Méthodes Monte Carlo Séquentielles pour l'Analyse Spectrale Bayésienne

Manuel DAVY[1], Pierre DEL MORAL[2], Arnaud DOUCET[3]

[1]IRCCyN – CNRS, 1, rue de la noë – BP92101, 44321 Nantes Cedex 3, France

[2]Laboratoire de Statistiques et Probabilités, 31062 Toulouse, France

[3]Engineering Department, Cambridge University, Trumpington street, Cambridge CB2 1PZ, UK

Manuel.Davy@irccyn.ec-nantes.fr, delmoral@cict.fr, ad2@eng.cam.ac.uk

Résumé – Nous présentons l'application d'un nouvel algorithme de simulation stochastique – proposé par deux des auteurs – à la détection/estimation bayésienne de sinusoïdes noyées dans du bruit. Cet algorithme de Monte Carlo Séquentiel génère des particules distribuées marginalement selon la distribution d'intérêt, et procède par échantillonnage/ rééchantillonnage.

Abstract – We present the application of a new simulation algorithm – recently introduced by two of the authors – to joint detection/ estimation of sinusoids in noise. This Sequential Monte Carlo algorithm generates particles that are marginaly distributed according to a target probability distribution via sampling/resampling steps.

1 Introduction

Supposons que l'on veuille simuler des échantillons selon une série de distributions de probabilités $\{\pi_n\}_{n\in\mathcal{N}}$ définies sur un espace E où $\mathcal{N} = \{0, 1, \ldots, p\}$, ou $\mathcal{N} = \mathbb{N}$. Par exemple, on peut considérer le cas où $\pi_n = \pi$ pour tout $n \in \mathcal{N}$. La distribution peut aussi varier en fonction de l'indice $n \in \mathcal{N}$. Par exemple, on pourrait s'intéresser à la série de distributions [1] $\pi_n(x) \propto \pi^{\gamma_n}(x)$ pour une série croissante $\{\gamma_n\}_{n\in\mathcal{N}}$ afin de maximiser π. Mais π_n pourrait être aussi la distribution *a posteriori* d'un paramètre étant données les données collectées jusqu'au temps n.

Les outils pour la simulation favorisés par les statisticiens et les traiteurs de signaux sont les méthodes de Monte Carlo par chaînes de Markov (MCMC) [6]. For simuler selon π_n, les méthodes MCMC construisent un noyau ergodique de Markov K_n de distribution invariante π_n en utilisant des combinaisons de pas de Metropolis-Hastings (MH), Gibbs etc . Les méthodes MCMC ont été utilisées dans de très nombreuses applications en physique, statistique et traitement du signal. Toutefois quand la distribution à échantillonner est multimodale, la chaîne de Markov générée peut être bloquée dans un mode. Une approche standard pour améliorer les résultats consiste à utiliser N chaînes interagissant via des mécanismes de "tempering" ou des procédures de "crossover" comme celles utilisées dans les algorithmes génétiques. Toutes ces méthodes ont en commun qu'elles correspondent à la simulation d'une unique chaîne de Markov avec une distribution invariante fixée sur l'espace produit E^N.

Une autre classe populaire de méthodes de simulation sont les méthodes de Monte Carlo Séquentielles (SMC ou *Sequential Monte Carlo* en anglais) [5]. Les méthodes SMC, appelées aussi méthodes particulaires, sont des méthodes pour simuler des échantillons selon une série de distributions de probabilités $\{\pi_n\}_{n\in\mathcal{N}}$ où π_n est définie sur un espace E_n et $\{E_n\}_{n\in\mathcal{N}}$ est une série d'espaces de dimension croissante; *e.g.* $E_n = \mathbb{R}^n$. L'idée de base consiste à simuler un grand nombre N ($N \gg 1$) d'échantillons $\left\{X_n^{(i)}\right\}_{i=1,\ldots,N}$ appelés particules dont la distribution marginale asymptotique ($N \to \infty$) est égale à π_n. Ces particules sont propagées dans le temps en utilisant une combinaison d'échantillonnage préférentiel, rééchantillonnage et MCMC. Cette approche est très différente de l'approche MCMC où l'on spécifie une loi invariante sur E^N. Toutefois dans leur forme originale ces méthodes ne permettent pas de résoudre le problème décrit précédemment car, dans notre cas, la dimension de l'espace E_n est fixe; $E_n = E$ pour tout n. Deux des auteurs ont récemment montré comment ils étaient possible d'adapter les méthodes SMC à ce cas très important en pratique [2], [3]. Cette nouvelle classe d'algorithmes est une alternative aux méthodes MCMC.

Dans cet article nous présentons une application détaillée de ces nouveaux algorithmes au problème de la détection/ estimation bayésienne jointe de sinusoïdes dans du bruit. Ce problème est complexe car les distributions *a posteriori* à maximiser/ intégrer sont multimodales et hautement non linéaires vis-à-vis des paramètres. Dans [1], un algorithme MCMC a été proposé. Nous présentons ici l'application des méthodes SMC au même problème.

Le reste du papier est organisé de la manière suivante. Dans la section 2, nous décrivons le modèle bayésien hiérarchique et présentons les problèmes numériques liés à l'estimation bayésienne. Dans la section 3, nous décrivons plusieurs algorithmes SMC permettant la résolution de ces problèmes. Finalement en Section 4, nous présentons des simulations.

1. Pour simplifier les notations, on admet que toutes les distributions admettent une densité par rapport à, par exemple, la mesure de Lebesgue.

2 Analyse spectrale Bayésienne

2.1 Modèle Bayésien hiérarchique

Considérons le modèle suivant correspondant à k sinusoïdes dans du bruit

$$\mathcal{M}_k : Y = D\left(\boldsymbol{\omega}_k\right)\boldsymbol{\beta}_k + n,$$

où $Y = (y_0, \ldots, y_{m-1})^{\mathrm{T}}$, $\boldsymbol{\beta}_k = (\beta_1, \ldots, \beta_{2k})^{\mathrm{T}}$, $n = (n_0, \ldots, n_{m-1})^{\mathrm{T}}$, $\boldsymbol{\omega}_k = (\omega_1, \ldots, \omega_k)^{\mathrm{T}} \in (0, \pi)^k$ et $D\left(\boldsymbol{\omega}_k\right)$ est une matrice de dimension $m \times 2k$ où pour $i = 0, \ldots, m-1,\ j = 1, \ldots, k$.

$$\left[D\left(\boldsymbol{\omega}_k\right)\right]_{i+1,2j-1} = \cos\left(\omega_j i\right),\ \left[D\left(\boldsymbol{\omega}_k\right)\right]_{i+1,2j} = \sin\left(\omega_j i\right).$$

Nous supposons que le bruit est blanc gaussien $n|\,\sigma^2 \sim \mathcal{N}\left(0, \sigma^2 I_p\right)$. Ceci définit la vraisemblance des observations Y. Le nombre k de sinusoïdes, leurs amplitudes/phases $\boldsymbol{\beta}_k$ et leurs pulsations $\boldsymbol{\omega}_k$ ainsi que la variance du bruit gaussien sont supposés inconnus. Nous suivons ici une approche bayésienne où ces paramètres suivent une distribution *a priori*. Plus précisément, nous adoptons la distribution *a priori* suivante

$$p\left(k, \sigma^2, \boldsymbol{\beta}_k, \boldsymbol{\omega}_k \middle|\, \delta^2, \Lambda\right) = p\left(k|\,\Lambda\right) p\left(\boldsymbol{\omega}_k|\,k\right) p\left(\boldsymbol{\beta}_k|\,\boldsymbol{\omega}_k, \sigma^2, \delta^2\right) p\left(\sigma^2\right)$$

où $\left(\delta^2, \Lambda\right)$ sont des hyperparamètres et

$$\begin{array}{lll} p(k|\Lambda) & \propto & \frac{\Lambda^k}{k!}, \quad k = 0, \ldots, k_{\max}, \\ \boldsymbol{\omega}_k|k & \sim & \mathcal{U}_{(0,k)^k}, \\ \boldsymbol{\beta}_k|(\boldsymbol{\omega}_k, \sigma^2, \delta^2) & \sim & \mathcal{N}(0, \sigma^2\delta^2[D^{\mathrm{T}}(\boldsymbol{\omega}_k)D(\boldsymbol{\omega}_k)]^{-1}), \\ \sigma^2 & \sim & \mathcal{IG}(\frac{\upsilon_0}{2}, \frac{\gamma_0}{2}) \end{array}$$

Finalement pour rendre le modèle robuste, les hyperparamètres sont supposés aléatoires et vérifient $\delta^2 \sim \mathcal{IG}\left(\alpha_{\delta^2}, \beta_{\delta^2}\right)$ et $\Lambda \sim \mathcal{G}\left(1/2 + \varepsilon_1, \varepsilon_2\right)$ ($\varepsilon_i \ll 1$ $i = 1, 2$). Les notations $\mathcal{G}$, $\mathcal{IG}$, $\mathcal{N}$ et $\mathcal{U}$ désignent respectivement les distributions gamma, inverse-gamma, normale et uniforme; voir [1] pour une justification de ce modèle Bayésien.

2.2 Inférence Bayésienne

Tout estimateur bayésien pour les paramètres $\left(k, \sigma^2, \boldsymbol{\beta}_k, \boldsymbol{\omega}_k\right)$ est basé sur la distribution jointe *a posteriori* $p\left(k, \sigma^2, \boldsymbol{\beta}_k, \boldsymbol{\omega}_k \middle|\, Y\right)$ obtenue en utilisant la formule de Bayes. En particulier, il est possible d'estimer par exemple l'ordre du modèle en utilisant $\arg\max_k\ p\left(k|\,Y\right)$ ou, pour un k donné et des valeurs d'hyperparamètres fixes, d'obtenir

$$\arg\max_{\boldsymbol{\omega}_k} p\left(\boldsymbol{\omega}_k|\,Y, k, \delta^2, \Lambda\right).$$

Toutefois, en pratique ces problèmes n'admettent pas de solution analytique. Calculer les facteurs de Bayes $p\left(k|\,Y\right)$ requiert le calcul et l'intégration numérique de la loi jointe. Maximiser $p\left(\boldsymbol{\omega}_k|\,Y, k, \delta^2, \Lambda\right)$ est un problème d'optimisation difficile. Nous proposons ici d'utiliser les méthodes SMC pour résoudre ces problèmes.

3 Méthodes de Monte Carlo séquentielles

3.1 Algorithme générique

Nous présentons tout d'abord un algorithme générique permettant de simuler selon une série de distributions $\{\pi_n\}_{n\in\mathcal{N}}$. Supposons qu'au temps $n-1$, un ensemble de particules $\left\{X_{n-1}^{(i)}\right\}$ $(i = 1, \ldots N)$ est distribué approximativement selon π_{n-1} est disponible. Au temps n, on simule de nouvelles particules selon $\widetilde{X}_n^{(i)} \sim M_n\left(X_{n-1}^{(i)}, \cdot\right)$ où M_n est une loi de proposition arbitraire. On en déduit que la distribution jointe d'un couple $\left(X_{n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$ est $\pi_{n-1}(x)\, M_n(x, x')$. On désire obtenir des échantillons $\left\{X_n^{(i)}\right\}$ distribués marginalement selon π_n. Pour cela, on utilise une méthode d'échantillonnage préférentiel avec une loi cible $\pi_n(x')\, L_n(x', x)$ où L_n est une distribution conditionnelle de probabilité arbitraire et on attribue à chaque particule le poids

$$G_n\left(X_{n-1}^{(i)}, \widetilde{X}_n^{(i)}\right) = \frac{\pi_n\left(\widetilde{X}_n^{(i)}\right) L_n\left(\widetilde{X}_n^{(i)}, X_{n-1}^{(i)}\right)}{\pi_{n-1}\left(X_{n-1}^{(i)}\right) M_n\left(X_{n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)}.$$

Pour résumer l'algorithme est le suivant.

Méthode de Monte Carlo séquentielle

Initialisation; $n = 0$. *Étape de simulation*

- Pour $i = 1, ..., N$, simuler $\widetilde{X}_0^{(i)} \sim \upsilon_0(\cdot)$.
- Pour $i = 1, ..., N$, calculer les poids d'importance normalisés $W_0^{(i)}$

$$W_0^{(i)} \propto G_0\left(\widetilde{X}_0^{(i)}\right) = \frac{\pi_0\left(d\widetilde{X}_0^{(i)}\right)}{\upsilon_0\left(d\widetilde{X}_0^{(i)}\right)}, \sum_{i=1}^{N} W_0^{(i)} = 1.$$

Étape de réechantillonnage

- Multiplier/Éliminer les particules $\left\{\widetilde{X}_0^{(i)}\right\}$ en fonction de poids grands/petits $\left\{W_0^{(i)}\right\}$ pour obtenir N particules $\left\{X_0^{(i)}\right\}$.

Itération n; $n \in \mathcal{N}\setminus\{0\}$. *Étape de simulation*

- Pour $i = 1, ..., N$, simuler $\widetilde{X}_n^{(i)} \sim M_n\left(X_{n-1}^{(i)}, \cdot\right)$.
- Pour $i = 1, ..., N$, calculer les poids d'importance normalisés $W_n^{(i)} \propto G_n\left(X_{n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$, $\sum_{i=1}^{N} W_n^{(i)} = 1$.

Étape de réechantillonnage

- Multiplier/Éliminer les particules $\left\{\widetilde{X}_n^{(i)}\right\}$ en fonction de poids grands/petits $\left\{W_n^{(i)}\right\}$ pour obtenir N particules $\left\{X_n^{(i)}\right\}$.

Dans cet algorithme, υ_0 est la distribution initiale selon laquelle on simule les particules. L'étape de rééchantillonnage peut être implémentée en utilisant une procédure classique; voir [5]. Au temps n, on peut estimer la distribution π_n par la mesure empirique des particules. Notons que cet algorithme est extrêmement général. Ainsi M_n et L_n sont complètement arbitraires. Toutefois, on ne peut espérer obtenir des performances raisonnables que lorsque la variance des poids d'importance $G_n\left(X_{n-1}^{(i)}, \widetilde{X}_n^{(i)}\right)$ est raisonnable et que le noyau L_n "mélange" rapidement. Enfin, similairement à l'algorithme de Metropolis-Hastings, en pratique cet algorithme ne met a jour qu'une partie des composantes à chaque itération [2].

3.2 Algorithmes SMC pour l'analyse spectrale Bayésienne

Nous présentons maintenant l'algorithme permettant de simuler selon la distribution $p\left(k, \sigma^2, \boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \delta^2, \Lambda \middle|\, Y\right)$. On note tout d'abord que $p\left(k, \boldsymbol{\omega}_k, \delta^2, \Lambda \middle| Y\right)$ admet une expression analytique ce qui permet de réduire la complexité du problème [1]. Nous omettons l'initialisation de l'algorithme qui est triviale. On suppose qu'au temps $n-1$ on dispose de N particules $\left(k^{(i)}, \boldsymbol{\omega}_k^{(i)}, \delta^{2(i)}, \Lambda^{(i)}\right)$ distribuées approximativement selon $p\left(k, \boldsymbol{\omega}_k, \delta^2, \Lambda \middle| Y\right)$. A titre d'exemple, nous présentons l'algorithme obtenu pour l'état $x = [k, \sigma^2, \boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \delta^2, \Lambda]$ en choisissant $L_n = M_n$ avec $\pi_n(x) \propto p\left(k, \sigma^2, \boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \delta^2, \Lambda \middle| Y\right)$ et où M_n est le noyau de proposition présenté dans [1]. L'algorithme est le suivant.

Echantillonnage séquentiel

Iteration n; $n \in \mathcal{N} \setminus \{0\}$.

Echantillonnage

- Pour $i = 1, ..., N$,
  - Avec probabilité $0.5\, P(k_{n+1} \leftarrow k_n + 1 | k_n)$, effectuer une *naissance* et mettre à jour les poids,
  - Avec probabilité $0.5\, P(k_{n+1} \leftarrow k_n - 1 | k_n)$, effectuer une *mort* et mettre à jour les poids,
  - Sinon, effectuer une *mise à jour* et mettre à jour les poids,
  - Pour $i = 1, ..., N$, normaliser les poids afin que $\sum_{i=1}^{N} W_n^{(i)} = 1$.

Rééchantillonnage

- Multiplier/Eliminer les particules $\left\{(\widetilde{k}_n^{(i)}, \widetilde{\boldsymbol{\omega}}_{k,n}^{(i)}, \widetilde{\delta^2}_n^{(i)}, \widetilde{\Lambda}_n^{(i)})\right\}$ en fonction de leur poids $\left\{W_n^{(i)}\right\}$ pour obtenir N les particules $\left\{(k_n^{(i)}, \boldsymbol{\omega}_{k,n}^{(i)}, {\delta^2}_n^{(i)}, \Lambda_n^{(i)})\right\}$.

Echantillonnage additionnel

- Pour $i = 1, ..., N$, échantillonner ${\sigma^2}_n^{(i)} \sim p(\sigma^2 | k_n^{(i)}, \boldsymbol{\omega}_{k,n}^{(i)}, Y)$
- Pour $i = 1, ..., N$, échantillonner
$$\boldsymbol{\beta}_{k,n}^{(i)} \sim p(\boldsymbol{\beta}_k | k_n^{(i)}, \boldsymbol{\omega}_{k,n}^{(i)}, {\sigma^2}_n^{(i)}, Y)$$
- Pour $i = 1, ..., N$, échantillonner ${\delta^2}_n^{(i)} \sim p(\delta^2 | k_n^{(i)}, \boldsymbol{\omega}_{k,n}^{(i)})$
- Pour $i = 1, ..., N$, échantillonner $\Lambda_n^{(i)} \sim p(\Lambda | k_n^{(i)}, \boldsymbol{\omega}_{k,n}^{(i)})$

Dans l'algorithme précédent, $P(k_{n+1} \leftarrow k_n + 1 | k_n)$ (resp. $P(k_{n+1} \leftarrow k_n - 1 | k_n)$ est la probabilité d'effectuer une naissance (resp. la probabilité d'effectuer une mort). Les mouvements correspondants, ainsi que la mise à jour, sont décrits dans les algorithmes suivants.

Naissance

- Faire $\widetilde{k}_n^{(i)} \leftarrow k_{n-1}^{(i)} + 1$
- Echantillonner $\widetilde{\omega}_{\widetilde{k}_n^{(i)},n} \sim \mathcal{U}_{[0,\pi]}(\omega)$ et faire $\widetilde{\boldsymbol{\omega}}_{k,n}^{(i)} \leftarrow [\boldsymbol{\omega}_{k,n-1}^{(i)}; \widetilde{\omega}_{\widetilde{k}_n^{(i)},n}]$
- Faire $W_n^{(i)} \leftarrow G_{\text{naiss}}(k_{n-1}^{(i)}, \boldsymbol{\omega}_{k,n-1}^{(i)}; \widetilde{k}_n^{(i)}, \widetilde{\boldsymbol{\omega}}_{k,n}^{(i)})$ comme est calculé à l'équation (1) ci-dessous.

$$G_{\text{naiss}} = \left[\frac{\gamma_0 + Y^{\mathsf{T}} \widetilde{\mathbf{P}}_{\widetilde{k}_n^{(i)},n}^{(i)} Y}{\gamma_0 + Y^{\mathsf{T}} \mathbf{P}_{k_{n-1}^{(i)},n-1}^{(i)} Y}\right]^{-\frac{N+v_0}{2}} \frac{1}{\widetilde{k}_n^{(i)}(1 + \delta^{2(i)})} \tag{1}$$

où $\mathbf{P}$ est défini comme dans [1]. Plus précisément $\widetilde{\mathbf{P}}_{\widetilde{k}_n^{(i)},n}^{(i)}$ (resp. $\mathbf{P}_{k_{n-1}^{(i)},n-1}^{(i)}$) est calculé en utilisant $\widetilde{k}_n^{(i)}, \widetilde{\boldsymbol{\omega}}_{k,n}^{(i)}$ (resp. $k_{n-1}^{(i)}, \boldsymbol{\omega}_{k,n-1}^{(i)}$).

Mort

- Faire $\widetilde{k}_n^{(i)} \leftarrow k_{n-1}^{(i)} - 1$
- Choisir aléatoirement j uniformément dans $1, \ldots, k_{n-1}^{(i)}$
- Retirer la composante j dans $\boldsymbol{\omega}_{k,n-1}^{(i)}$ pour obtenir $\widetilde{\boldsymbol{\omega}}_{k,n}^{(i)}$
- Faire $W_n^{(i)} \leftarrow G_{\text{mort}}(k_{n-1}^{(i)}, \boldsymbol{\omega}_{k,n-1}^{(i)}; \widetilde{k}_n^{(i)}, \widetilde{\boldsymbol{\omega}}_{k,n}^{(i)})$ où $G_{\text{mort}} = 1/G_{\text{naiss}}$

Mise à jour

- Faire $\widetilde{k}_n^{(i)} \leftarrow k_{n-1}^{(i)}$
- Pour $i = 1, ..., N$,
  - Pour $k = 1, ..., \widetilde{k}_n^{(i)}$,
    - Echantillonner $u \sim \mathcal{U}([0, 1])$,
    - Si $u < \lambda$, alors échantillonner (proposition locale)
    $$\widetilde{\omega}_{j,n} \sim \mathcal{N}(\omega; \omega_{j,n-1}^{(i)}, \sigma_\omega^2) = q_l(\omega)$$
    et faire $\widetilde{\alpha}^{(i)} = q_l(\boldsymbol{\omega}_{k,n-1}^{(i)}) / q_l(\widetilde{\boldsymbol{\omega}}_{k,n}^{(i)}) = 1$
    - Sinon échantillonner (proposition globale, voir [1]) $\widetilde{\omega}_{j,n} \sim q_g(\omega)$ et faire $\widetilde{\alpha}^{(i)} = q_g(\boldsymbol{\omega}_{k,n-1}^{(i)}) / q_g(\widetilde{\boldsymbol{\omega}}_{k,n}^{(i)})$
  - Faire $W_n^{(i)} \leftarrow \left[\frac{\gamma_0 + Y^{\mathsf{T}} \widetilde{\mathbf{P}}_{\widetilde{k}_n^{(i)},n}^{(i)} Y}{\gamma_0 + Y^{\mathsf{T}} \mathbf{P}_{k_{n-1}^{(i)},n-1}^{(i)} Y}\right]^{-\frac{N+v_0}{2}} \widetilde{\alpha}^{(i)}$

On voit que l'algorithme proposé est très proche de l'approche MCMC de [1], et les poids G sont en fait les ratios de Metropolis-Hastings. Par exemple, dans le cas de la mise à jour, on a bien

$$G_n\left(\omega_{n-1}^{(i)}, \widetilde{\omega}_n^{(i)}\right) = \frac{p\left(\widetilde{\omega}_n^{(i)|Y}\right) M_n\left(\omega_{n-1}^{(i)} | \widetilde{\omega}_n^{(i)}\right)}{p\left(\omega_{n-1}^{(i)} | Y\right) M_n\left(\widetilde{\omega}_n^{(i)} | \omega_{n-1}^{(i)}\right)}. \tag{2}$$

Plus généralement, il est très simple de convertir un algorithme MCMC en algorithme de Monte Carlo séquentiel en insérant une boucle portant sur les particules et en effectuant le rééchantillonnage.

4 Simulations

Nous présentons des résultats de simulation selon la distribution $p\left(k, \sigma^2, \boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \delta^2, \Lambda \middle| Y\right)$, avec 500 particules. Le signal traité est composé de six sinusoïdes de fréquences normalisées $\boldsymbol{\omega} = [0.08\ 0.13\ 0.21\ 0.29\ 0.35\ 0.42]$ et d'amplitudes $\boldsymbol{\beta} = [1.24\ 0.00\ 1.23\ 0.43\ 0.67\ 1.00\ 1.11\ 0.39\ 1.31\ 0.16\ 1.28\ 0.13]$, et la longueur des signaux est $m = 200$ échantillons. La variance du bruit additif est $\sigma^2 = 5$, ce qui correspond à un rapport signal à bruit d'environ 0 dB. Le spectre fréquentiel du signal est présenté à la figure 1. Le premier algorithme testé correspond à la simulation de la loi $p\left(k, \sigma^2, \boldsymbol{\beta}_k, \boldsymbol{\omega}_k, \delta^2, \Lambda \middle| Y\right)$, dans le cas $M_n = L_n$ (algorithme présenté dans la section 3. Pour 500 particules, l'algorithme a convergé après environ 50 itérations

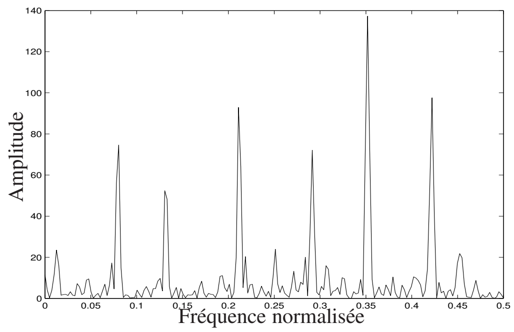

FIG. 1: *Spectre du signal traité (à gauche), estimation MMAP du nombre de sinusoïdes (au centre) et estimation MMSE des fréquences (à droite) au cours des 150 itérations de l'algorithme particulaire, pour 500 particules.*

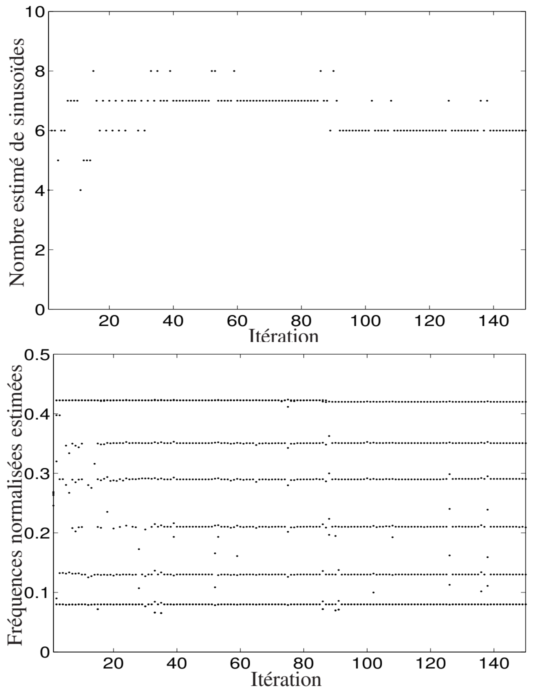

FIG. 2: *Estimation MMAP du nombre de sinusoïdes (en haut) et estimation MMSE des fréquences (en bas) au cours des 150 itérations de l'algorithme particulaire, pour 500 particules.*

(voir figure 2) et l'on estime $\widehat{k} = 6$ par maximum *a posteriori* marginal à l'itération 150. A cette même itération, l'estimation MMSE des fréquences et de σ^2 sont respectivement $\widehat{\omega} = [0.0800\ 0.1301\ 0.2093\ 0.2904\ 0.3505\ 0.4200]$, $\widehat{\sigma}^2 = 4.94$.

Le deuxième algorithme concerne l'optimisation. La loi simulée est proportionnelle à $p\left(k, \sigma^2, \boldsymbol{\omega}_k \middle| Y\right)^{\gamma_n}$ où $\gamma_n = n^{0.2}$, et on garde $M_n = L_n$. Sur la figure 3, on constate que le maximum de la loi *a posteriori* est atteint, avec notamment une bonne estimation des fréquences et du nombre de sinusoïdes.

5 Conclusions

Les algorithmes présentés dans [2] peuvent être utilisés dans de nombreux problèmes de traitement du signal. Dans cet article, nous avons montré leurs performances sur le problème générique de l'estimation de sinusoïdes dans du bruit. D'autres choix d'algorithmes sont étudiés actuellement, notamment des cas où $M \neq L$, ou encore des cas où M est un noyau MCMC.

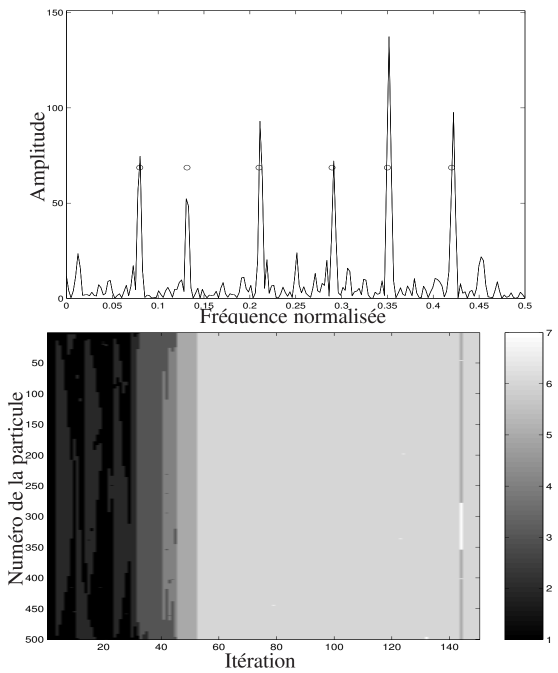

FIG. 3: *Résultats d'optimisation. En haut : les cercles représentent les fréquences des sinusoïdes détectées. En bas : évolution du nombre de sinusoïdes par particule au cours des 150 itérations.*

Références

[1] Andrieu, C. et Doucet, A. (1999) Joint Bayesian detection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Trans. Signal Processing*, **47**, 2667-2676.

[2] Del Moral, P. and Doucet, A. (2002) Sequential Monte Carlo samplers. Technical report Cambridge University CUED-F-INFENG 343. Disponible à `http://www-sigproc.eng.cam.ac.uk/~ad2/arnaud_doucet.html`.

[3] Del Moral, P. et Doucet, A. (2003) On a class of genealogical and interacting Metropolis models. In *Séminaire de Probabilités XXXVII*, Ed. J. Azéma, M. Emery, M. Ledoux et M. Yor, *Lecture Notes in Mathematics*, Berlin : Springer-Verlag, 34pp., à paraître.

[4] Del Moral, P. et Miclo, L. (2000) Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering. In *Séminaire de Probabilités XXXIV*, Ed.Azéma, J., Emery, M., Ledoux, M. and Yor, M., *Lecture Notes in Mathematics*, Berlin: Springer-Verlag, **1729**, 1-145.

[5] Doucet, A., de Freitas, J.F.G. et Gordon, N.J. (eds.) (2001) *Sequential Monte Carlo Methods in Practice*. New York : Springer-Verlag.

[6] Robert, C.P. and Casella, G. (1999) *Monte Carlo Statistical Methods*. New York : Springer Verlag.