mu_{k-1}(dx) Q_k(x, dx') = \mu_{k-1} Q_k(dx').$$

**Étape de correction** À l'instant  $k$ , une nouvelle observation  $Y_k$  est disponible : on utilise

- l'information *a priori*  $\mu_{k|k-1}$  sur l'état caché  $X_k$ ,
  - et la fonction de vraisemblance  $\Psi_k(x')$ ,
- pour corriger l'information sur l'état caché  $X_k$ , à l'aide de la formule de Bayes, d'où

$$\mu_k = \Psi_k \cdot \mu_{k|k-1} \quad \text{i.e.} \quad \mu_k(dx') = \frac{\Psi_k(x') \mu_{k|k-1}(dx')}{\langle \mu_{k|k-1}, \Psi_k \rangle}.$$

On a donc obtenu l'équation récurrente suivante

$$\mu_k = \Phi_k(\mu_{k-1}) \quad \text{où} \quad \Phi_k(\mu)(dx') = \frac{\Psi_k(x') \mu Q_k(dx')}{\langle \mu Q_k, \Psi_k \rangle}.$$

Au lieu du noyau de transition  $Q_k(x, dx')$  correspondant au modèle *a priori*, on pourra utiliser un noyau  $P_k(x, dx')$ , qui peut dépendre de l'observation  $Y_k$ , voire même de toutes les observations passées  $(Y_0, \dots, Y_k)$ , et qui domine le noyau positif (non-normalisé)  $\Psi_k(x') Q_k(x, dx')$ , avec la dérivée de Radon-Nikodym  $W_k(x, x')$ , i.e.

$$Q_k(x, dx') \Psi_k(x') = W_k(x, x') P_k(x, dx'). \quad (3)$$

## 4 Méthodes de Monte Carlo séquentielles

À chaque instant  $k$ , le filtre optimal  $\mu_k$  est approché par la distribution empirique pondérée  $\mu_k^N$

$$\mu_k \approx \mu_k^N = \sum_{i=1}^N w_k^i \delta_{\xi_k^i} \quad \text{avec} \quad \sum_{i=1}^N w_k^i = 1,$$

d'une population de particules caractérisée par les positions  $(\xi_k^1, \dots, \xi_k^N)$  et les poids  $(w_k^1, \dots, w_k^N)$ . Si on applique successivement l'étape de prédiction puis l'étape de correction à l'approximation  $\mu_{k-1}^N$ , on obtient

$$\begin{aligned} \Phi_k(\mu_{k-1}^N)(dx') &= \frac{1}{c_k} \Psi_k(x') \sum_{i=1}^N w_{k-1}^i Q_k(\xi_{k-1}^i, dx') \\ &= \frac{1}{c_k} \sum_{i=1}^N W_k(\xi_{k-1}^i, x') w_{k-1}^i P_k(\xi_{k-1}^i, dx'), \end{aligned}$$

avec la constante de normalisation

$$c_k = \sum_{i=1}^N w_{k-1}^i c_k^i \quad \text{où} \quad c_k^i = \int_E Q_k(\xi_{k-1}^i, dx') \Psi_k(x'),$$

pour  $i = 1, \dots, N$ , d'où (en interprétant  $\Phi_k(\mu_{k-1}^N)$  comme la marginale d'une loi définie sur l'espace produit  $E \times E$ ) l'approximation par échantillonnage préférentiel

$$\Phi_k(\mu_{k-1}^N) \approx \mu_k^N = \sum_{i=1}^N w_k^i \delta_{\xi_k^i},$$

où  $(\xi_k^1, \dots, \xi_k^N)$  est un  $N$ -échantillon distribué selon la loi d'importance  $\mu_{k-1}^N P_k$ , i.e. indépendamment pour  $i = 1, \dots, N$

$$\tau_{k-1}^i \sim (w_{k-1}^1, \dots, w_{k-1}^N) \quad \text{et} \quad \xi_k^i \sim P_k(\xi_{k-1}^i, dx'),$$

et

$$w_k^i = W_k(\xi_{k-1}^i, \xi_k^i) / \left[ \sum_{j=1}^N W_k(\xi_{k-1}^j, \xi_k^j) \right].$$

**Choix du noyau d'importance** Si  $P_k(x, dx') = Q_k(x, dx')$ , alors  $W_k(x, x') = \Psi_k(x')$ , et on retrouve ainsi, pour un modèle plus général, les caractéristiques du *bootstrap filter* présenté à la Section 2, où les particules explorent d'abord l'espace d'état en aveugle, selon le noyau de transition du modèle *a priori*, avant d'être pondérées (puis sélectionnées) grâce à la fonction de vraisemblance. Le choix optimal [16] est

$$P_k(x, dx') = \widehat{Q}_k(x, dx') = \frac{Q_k(x, dx') \Psi_k(x')}{\widehat{\Psi}_k(x)},$$

auquel cas

$$W_k(x, x') = \widehat{\Psi}_k(x) = \int_E Q_k(x, dx') \Psi_k(x'),$$

ne dépend pas de  $x'$ , et peut donc être utilisé pour la sélection, qui sera ainsi guidée par l'observation  $Y_k$ . En effet, on peut réécrire

$$\Phi_k(\mu_{k-1}^N)(dx') = \frac{\bar{c}_k}{c_k} \sum_{i=1}^N w_{k-1}^i \underbrace{\frac{\widehat{\Psi}_k(\xi_{k-1}^i)}{\bar{c}_k}}_{\pi_k^i} \widehat{Q}_k(\xi_{k-1}^i, dx'),$$

d'où (en interprétant  $\Phi_k(\mu_{k-1}^N)$  comme la marginale d'une loi définie sur l'espace produit  $\{1, \dots, N\} \times E$ ) l'approximation par échantillonnage préférentiel

$$\Phi_k(\mu_{k-1}^N) \approx \mu_k^N = \sum_{i=1}^N w_k^i \delta_{\xi_k^i},$$

où indépendamment pour  $i = 1, \dots, N$

$$\tau_k^i \sim (\pi_k^1, \dots, \pi_k^N) \quad \text{et} \quad \xi_k^i \sim \widehat{Q}_k(\xi_{k-1}^i, dx'),$$

et

$$w_k^i = w_{k-1}^i / \left[ \sum_{j=1}^N w_{k-1}^j \right].$$

Malheureusement, sauf dans quelques cas particuliers [16], on ne sait ni *calculer* la fonction de vraisemblance prédite  $\widehat{\Psi}_k(x')$ , ni *simuler* selon le noyau de transition *a posteriori*  $\widehat{Q}_k(x, dx')$ .

**Particules auxiliaires [39]** En reprenant la même idée, on peut réécrire

$$\Phi_k(\mu_{k-1}^N)(dx') = \frac{\bar{\Psi}_k}{c_k} \sum_{i=1}^N w_{k-1}^i \frac{\Psi_k(x')}{\Psi_k^i} \underbrace{\frac{\Psi_k^i}{\Psi_k}}_{\pi_k^i} Q_k(\xi_{k-1}^i, dx'),$$

où  $\Psi_k^i$  est aussi proche que possible de  $c_k^i = \widehat{\Psi}_k(\xi_{k-1}^i)$ , par exemple  $\Psi_k^i = \Psi_k(\mu_k^i)$ , où  $\mu_k^i$  est caractéristique de la distribution de probabilité  $Q_k(\xi_{k-1}^i, dx')$ , d'où (en interprétant  $\Phi_k(\mu_{k-1}^N)$  comme la marginale d'une loi définie sur l'espace produit  $\{1, \dots, N\} \times E$ ) l'approximation par échantillonnage préférentiel

$$\Phi_k(\mu_{k-1}^N) \approx \mu_k^N = \sum_{i=1}^N w_k^i \delta_{\xi_k^i},$$

où indépendamment pour  $i = 1, \dots, N$

$$\tau_k^i \sim (\pi_k^1, \dots, \pi_k^N) \quad \text{et} \quad \xi_k^i \sim Q_k(\xi_{k-1}^i, dx'),$$

et

$$w_k^i = w_{k-1}^i \frac{\Psi_k(\xi_k^i)}{\Psi_k^i} / \left[ \sum_{j=1}^N w_{k-1}^j \frac{\Psi_k(\xi_k^j)}{\Psi_k^j} \right].$$

**Algorithme générique** Un algorithme de Monte Carlo séquentiel assez général est décrit dans le Tableau 2, où

- les particules n'explorent pas l'espace d'état selon le noyau de transition  $Q_k(x, dx')$  associé au modèle *a priori*, mais selon un autre noyau de transition  $P_k(x, dx')$ , voir (3),
- les particules ne sont redistribuées que si les poids affectés aux particules sont trop déséquilibrés, selon un des critères décrits dans l'Annexe C,
- le cas échéant, les particules ne sont pas ré-échantillonnées, mais sont redistribuées selon un des mécanismes décrits dans l'Annexe B.

## 5 Rao-Blackwellisation

Bien que les méthodes de Monte Carlo soient en principe applicables en dimension quelconque, il est évidemment important dans chaque application de tirer parti de toutes les particularités du problème, de façon à réduire la partie de l'espace d'état où on aura vraiment besoin d'utiliser des particules : s'il existe des méthodes déterministes qui permettent de traiter certaines composantes du vecteur d'état, il est bien sûr recommandé de les utiliser. Dans chacun des deux exemples ci-dessous, l'approximation particulaire prend la forme

À l'instant initial  $k = 0$  :

**initialisation** indépendamment pour  $i = 1, \dots, N$  :

- générer  $\xi_0^i \sim \mu_0(dx)$ .

**pondération** pour  $i = 1, \dots, N$  :

- poser  $w_0^i \propto \Psi_0(\xi_0^i)$ .

À tout instant  $k \geq 1$  :

**mutation** indépendamment pour  $i = 1, \dots, N$  :

- générer  $\xi_k^i \sim P_k(\xi_{k-1}^i, dx')$ ,

**pondération** pour  $i = 1, \dots, N$  :

- poser  $w_k^i \propto w_{k-1}^i W_k(\xi_{k-1}^i, \xi_k^i)$ , où  $W_k(x, x')$  est défini en (3).

**sélection** si  $H(w_k^1, \dots, w_k^N) \geq H_{\text{red}}$ , alors redistribuer les particules de façon à obtenir un nouveau système de particules  $(\xi_k^1, \dots, \xi_k^N)$ , avec des poids égaux à  $1/N$ .

TAB. 2 – Algorithme de Monte Carlo séquentiel

- d'un système de particules à valeurs dans la partie « non-linéaire » de l'espace d'état,
- à chaque particule est attaché un filtre de Kalman, dont dépendent
  - une distribution gaussienne sur la partie « linéaire » de l'espace d'état,
  - le poids de la particule.

Ce principe est très largement utilisé dans les applications [6], [17], [23], [8].

**Système conditionnellement linéaire gaussien [6], [7]** L'état  $X_n = (X_n^1, X_n^2)$  se décompose en deux parties :  $\{X_n^1\}$  individuellement est une chaîne de Markov, et conditionnellement à  $\{X_n^1\}$ , le système  $\{X_n^2, Y_n\}$  est linéaire gaussien, i.e.

$$X_n^2 = A(X_n^1) X_{n-1}^2 + B(X_n^1) W_n,$$

$$Y_n = C(X_n^1) X_n^2 + V_n.$$

Les deux lois conditionnelles  $X_n^2 | X_0^1, \dots, X_n^1, Y_0, \dots, Y_n$  et  $Y_0, \dots, Y_n | X_0^1, \dots, X_n^1$  sont gaussiennes.

**Système linéaire gaussien partiellement observé [14]** L'état  $X_n = (X_n^1, X_n^2)$  vérifie un système linéaire gaussien triangulaire, et seule la composante  $X_n^2$  est observée, dans un bruit blanc additif :

$$X_{n+1}^1 = A_n X_n^1 + B_n W_n,$$

$$X_n^2 = C_n X_n^1 + D_n X_{n-1}^2 + E_n W_n,$$

$$Y_n = h(X_n^2) + V_n.$$

Dans ce modèle, les deux lois conditionnelles  $X_n^1 | X_0^2, \dots, X_n^2$  et  $X_n^2 | X_0^2, \dots, X_{n-1}^2$  sont gaussiennes.

## A Échantillonnage préférentiel

Il s'agit d'une méthode classique permettant d'approcher une distribution de probabilité  $\mu$ , appelée distribution *cible*, en simulant des v.a. selon une autre distribution de probabilité  $\pi$ , appelée distribution *instrumentale* (ou distribution d'*importance*

dans le cas de l'échantillonnage préférentiel). L'hypothèse de base est que  $\pi$  domine  $\mu$ , i.e.

$$\mu(dx) = \frac{1}{c} w(x) \pi(dx),$$

où il est facile de générer une v.a. de loi  $\pi$ , et où la constante de normalisation

$$c = \int_E w(x) \pi(dx)$$

n'est pas nécessairement connue. Par définition

$$\int_E f(x) \mu(dx) = \frac{\int_E f(x) w(x) \pi(dx)}{\int_E w(x) \pi(dx)} \approx \frac{\sum_{i=1}^N f(X^i) w(X^i)}{\sum_{j=1}^N w(X^j)}$$

où  $(X^1, \dots, X^N)$  est un  $N$ -échantillon distribué selon la loi d'importance  $\pi$ , et on obtient l'approximation (biaisée, mais asymptotiquement sans biais quand le nombre  $N$  de particules croît)

$$\mu \approx \mu^N = \sum_{i=1}^N W^i \delta_{X^i} \quad \text{où} \quad W^i = w(X^i) / \left[ \sum_{j=1}^N w(X^j) \right].$$

On notera que les v.a.  $(X^1, \dots, X^N)$  ne sont pas distribuées selon la loi cible  $\mu$ , et que les poids  $(W^1, \dots, W^N)$  ont précisément pour fonction de compenser cette erreur.

## B Redistribution

Dans l'étape de sélection, il faut générer une v.a.  $\Xi$  (en fait, un échantillon de taille  $N$ ), selon une distribution de probabilité discrète de la forme

$$\mu = \sum_{i=1}^N W^i \delta_{X^i},$$

Il suffit de générer d'abord une v.a.  $\tau$  à valeurs dans l'ensemble fini  $\{1, \dots, N\}$ , selon les poids  $(W^1, \dots, W^N)$ , puis de poser  $\Xi = X^\tau$ . La méthode la plus directe est la méthode d'inversion, qui consiste à générer une v.a.  $U$  uniforme sur  $[0, 1]$ . Si  $U$  appartient au  $j$ -ème segment, i.e. si

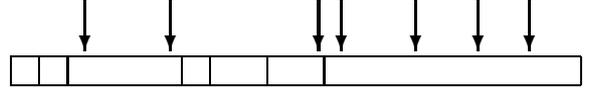
$$W^1 + \dots + W^{j-1} \leq U < W^1 + \dots + W^j,$$

alors on pose  $\tau = j$ . Une recherche binaire en  $O(\log N)$  opérations permet d'obtenir ce résultat, et il suffit donc de  $O(N \log N)$  opérations pour produire le  $N$ -échantillon  $(\Xi^1, \dots, \Xi^N)$ .

**Statistique d'ordre uniforme** On divise l'intervalle  $[0, 1]$  en  $N$  segments contigus de longueurs  $W^1, \dots, W^N$  respectivement, et on génère en une fois  $N$  variables aléatoires  $U_1 < \dots < U_N$  ordonnées, uniformément distribuées sur  $[0, 1]$ . Si  $U_i$  appartient au  $j$ -ème segment, i.e. si

$$W^1 + \dots + W^{j-1} \leq U_i < W^1 + \dots + W^j,$$

alors on pose  $\tau^i = j$ . Générer une statistique d'ordre uniforme peut se faire en  $O(N)$  opérations [5], et il suffit de  $O(N)$  tests de comparaison pour produire le  $N$ -échantillon  $(\Xi^1, \dots, \Xi^N)$ .



Approcher la loi discrète pondérée

$$\mu = \sum_{i=1}^N W^i \delta_{X^i} \approx \frac{1}{N} \sum_{i=1}^N \delta_{\Xi^i}$$

où  $\Xi^i \sim \mu$  indépendamment pour  $i = 1, \dots, N$ , revient à faire  $K^i$  copies de  $X^i$  pour  $i = 1, \dots, N$ , i.e.

$$\mu = \sum_{i=1}^N W^i \delta_{X^i} \approx \sum_{i=1}^N \frac{K^i}{N} \delta_{X^i}$$

où

$$(K^1, \dots, K^N) \sim \mathcal{M}(N, W^1, \dots, W^N)$$

suit une loi multinomiale. Dans ce cas

$$\mathbb{E}(K^i) = N W^i \quad \text{et} \quad \text{var}(K^i) = N W^i (1 - W^i).$$

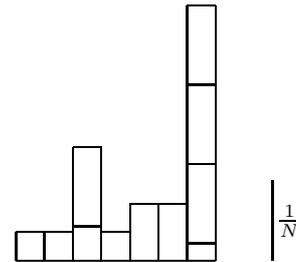
Plus généralement, pour choisir une approximation entière  $K^i$  de  $N W^i$  pour  $i = 1, \dots, N$  tout en rajoutant le moins d'aléa possible, l'idée naturelle est de commencer par créer  $\bar{K}^i = \lfloor N W^i \rfloor$  copies de la particule  $X^i$ . On sélectionne ainsi

$$\bar{N} = \sum_{i=1}^N \bar{K}^i = \sum_{i=1}^N \lfloor N W^i \rfloor$$

particules de manière complètement déterministe, chacune avec un poids égal à  $1/N$ , et il reste

$$R = N - \bar{N} = \sum_{i=1}^N \underbrace{(N W^i - \lfloor N W^i \rfloor)}_{W_R^i}$$

particules à choisir. On peut citer par exemple



– la méthode d'échantillonnage résiduel [32], qui consiste à générer un  $R$ -échantillon  $(\Xi^1, \dots, \Xi^R)$  selon la distribution résiduelle

$$\mu_R = \frac{1}{R} \sum_{i=1}^N W_R^i \delta_{X^i},$$

– ou la méthode de branchement de Bernoulli [9] (dans laquelle le nombre de particules n'est pas nécessairement constant), qui consiste à attribuer 0 ou 1 descendant supplémentaire à la particule  $X^i$ , avec probabilité  $W_R^i$ , indépendamment pour  $i = 1, \dots, N$  : dans ce cas

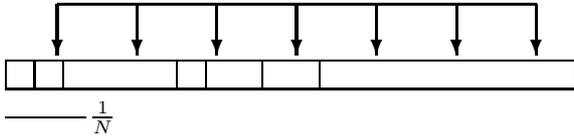
$$\mathbb{E}(K^i) = N W^i \quad \text{et} \quad \text{var}(K^i) = W_R^i (1 - W_R^i) < \frac{1}{4}.$$

La méthode suivante donne aussi une variance minimale.

**Échantillonnage systématique [29], [5]** On génère une *unique* v.a.  $U$  uniforme sur  $[0, 1/N]$ , et on pose  $U_1 = U$  et  $U_i = U_{i-1} + 1/N$  pour  $i = 2, \dots, N$ . Si  $U_i$  appartient au  $j$ -ème segment, i.e. si

$$W^1 + \dots + W^{j-1} \leq U_i < W^1 + \dots + W^j,$$

alors on pose  $\Xi^i = X^j$ . Il suffit de  $O(N)$  tests de comparaison pour produire le  $N$ -échantillon  $(\Xi^1, \dots, \Xi^N)$ .



Dans ce cas

$$\mathbb{E}(K^i) = N W^i \quad \text{et} \quad \text{var}(K^i) = W_R^i (1 - W_R^i) < \frac{1}{4}.$$

## C Critère de redistribution

Étant donné la probabilité discrète

$$\mu = \sum_{i=1}^N W^i \delta_{X^i},$$

il n'est véritablement intéressant de redistribuer les particules  $(X^1, \dots, X^N)$  que si les poids  $(W^1, \dots, W^N)$  sont très dés-équilibrés. Deux mesures heuristiques ont été proposés pour quantifier l'écart à l'équidistribution, et pour prendre la décision de redistribuer les particules.

**Taille effective de l'échantillon [30]** L'inégalité

$$N_{\text{eff}} = 1 / \left[ \sum_{i=1}^N |W^i|^2 \right] \leq N,$$

où l'égalité est atteinte à l'équidistribution, suggère de redistribuer si

$$H(W^1, \dots, W^N) = \frac{1}{N} \sum_{i=1}^N |N W^i|^2 \geq H_{\text{red}} > 1.$$

**Entropie de l'échantillon [38]** L'inégalité

$$-\sum_{i=1}^N W^i \log W^i \leq \log N,$$

où l'égalité est atteinte à l'équidistribution, suggère de redistribuer si

$$H(W^1, \dots, W^N) = \frac{1}{N} \sum_{i=1}^N (N W^i) \log(N W^i) \geq H_{\text{red}} > 0.$$

## Références

- [1] Special issue on Monte Carlo Methods for Statistical Signal Processing (P. M. Djurić and S. J. Godsill, guest editors). *IEEE Transactions on Signal Processing*, SP-50(2), Feb. 2002.
- [2] M. S. Arulampalam, S. Maksell, N. J. Gordon, and T. Clapp. A tutorial on particle filters for online non-linear / non-Gaussian Bayesian tracking. In [1], pages 174–188.
- [3] N. Bartoli and P. Del Moral. *Simulation et Algorithmes Stochastiques*. Cépaduès, Toulouse, 2001.
- [4] F. Ben Salem. *Réception particulière pour canaux multi-trajets évanescents en communications radiomobiles*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, Nov. 2002.
- [5] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for non-linear problems. *IEE Proceedings — Radar, Sonar and Navigation*, 146(1) :2–7, Feb. 1999.
- [6] H. Carvalho. *Filtrage optimal non-linéaire du signal GPS NAVSTAR en recalage de centrales de navigation*. Thèse de Doctorat, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, Sept. 1995.
- [7] R. Chen and J. S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society, Series B*, 62(3) :493–508, 2000.
- [8] R. Chen, X. Wang, and J. S. Liu. Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filters. *IEEE Transactions on Information Theory*, IT-46(6) :2079–2094, Sept. 2000.
- [9] D. Crişan, P. Del Moral, and T. J. Lyons. Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields*, 5(3) :293–318, 1999.
- [10] D. Crişan and A. Doucet. Convergence of sequential Monte Carlo methods. Technical Report CUED/F-INFENG/TR381, Department of Engineering, University of Cambridge, 2000.
- [11] P. Del Moral and J. Jacod. Interacting particle filtering with discrete observations. In [15], chapter 3, pages 43–75.
- [12] P. Del Moral and L. Miclo. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to nonlinear filtering. In J. Azéma, M. Émery, M. Ledoux, and M. Yor, editors, *Séminaire de Probabilités XXXIV*, volume 1729 of *Lecture Notes in Mathematics*, pages 1–145. Springer-Verlag, Berlin, 2000.
- [13] P. Del Moral, G. Rigal, and G. Salut. Estimation et commande optimale non-linéaire. Rapport de fin de contrat DRET, LAAS, Toulouse, Mar. 1992.
- [14] A. Doucet and C. Andrieu. Particle filters for partially observed Gaussian state space models. *Journal of the Royal Statistical Society, Series B*, 64(4) :827–836, Dec. 2002.
- [15] 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.
- [16] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3) :197–208, July 2000.
- [17] 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) :613–624, Mar. 2001.
- [18] W. Fong, S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing with application to audio signal enhancement. In [1], pages 438–449.
- [19] D. Fox, S. Thrun, W. Burgard, and F. Dellaert. Particle filters for mobile robot localization. In [15], chapter 19, pages 401–428.
- [20] 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*, 63(1) :127–146, 2001.
- [21] S. J. Godsill and T. Clapp. Improvement strategies for Monte Carlo particle filters. In [15], chapter 7, pages 139–158.
- [22] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear / non-Gaussian Bayesian state estimation. *IEE Proceedings, Part F*, 140(2) :107–113, Apr. 1993.
- [23] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. In [1], pages 425–437.
- [24] C. Hue. *Méthodes séquentielles de Monte Carlo pour le filtrage non-linéaire multi-objets dans un environnement bruité. Applications au pistage multi-cibles et à la trajectographie d'entités dans des séquences d'images 2D*. Thèse de Doctorat, Université de Rennes 1, Rennes, Jan. 2003. <ftp://ftp.irisa.fr/techreports/theses/2003/hue.pdf>.
- [25] C. Hue, J.-P. Le Cadre, and P. Pérez. Sequential Monte Carlo methods for multiple target tracking and data fusion. In [1], pages 309–325.
- [26] C. Hue, J.-P. Le Cadre, and P. Pérez. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-38(3) :791–812, Aug. 2002.
- [27] M. Hürzeler and H. R. Künsch. Monte Carlo approximations for general state space models. *Journal of Computational and Graphical Statistics*, 7(2) :175–193, June 1998.
- [28] M. Isard and A. Blake. CONDENSATION — Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1) :5–28, Aug. 1998.
- [29] 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.
- [30] A. Kong, J. S. Liu, and W.-H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425) :278–288, Mar. 1994.
- [31] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer-Verlag, New York, 2001.
- [32] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443) :1032–1044, Sept. 1998.
- [33] C. Musso, N. Oudjane, and F. Le Gland. Improving regularized particle filters. In [15], chapter 12, pages 247–271.
- [34] R. M. Neal. Annealing importance sampling. *Statistics and Computing*, 11(2) :125–139, Apr. 2001.
- [35] N. Oudjane. *Stabilité et approximations particulières en filtrage non-linéaire. Application au pistage*. Thèse de Doctorat, Université de Rennes 1, Rennes, Dec. 2000. <ftp://ftp.irisa.fr/techreports/theses/2000/oudjane.pdf>.
- [36] P. Pérez, A. Blake, and M. Gangnet. *JetStream* : Probabilistic contour extraction with particles. In *Proceedings of the 8th International Conference on Computer Vision (ICCV'01), Vancouver 2001*, volume II, pages 524–531. IEEE-CS, July 2001.
- [37] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proceedings of the 7th European Conference on Computer Vision (ECCV'02), Copenhagen 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 661–675. Springer-Verlag, Berlin, June 2002.
- [38] D.-T. Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review*, 129(5) :217–244, 2001.
- [39] M. K. Pitt and N. Shephard. Filtering via simulations : auxiliary particle filter. *Journal of the American Statistical Association*, 94(446) :590–599, June 1999.
- [40] E. Punsakaya, C. Andrieu, A. Doucet, and W. J. Fitzgerald. Particle filtering for demodulation in fading channels. Technical Report CUED/F-INFENG/TR384, Department of Engineering, University of Cambridge, 2000.
- [41] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill. Particle methods for Bayesian modeling and enhancement of speech signals. *IEEE Transactions on Speech and Audio Processing*, SAP-10(3) :173–185, Mar. 2002.
- [42] X. Wang, R. Chen, and J. S. Liu. Monte Carlo Bayesian signal processing for wireless communications. *Journal of VLSI Signal Processing*, 30(1–3 (Special issue on Signal Processing for Wireless Communication Systems)) :89–105, Jan.–Feb.–Mar. 2002.
- [43] Z. Yang and X. Wang. A sequential Monte Carlo blind receiver for OFDM systems in frequency-selective fading channels. In [1], pages 271–280.