Joint Model Selection and Parameter Estimation by Population Monte Carlo Simulation

Mingyi Hong, *Student Member, IEEE*, Mónica F. Bugallo, *Member, IEEE*, and Petar M. Djurić, *Fellow, IEEE*

***Abstract*—In this paper, we study the problem of joint model selection and parameter estimation under the Bayesian framework. We propose to use the Population Monte Carlo (PMC) methodology in carrying out Bayesian computations. The PMC methodology has recently been proposed as an efficient sampling technique and an alternative to Markov Chain Monte Carlo (MCMC) sampling. Its flexibility in constructing transition kernels allows for joint sampling of parameter spaces that belong to different models. The proposed method is able to estimate the desired *a posteriori* distributions accurately. In comparison to the Reversible Jump MCMC (RJMCMC) algorithm, which is popular in solving the same problem, the PMC algorithm does not require burn-in period, it produces approximately uncorrelated samples, and it can be implemented in a parallel fashion. We demonstrate our approach on two examples: sinusoids in white Gaussian noise and direction of arrival (DOA) estimation in colored Gaussian noise, where in both cases the number of signals in the data is *a priori* unknown. Both simulations show the effectiveness of our proposed algorithm.**

***Index Terms*—Bayesian methods, Markov Chain Monte Carlo (MCMC), model selection, Population Monte Carlo (PMC).**

I. Introduction

MODEL selection is an important topic in signal processing. It has found application in various areas including array signal processing, communications, and speech signal processing and therefore has been studied extensively. A recent review provides some common approaches and criteria for model selection [1]. The model selection problem is often presented as a problem of joint model selection and parameter estimation. Many researchers have addressed it within the Bayesian framework. The main difficulty of this approach lies in solving multidimensional integrals. Some early works on model selection are based on the use of large sample theory and approximating the final posterior by Taylor expansion around the maximum-likelihood (ML) estimates of the unknown parameters [2]–[4]. In [5], the authors developed an efficient iterative algorithm for carrying out the maximization needed for obtaining maximum *a posterior* (MAP) estimates. More recently, Reversible Jump Markov Chain Monte Carlo (RJMCMC) sampling [6] has been introduced for approximating joint posteriors and computing estimates of model order and parameters of interest [7]–[9]. Although computationally intensive, this algorithm was shown to have very good performance, especially when the sizes of available data are small. However, algorithms based on RJMCMC have several drawbacks. First, a burn-in period, whose samples are discarded, is required. Second, typical MCMC implementation may have poor mixing, i.e., the chain may converge to a final distribution which depends on the starting point of the chain. Third, one may argue that it is hard to implement the algorithm in a parallel fashion because at each time step t, the algorithm produces only one sample, which depends on the sample produced in the previous time step.

PMC has recently been introduced in [10] and [11]. It is essentially an iterative sampling method which at each iteration employs importance sampling (IS) to produce a set of approximately uncorrelated samples from the target distribution. It also uses resampling [12] to prevent sample degeneration when needed. However, it is well known that for IS, the importance function (IF) needs to be carefully chosen to ensure that the "region of importance" is reached quickly [13]. It is the ability of PMC to accommodate multiple IFs (or rather, transition kernels), and to adaptively improve their sampling efficiency that makes it superior to pure IS. Between different iterations, the algorithm can, based on certain criteria, change the structure of the transition kernels to ensure that the subsequent sampling procedure is carried out more efficiently.

In [11], a fixed number of preselected transition kernels have been used and each of them has been assigned different weights at different iterations. The efficiency of the algorithm has been demonstrated by an example with the posterior being a mixture Gaussian distribution. It has been shown that the produced samples by the algorithm accurately approximate the distribution. In [14], the authors have proposed an algorithm to adaptively choose transition kernels so that the asymptotic variance of the estimates decreases. In [15], it has been demonstrated that the PMC algorithm could progressively sample from distributions that had diminishing Kullback distance from the target distribution. Comparisons between MCMC and PMC have been made in [11] and [16]. In both cases, PMC has outperformed MCMC,

Manuscript received April 20, 2009; revised December 11, 2009; accepted February 24, 2010. First published April 15, 2010; current version published May 14, 2010. This work was supported in part by the National Science Foundation under Award CCF-0515246 and in part by the Office of Naval Research under Award N00014-09-1-1154. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Michael Wicks.

M. Hong is with the Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA 22903 USA (e-mail: mh4tk@virginia.edu).

M. F. Bugallo and P. M. Djurić are with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: monica@ece.sunysb.edu; djuric@ece.sunysb.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2010.2048385

mainly because of the slow mixing property of the MCMC. Improved performance of PMC in parameter estimation by the use of Rao–Blackwellization has been shown in [17].

In this paper, we propose to apply the PMC methodology to joint model selection and parameter estimation. We use a two-stage sampling procedure: we first sample the model order from a set of discrete transition kernels, and then we sample the parameters from a set of transition kernels that correspond to the sampled model. This two-stage sampling procedure allows us to sample from parameter spaces with different dimensions. When the samples are properly weighted, the samples and the weights produce approximations of the desired posteriors.

The paper is organized as follows. In Section II we provide the formulation of the problem, and in Section III, we describe the steps of the algorithm. There we also give a proof of convergence of the algorithm. In Section IV, we introduce two experiments: 1) detection of sinusoids in white Gaussian noise and estimation of their frequencies and 2) detection of number of sources whose signals impinge on an array of sensors. In Section V, we provide numerical results that show the performance of the proposed method and we compare it with the RJMCMC algorithm. Our main objective is to demonstrate that PMC is a valuable alternative to RJMCMC.

II. Formulation of the Problem

In this section, we formulate the problem of joint model selection and parameter estimation in a Bayesian framework. Assume we have an observation vector $\mathbf{y}$ which contains d_y data samples. We also have K competing models $\mathcal{M}_0, \ldots, \mathcal{M}_{K-1}$, and one of them generates the observations. Associated with each model, there is a vector of parameters $\boldsymbol{\theta}_k \in \mathcal{S}_k$, where $\mathcal{S}_k$ denotes the parameter space of $\mathcal{M}_k$. The objective is to identify the true model as well as to estimate the parameters $\boldsymbol{\theta}_k$ associated with the model.

We can view the model order as a realization of a discrete random variable, so that the total parameter space $\boldsymbol{\Theta}$ could be expressed as follows: $\boldsymbol{\Theta} = \cup_{k=0}^{K-1}\{k\} \times \mathcal{S}_k$. Note that each $\mathcal{S}_k$ may have different dimension and may include different parameters. The objective of Bayesian inference is to obtain the posterior $p(k, \boldsymbol{\theta}|\mathbf{y})$, which can then be used (if needed) to compute point estimates.

In the Bayesian context, one typically employs the MAP model selection rule, which can be expressed as

$$\begin{aligned} k^* &= \arg_k \max \{p(k|\mathbf{y})\} \\ &= \arg_k \max \left\{ \int_{\boldsymbol{\theta}\in\mathcal{S}_k} p(k, \boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \right\} \\ &= \arg_k \max \left\{ \int_{(\widetilde{k},\boldsymbol{\theta})\in\boldsymbol{\Theta}} \Pi_k(\widetilde{k}) p(\widetilde{k}, \boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \right\} \end{aligned} \tag{1}$$

where $\Pi_k(\widetilde{k})$ is an indicator function that takes the value 1 when $\widetilde{k} = k$ and is 0 otherwise. The difficulty of using (1) for model selection is that the posterior is usually highly nonlinear in $\boldsymbol{\theta}$, and the integration does not have a close form expression. In that case, one can resort to a Monte Carlo technique to approximate the general integral $\int_{\boldsymbol{\Theta}} f(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$ by first drawing a sample $\boldsymbol{\theta}^{(i)}$ of size N directly from the distribution $p(\boldsymbol{\theta})$, and then performing Monte Carlo integration by

$$\int_{\boldsymbol{\Theta}} f(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \simeq \frac{1}{N} \sum_{i=1}^{N} f\left(\boldsymbol{\theta}^{(i)}\right). \tag{2}$$

From the strong law of large numbers, the above approximation converges to the true value of the integration with probability one. In our case, if one can draw the samples $\{(k^{(i)}, \boldsymbol{\theta}^{(i)})\}_{i=1}^{N}$ directly from the posterior $p(k, \boldsymbol{\theta}|\mathbf{y})$, then (1) becomes

$$k^* \simeq \arg_k \max \left\{ \frac{1}{N} \sum_{i=1}^{N} \Pi_k\left(k^{(i)}\right) \right\}. \tag{3}$$

As stated above, $\Pi_k(k^{(i)})$ is an indicator function, and $\sum_{i=1}^{N} \Pi_k(k^{(i)})$ essentially calculates the total number of drawn samples $\{(k^{(i)}, \boldsymbol{\theta}^{(i)})\}_{i=1}^{N}$ with $k^{(i)} = k$. Thus, one can calculate k^* by first drawing samples $\{(k^{(i)}, \boldsymbol{\theta}^{(i)})\}_{i=1}^{N}$ from $p(k, \boldsymbol{\theta}|\mathbf{y})$, then selecting the model order k that is most frequently sampled.

It is well known that one can directly generate samples from the posterior distribution only in a very few cases. When such generation is impossible, one may resort to the use of IS. In the following, we first briefly review this technique and then apply it to solve the model selection problem.

IS has been used mainly for numerical integration [13] and has many applications in signal processing and communications [18], [19]. For example, in signal processing, in order to obtain the minimum mean square (MMSE) estimate of a parameter x from the observation y, we have to solve the following integration:

$$\hat{x} = \int x p(x|y) dx$$

where $p(x|y)$ is the posterior distribution of x, and $\hat{x}$ is the MMSE estimate of x. The above integration is usually hard to solve directly. It may also be hard to draw samples $x^{(i)}$ directly from the distribution $p(x|y)$ in order to perform the classical Monte Carlo numerical integration. Alternatively, we can draw samples $x^{(i)}$ from an IF $q(x)$, and perform numerical integration as follows:

$$\begin{aligned} \hat{x} &= \int x p(x|y) dx \\ &= \int x \frac{p(x|y)}{q(x)} q(x) dx \\ &\simeq \frac{1}{N} \sum_{i=1}^{N} \frac{x^{(i)} p\left(x^{(i)}|y\right)}{q\left(x^{(i)}\right)}. \end{aligned}$$

It has been shown that when the sample size N is large, the estimate $\hat{x}$ obtained by the above IS algorithm converges to the mean of the posterior. If we define $\tilde{w}^{(i)} = (p(x^{(i)}|y)/q(x^{(i)}))(1/N)$, the above estimate can be interpreted as the weighted mean of the drawn samples $x^{(i)}$.

If $p(x|y)$ is only known up to its proportionality constants (in other words, $p(x|y)$ is unscaled), then we calculate $w^{(i)}$ by

$$\bar{w}^{(i)} = \frac{p\left(x^{(i)}|y\right)}{q\left(x^{(i)}\right)}$$

and normalize the weights according to

$$w^{(i)} = \frac{\bar{w}^{(i)}}{\sum_{j=1}^{N} \bar{w}^{(j)}}.$$

One interesting interpretation of the sample weight pairs $\{x^{(i)}, w^{(i)}\}$ is that they form a discrete random measure $\chi = \{x^{(i)}, w^{(i)}\}$, where the samples $x^{(i)}$ constitute the support of this measure, and $w^{(i)}$ are weights associated to these samples. When the sample size is large, the measure χ "approximates" the posterior distribution of x.

In light of the above IS algorithm, we can draw samples $\{(k^{(i)}, \boldsymbol{\theta}^{(i)})\}_{i=1}^{N}$ (where $\boldsymbol{\theta}^{(i)} \in \mathcal{S}_{k^{(i)}}$) from an IF $g(k, \boldsymbol{\theta})$, and assign each sample a proper weight $w^{(i)}$, so that the random measure $\{(k^{(i)}, \boldsymbol{\theta}^{(i)}), w^{(i)}\}_{i=1}^{N}$ approximates the posterior distribution. We can then approximate the integration in (1) according to

$$\begin{aligned} k^* &= \arg_k \max \left\{ \int_{(\widetilde{k},\boldsymbol{\theta})\in\boldsymbol{\Theta}} \Pi_k(\widetilde{k}) p(\widetilde{k}, \boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \right\} \\ &= \arg_k \max \left\{ \int_{(\widetilde{k},\boldsymbol{\theta})\in\boldsymbol{\Theta}} \Pi_k(\widetilde{k}) \frac{p(\widetilde{k}, \boldsymbol{\theta}|\mathbf{y})}{g(\widetilde{k}, \boldsymbol{\theta})} g(\tilde{k}, \boldsymbol{\theta}) d\boldsymbol{\theta} \right\} \\ &\simeq \arg_k \max \left\{ \frac{1}{N} \sum_{i=1}^{N} \Pi_k\left(k^{(i)}\right) \frac{p\left(k^{(i)}, \boldsymbol{\theta}^{(i)}|\mathbf{y}\right)}{g\left(k^{(i)}, \boldsymbol{\theta}^{(i)}\right)} \right\} \\ &= \arg_k \max \left\{ \sum_{i=1}^{N} \Pi_k\left(k^{(i)}\right) w^{(i)} \right\} \end{aligned} \tag{4}$$

where

$$w^{(i)} \triangleq \frac{p\left(k^{(i)}, \boldsymbol{\theta}^{(i)}|\mathbf{y}\right)}{g\left(k^{(i)}, \boldsymbol{\theta}^{(i)}\right) N}. \tag{5}$$

The above expression indicates that one can approximate the *a posteriori* probability of the model with index k, $p(k|\mathbf{y})$ by summing up all the weights of the samples $\{(k^{(i)}, \boldsymbol{\theta}^{(i)})\}_{i=1}^{N}$, where $k^{(i)} = k$. With the MAP criterion, we choose the model whose sum of weights is maximum. Moreover, one can use the weights to estimate the unknown parameters of each model k by

$$\widehat{\boldsymbol{\theta}}_k = \frac{\sum_{i=1}^{N} w^{(i)} \boldsymbol{\theta}^{(i)} \Pi_k\left(k^{(i)}\right)}{\sum_{i=1}^{N} w^{(i)} \Pi_k\left(k^{(i)}\right)}. \tag{6}$$

The above importance sampling procedure may suffer from poor choices of IF which can, for example, make most of the sample weights negligible and cause the subsequent estimation inaccurate. This situation may arise when the IF generates samples that concentrate on regions with low probability mass under the target posterior. The choice of the IF is both art and science, and many criteria for selecting good IFs have been developed [13]. One can, for example, choose an IF with heavy tails that dominate the tails of the target distribution, or work with an IF that mimics the behavior of the target distribution. However, both of these strategies require certain information about the target distribution, which is usually not available. In the following section, we propose to use a PMC algorithm which obtains samples from an IF that is adaptively modified so that the "quality" of the samples improves with iterations and one can evaluate the integral in (1) with greater accuracy.

III. Proposed Algorithm

PMC has the ability to progressively learn about the target distribution and to adaptively modify the IF based on the gathered information so that the sampling procedure becomes more efficient. We first introduce various elements of the PMC algorithm in Section III-A, and then present the algorithm used to solve our model selection problem in Section III-B. We present the convergence result for the proposed algorithm in Section III-C.

A. Generic PMC Methodology

One way to overcome the main difficulty encountered by IS—the poor choice of IFs—is to introduce into the sampling procedure multiple IFs with different properties, and iteratively and adaptively select them according to their performance. This iterative procedure is a learning process, during which we gain knowledge about the target distribution. For example, in the initial stages of the sampling, it is preferred to have IFs with heavy tails so that the parameter space could be explored fully, while in the later stages, the IFs with good local exploring property may be preferred to stabilize the samples. This process could be implemented in the following way. Before the start of the algorithm, D IFs $g_1, \cdots, g_D$ are selected. Define the vector of random variables $\mathbf{x} \triangleq [k\ \boldsymbol{\theta}]$ and the vector of samples $\mathbf{x}^{(i,t)} \triangleq [k^{(i,t)}\ \boldsymbol{\theta}^{(i,t)}]$, where t stands for iteration. Then the overall IF $g^{(t)}(\mathbf{x})$ can be constructed at each iteration t as follows:

$$\begin{aligned} g^{(t)}(\mathbf{x}) &= \sum_{d=1}^{D} \alpha_d^{(t)} g_d(\mathbf{x}) \\ \sum_{d=1}^{D} \alpha_d^{(t)} &= 1. \end{aligned} \tag{7}$$

It is clear that $g^{(t)}(\mathbf{x})$ is a mixture of D functions and that $\alpha_d^{(t)}$ could be interpreted as frequency of using the IF $g_d(\mathbf{x})$ for sampling. The following procedure is used to sample from $g^{(t)}(\mathbf{x})$. Let $d^{(i,t)} \in \{1, \cdots, D\}$ denote the index of the IF that is used for sampling $\mathbf{x}^{(i,t)}$. Then $d^{(i,t)}$ is determined by drawing it from a multinomial distribution $\mathcal{M}(\alpha_1^{(t)}, \cdots, \alpha_D^{(t)})$ followed by generating $\mathbf{x}^{(i,t)}$ from $g_{d^{(i,t)}}(\mathbf{x})$.

The significance of the above construction of $g^{(t)}(\mathbf{x})$ is that, one can modify $g^{(t)}(\mathbf{x})$ at each iteration by changing the weights $\alpha_d^{(t)}$, according to the performance of each $g_d(\mathbf{x})$ in the past.

One criterion to evaluate the performance of the $g_d(\mathbf{x})$ is the sum of the weights of the samples it generates [11]. This criterion favors the IFs that focus on exploring the so called "region of importance" of the target distribution, and thus generate samples with large total weights. According to this criterion, $\alpha_d^{(t)}$ is updated by the following equation:

$$\alpha_d^{(t)} = \frac{\sum_{i=1}^{N} w^{(i,t-1)} \Pi_{g_d}\left(\mathbf{x}^{(i,t-1)}\right)}{\sum_{i=1}^{N} w^{(i,t-1)}} \tag{8}$$

where $\Pi_{g_d}(\mathbf{x}^{(i,t-1)})$ is an indicator function that takes a value 1 if the sample $\mathbf{x}^{(i,t-1)}$ is generated by the IF $g_d(\mathbf{x})$, and 0 otherwise.

These time-varying IFs $g^{(t)}(\mathbf{x})$ generate N samples at each iteration, and with the weights $\{w^{(i,t)}\}_{i=1}^{N}$, they form a random measure $\mu^{(t)} \triangleq \{\mathbf{x}^{(i,t)}, w^{(i,t)}\}_{i=1}^{N}$ that approximates the distribution of the random variable $\mathbf{x}^{(t)}$. The objective of the algorithm is to ensure that this distribution converges to the target distribution when t is large.

For monitoring the sampling efficiency, we can use the entropy relative to uniformity [20], i.e.,

$$H^{(t)} = -\sum_{i=1}^{N} w^{(i,t)} \frac{\log w^{(i,t)}}{\log N} \tag{9}$$

where $H^{(t)}$ is a measure of uniformity of the weights $w^{(i,t)}$ at time t. If the IF converges to the target distribution, then the weight of each sample $\mathbf{x}^{(i,t)}$ converges to $1/N$, and $H^{(t)}$ converges to 1, and every sample can be seen as drawn approximately from the target distribution.

For the presented iterative IS algorithm, it is beneficial to relate the samples in a current stage with the samples from the previous stage. PMC introduces dependence of the current samples $\{\mathbf{x}^{(i,t)}\}_{i=1}^{N}$ on the samples in the previous iteration $\{\mathbf{x}^{(i,t-1)}\}_{i=1}^{N}$ by replacing the IFs $g_d(\mathbf{x})$ (which do not depend on the past) with transition kernels. This idea is closely related to smoothly approximating the posterior distribution using the kernel technique [20], except that PMC uses multiple kernels at each stage. Recall that $\mathbf{x}^{(t)}, t \in \{1, 2, \ldots T\}$ is the set of random variables approximated by the iterative importance sampling algorithm at each iteration t. A transition kernel $Q(\mathbf{x}^{(i,t-1)}, \mathbf{x})$ is defined as follows:

$$Q\left(\mathbf{x}^{(i,t-1)}, \mathbf{x}\right) = \frac{\partial P\left(\mathbf{x}^{(t)} \leq \mathbf{x} | \mathbf{x}^{(t-1)} = \mathbf{x}^{(i,t-1)}\right)}{\partial \mathbf{x}}. \tag{10}$$

The transition kernel is a generalization of a transition matrix of discrete state Markov chains, where $Q(\mathbf{x}^{(i,t-1)}, \mathbf{x})$ is the probability density of the state of the chain at iteration t conditional on the chain being in state $\mathbf{x}^{(i,t-1)}$ at iteration $t-1$. We can replace our previous IF with a transition kernel $Q(\mathbf{x}^{(i,t-1)}, \mathbf{x})$, where $Q(\mathbf{x}^{(i,t-1)}, \mathbf{x})$ can take the form, for example, of a Gaussian kernel (in the case when $\mathbf{x}$ is one-dimensional): $Q(x^{(i,t-1)}, x) = (1/\sqrt{2\pi\sigma^2})\exp\{-((x - x^{(i,t-1)})^2/2\sigma^2)\}$. It is clear that the new samples $x^{(i,t)}$ drawn from this kernel will be located near $x^{(i,t-1)}$, and the shape of the kernel is determined by σ^2. In this case, the kernel $Q(x^{(i,t-1)}, x)$ is a family of normal distributions with $x^{(i,t-1)}$ as location parameter.

PMC uses a mixture of multiple kernels $\{Q_d(., \mathbf{x})\}_{d=1}^{D}$ in each iteration to improve the sampling efficiency, and the sampling is performed as in (7), i.e.,

$$\mathbf{x}^{(i,t)} \sim \sum_{d=1}^{D} \alpha_d^{(t)} Q_d\left(\mathbf{x}^{(i,t-1)}, \mathbf{x}\right) \tag{11}$$

$$\sum_{d=1}^{D} \alpha_d^{(t)} = 1 \tag{12}$$

where $\alpha_d^{(t)}$ is determined as in (8).

We summarize the procedure by explaining how we generate one particular sample at iteration t. First, we select the index d of the transition kernel from the multinomial distribution $\mathcal{M}(\alpha_1^{(t)}, \cdots, \alpha_D^{(t)})$; then we generate a sample $\mathbf{x}^{(i,t)}$ by the kernel with index d. Because the sample $\mathbf{x}^{(i,t)}$ is actually generated by the kernel $Q_d(\mathbf{x}^{(i,t-1)}, \mathbf{x}^{(i,t)})$, the weight of this sample can be obtained by

$$w^{(i,t)} \propto \frac{p\left(\mathbf{x}^{(i,t)} | \mathbf{y}\right)}{Q_d\left(\mathbf{x}^{(i,t-1)}, \mathbf{x}^{(i,t)}\right)} \tag{13}$$

which follows from (5). We use the proportionality operator $\propto$ here to include the case when the posterior distribution is not scaled. Equation (13) represents the weight in the original PMC algorithm, and it has been used in many applications of the PMC, for example, [16], [17], and [21].

The samples produced by so called iterated particle systems [18] like PMC and particle filters [22] may be degenerated, i.e., the weights of a few samples dominate the remaining samples. Resampling [23] is thus introduced to prevent the samples from degeneration. We use $\{x^{*(i,t)}\}_{i=1}^{N}$ to denote the samples after resampling.

We summarize a generic PMC implementation as follows. For each iteration t:

1) generate samples from the mixture of transition kernels;
2) compute the weights of the samples $w^{(i,t)}$;
3) perform estimation based on the weights;
4) resample;
5) compute the weights $\alpha_d^{(t+1)}$ for each kernel $Q_d(\cdot, \cdot)$.

B. PMC for Model Selection

In a parameter estimation problem only, as demonstrated in [11], [17], the above steps are sufficient to produce approximation of the target distribution. Under model uncertainty, a direct extension of the above method would be to run K parallel PMC algorithms, one for each model, and compare their performance to determine the model. Of course, this naive approach is very computationally expensive when the number of competing models K is large, because the computational load is "equally" distributed upon the different models. Similar observation, which serves as inspiration for using RJMCMC instead of multiple MCMC under model uncertainty, has been made in [8].

Observe that there is a natural hierarchy in our full parameter space $\boldsymbol{\Theta}$: the space for the parameter $\boldsymbol{\theta}$ is determined only by the choice of k. After the determination of k, it is sufficient to sample only from the parameter space $\mathcal{S}_k$ to produce samples

that approximate the distribution $p(\boldsymbol{\theta}|k,\mathbf{y})$. It is thus natural to decompose our sampling kernels into two components: one for sampling of the model order k from the index set $\{k\}_{k=0}^{K-1}$, and the other for sampling from the parameter space $\mathcal{S}_k$. Let $Q(\cdot,\cdot)$ denote the kernel we use for sampling the model order, and let $Q_{\theta_k}(\cdot,\cdot)$ denote the kernel we use for sampling $\boldsymbol{\theta}_k$. We propose the following two-stage sampling scheme:

At iteration t:

1) draw a model order sample $k^{(i,t)}$ from $Q(k^{(i,t-1)},k)$;
2) draw a parameter sample $\boldsymbol{\theta}^{(i,t)}$ from $Q_{\theta_{k^{(i,t)}}}(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta})$, where $\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)}$ is the estimate of the parameter of the specific model at iteration $t-1$.

It is worth mentioning that the model order is a discrete random variable, so that the kernel $Q(\cdot,\cdot)$ is represented by a $K\times K$ matrix with transition probabilities. For example, if $Q(\cdot,\cdot)=\mathbf{I}_{K\times K}$, which is a K by K identity matrix, we will always have $k^{(i,t-1)}=k^{(i,t)}$.

In the following we use D predetermined kernels $Q_d(.,.)$ for generating model order, where $d=1,2,\cdots,D$, and for each model k, we have a single kernel $Q_{\theta_k}(\cdot,\cdot)$ for generating the parameters that belong to the parameter space $\mathcal{S}_k$. One reason of using multiple model kernels is to improve the sampling efficiency, as stated in Section III-A. An extension can be easily made to generate parameters using multiple kernels under each model. However, the rationale is the same, and we keep our choice of single kernel for the parameters in order to maintain the presentation clear.

Based on the above decomposition of transition kernels, the sampling of $(k^{(i,t)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)})$ can be expressed by

$$\left(k^{(i,t)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}\right)\sim\sum_{d=1}^{D}\alpha_d^{(t)}Q_d\left(k^{(i,t-1)},k\right)Q_{\theta_k}(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$$

and the weight for each sample can be expressed according to (13) as follows:

$$w^{(i,t)}\propto\frac{p\left(k^{(i,t)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}|\mathbf{y}\right)}{Q_{d^{(i,t)}}\left(k^{(i,t-1)},k^{(i,t)}\right)Q_{\theta_{k^{(i,t)}}}\left(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}\right)}\tag{14}$$

where $\sum_{i=1}^{N}w^{(i,t)}=1$. The estimate of the marginal posterior for the model order can be obtained by summing up the normalized weights for each model

$$P(k|\mathbf{y})\simeq\sum_{i=1}^{N}w^{(i,t)}\Pi_k\left(k^{(i,t)}\right).$$

The resampling procedure equalizes the weight of the samples, and is commonly carried out by duplicating samples proportional to their weights [24]. Namely, for $i=1,\cdots,N$, we first sample the index $j_i\sim\mathcal{M}(w^{(1,t)},\cdots,w^{(N,t)})$ and then we let $\mathbf{x}^{*(i,t)}=\mathbf{x}^{(j_i,t)}$. After resampling, the models with large total weights (thus large estimated marginal posterior) occupy greater portion of the sample.

Now, we can provide another justification for using multiple kernels $Q_d(\cdot,\cdot)$ for sampling the model order. Let $\mathbf{U}$ denote a matrix with entries all ones. By choosing either $Q_d(\cdot,\cdot)=(1/K)\mathbf{U}$ or $Q_d(\cdot,\cdot)=\mathbf{I}$, and assigning $\alpha_d^{(1)}=1/D$ for all d in the initial stage of the algorithm, there will always be equal number of samples for each model at every iteration, which corresponds to the naive approach mentioned in the beginning of this section. Consequently, it is desirable to find a transition kernel that distributes the right amount of computational time to each competing model, preferably according to their true posterior, i.e., the models with high posteriors should get more "attention." Since the true posterior is not known, one can use multiple transition kernels and let the algorithm choose the most efficient distribution of computational time. In Section V, we demonstrate that, indeed, by using D different transition kernels, we get better performance than when we use only one kernel.

We also provide some heuristic guidelines for designing the set of predetermined kernels $Q_d(\cdot,\cdot)$ for model order. First, we would like to build our current computation based on the distribution obtained from the previous iteration. As a result, we would expect that the majority of the model order samples $\{k^{(i,t)}\}_{i=1}^{N}$ generated at iteration t correspond to the model orders with the largest total weights in iteration $t-1$. Second, we would also like to improve upon the previous distribution by exploring the parameter spaces more thoroughly. One way to achieve this is to allow portions of the majority samples in iteration $t-1$ to "move" to the other model orders, and this behavior is determined by the off-diagonal entries of $Q_d(\cdot,\cdot)$. Specifically, if $Q_d(2,3)=0.5$, then 50% of the samples that have model order $k^{(i,t-1)}=2$ at iteration $t-1$ will be moved to $k^{(i,t)}=3$ in the next iteration. Now it is clear that the above requirements are somewhat conflicting with each other: the first requirement translates to the strategy of selecting $Q_d(\cdot,\cdot)$ that have large diagonal entries, while the second requirement dictates how to choose $Q_d(\cdot,\cdot)$ with off-diagonal entries not too small. As a result, we suggest to design multiple kernels that achieve a tradeoff between these requirements: some kernels may have negligible off-diagonal entries, while some may have relatively large off-diagonal entries. Our choice of transition kernels in the following simulation section is an example of the above suggested design. We will also show in the simulations that strategies that do not obey the above suggestions may lead to poor performance of the PMC algorithm.

In Table I, we summarize the proposed algorithm and in Fig. 1, we present a graphical illustration of it.

C. Convergence

A natural question is whether the above stated algorithm can approximate the target distribution, or specifically, if for each iteration t, when the number of samples goes to infinity, we have

$$\lim_{N\to\infty}\sum_{i=1}^{N}w^{(i,t)}f\left(\mathbf{x}^{(i,t)}\right)=\int f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x}\tag{15}$$

for any function $f(\cdot)$ that satisfies certain regularity conditions. If (15) is true, then when the function $f(\cdot)$ is $\Pi_k(k^{(i,t)})$, the above equation becomes

$$\lim_{N\to\infty}\sum_{i=1}^{N}w^{(i,t)}\Pi_k\left(k^{(i,t)}\right)=\sum_{j=1}^{K}\Pi_k(j)\int_{\boldsymbol{\theta}\in S_j}p(j,\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}\tag{16}$$

$$=p(k|\mathbf{y}).\tag{17}$$

TABLE I
PROPOSED ALGORITHM

At $t = 0$
  For $i = 1, \cdots, N$
    Draw $k^{(i,0)}$ from the prior $p(k)$
    Draw $\boldsymbol{\theta}^{(i,0)}$ from the prior $p(\boldsymbol{\theta}_{k^{(i,0)}})$
    Compute the normalized weights $w^{(i,0)} \propto p(\mathbf{y}|k^{(i,0)}, \boldsymbol{\theta}^{(i,0)}_{k^{(i,0)}})$
  Obtain estimates $\widehat{k}^{(0)}$, $\widehat{\boldsymbol{\theta}}^{(0)}$ using $\{w^{(i,0)}\}_{i=1}^{N}$ and the samples
  Assign $\alpha_d^{(1)} = 1/D, d = 1, \cdots, D$
  Resample and get $k^{*(i,0)}$, $\boldsymbol{\theta}^{*(i,0)}$
For $t = 1, \cdots, T$
  For $i = 1, \cdots, N$
    Determine $d^{(i,t)}$ by sampling from $\mathcal{M}(\alpha_1^{(t)}, \cdots, \alpha_D^{(t)})$
    Draw $k^{(i,t)}$ from $Q_{d^{(i,t)}}(k^{*(i,t-1)}, k)$
    Draw $\boldsymbol{\theta}^{(i,t)}_{k^{(i,t)}}$ from $Q_{\theta_{k^{(i,t)}}}(\widehat{\boldsymbol{\theta}}^{(t-1)}_{k^{(i,t)}}, \boldsymbol{\theta})$
    Compute the normalized weights

$$w^{(i,t)} \propto \frac{p(\mathbf{y}|k^{(i,t)}, \theta^{(i,t)}_{k^{(i,t)}})}{Q_{d^{(i,t)}}(k^{*(i,t-1)}, k^{(i,t)}) Q_{\theta_{k^{(i,t)}}}(\widehat{\boldsymbol{\theta}}^{(t-1)}_{k^{(i,t)}}, \boldsymbol{\theta}^{(i,t)}_{k^{(i,t)}})}$$

  Obtain estimates $\widehat{k}^{(t)}$, using (4)
  Obtain estimates $\{\widehat{\boldsymbol{\theta}}_k^{(t)}\}_{k=0}^{K-1}$ using (6)
  Assign $\alpha_d^{(t+1)} = \sum_{i=1}^{N} w^{(i,t)} \Pi_d(d^{(i,t)})$
  Resample to obtain $k^{*(i,t)}$ and $\boldsymbol{\theta}^{*(i,t)}$.

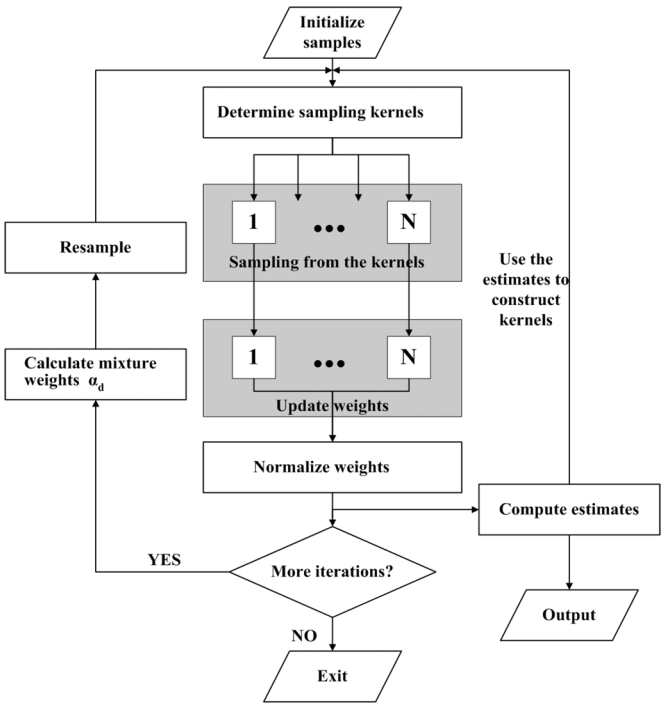

Fig. 1. Graphical illustration of the proposed algorithm.

We prove the following theorem regarding the convergence of the PMC algorithm for model selection.

Theorem 1: For the PMC algorithm detailed in Table I, we have the following convergence result:

$$\sum_{i=1}^{N} w^{(i,t)} f\left(\mathbf{x}^{(i,t)}\right) \overset{N\to\infty}{\longrightarrow}_p \int f(\mathbf{x}) p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \tag{18}$$

or more specifically, when $f(\cdot)$ takes the form of $\Pi_k(k^{(i,t)})$, we have

$$\sum_{i=1}^{N} w^{(i,t)} \Pi_k\left(k^{(i,t)}\right) \overset{N\to\infty}{\longrightarrow}_p p(k|\mathbf{y}).$$

where "$\longrightarrow_p$" stands for convergence in probability.

Proof: The proof is given in the Appendix. ■

D. Discussion

It is easy to extend the above algorithm to support multiple kernels for exploring the parameter space $\mathcal{S}_{k^{(i)}}$ instead of using a single kernel. Alternatively, we can adaptively change the single kernel $Q_{\theta_{k^{(i)}}}(\cdot,\cdot)$ to achieve improved sampling efficiency. Assume that $Q_{\theta_{k^{(i,t)}}}(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t+1)})$ is a Gaussian kernel $\mathcal{N}(\boldsymbol{\theta}^{(t)}, \sigma^2_{k^{i,t}}\mathbf{I})$. We propose to use a time-varying kernel $Q_{\theta_{k^{(i,t)}}}(\cdot,\cdot) = \mathcal{N}(\boldsymbol{\theta}^{(t)}, \sigma^2_{k^{(i,t)}}\mathbf{I})$ as follows. If $k^{(i,t)}$ is the MAP estimate of the model order at iteration t, let

$$\sigma^2_{k^{(i,t+1)}} = \frac{t}{t+c}\sigma^2_{k^{(i,t)}} \tag{19}$$

and else, let

$$\sigma^2_{k^{(i,t+1)}} = \frac{t}{t+c}\sigma^2_{k^{(i,t)}} + \frac{c}{t+c}\sigma^2_{k^{(i,1)}} \tag{20}$$

where c is a constant. It is clear that if t is large enough and the estimates for $k^{(i,t)}$ stabilize, then $\sigma^2_{k^{(i,t)}}$ stabilizes for all $k^{(i,t)}$. By setting $\sigma^2_{k^{(i,1)}}$ relatively large, the above scheme allows us to have kernels with heavy tails at the initial stages of the algorithm so that we can explore the whole parameter space, and have lighter tails to focus on local sampling in the later stages of the algorithm. Once the order estimates stabilize, $\sigma^2_{k^{(i,t)}}$ shrinks and allows the kernel to explore the local parameter space instead. We will demonstrate the improvement of the sampling efficiency for this choice of $Q_{\theta_{k^{(i,t)}}}(\cdot,\cdot)$.

It is also worth mentioning that our proposed algorithm is capable of sampling nuisance parameters if their conditional distributions are known. Specifically, denote with $\mathbf{z}$ the vector of nuisance parameters. Then we can sample $\mathbf{z}$ at iteration t according to $\mathbf{z}^{(i,t)} \sim p(\mathbf{z}^{(i,t)}|k^{*(i,t)}, \boldsymbol{\theta}^{*(i,t)}, \mathbf{y})$, where $\boldsymbol{\theta}^{*(i,t)}$ and $k^{*(i,t)}$ are the resampled parameters of interest and the model order, respectively.

IV. EXAMPLES

In this section, we present two examples that will later be used to test the performance of the proposed PMC algorithm. The first example is the joint detection and estimation of sinusoids in white Gaussian noise [5]. The second example is the joint detection of number of sources and estimation of direction-of-arrival (DOA) of signals emitted by the sources [25]. The signals impinge on an array of sensors and are corrupted by colored Gaussian noise. In Sections IV-A and IV-B, we briefly present the mathematical model for these problems. For their detailed description, we refer the readers to [8] and [9].

A. Detection and Estimation of Sinusoids in White Noise

We have an observation vector $\mathbf{y}$ with d_y data samples. The observations are generated by one of the following K models:

$$\mathcal{M}_0 : y[n] = \epsilon[n]$$
$$\mathcal{M}_k : y[n] = \sum_{j=1}^{k} a_{c,j}\cos(2\pi f_j\, n) + a_{s,j}\sin(2\pi f_j\, n) + \epsilon[n]$$

where $n = 0, 1, \cdots, d_y - 1$, $k = 1, \ldots, K-1$, $\epsilon[n]$ are independent and identically distributed (i.i.d.) noise samples, and $\epsilon[n] \sim \mathcal{N}(0, \sigma^2)$. In matrix form, the observation vector can be expressed as follows:

$$\mathbf{y} = \mathbf{H}(\mathbf{f}_k)\mathbf{a}_k + \boldsymbol{\epsilon} \tag{21}$$

where $\mathbf{y}$ is the $d_y \times 1$ observation vector, $\boldsymbol{\epsilon}$ is a $d_y \times 1$ Gaussian noise vector, i.e., $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, $\mathbf{f}_k$ is a $k \times 1$ frequency vector defined as $\mathbf{f}_k \triangleq [f_1\ f_2\ \cdots\ f_k]^\top$, $\mathbf{a}_k$ is a $2k \times 1$ amplitude vector, and $\mathbf{H}(\mathbf{f}_k)$ is a $d_y \times 2k$ matrix whose elements can be expressed according to

$$\mathbf{H}(\mathbf{f}_k)_{n,2j-1} = \cos(2\pi f_j n)$$
$$\mathbf{H}(\mathbf{f}_k)_{n,2j} = \sin(2\pi f_j n).$$

The unknown parameter vector is $\boldsymbol{\theta}_k \triangleq [\sigma^2\ \mathbf{a}_k^\top\ \mathbf{f}_k^\top]^\top$. Our objective is to jointly determine which model generated the observations $\mathbf{y}$ and estimate the parameters $\mathbf{f}_k$. Note that σ^2 and $\mathbf{a}_k$ are treated as nuisance parameters. Therefore, we use the Bayesian methodology: we integrate out the nuisance parameters and determine the model order and frequency from the joint posterior distribution $p(k, \mathbf{f}_k|\mathbf{y})$.

We first assign prior distributions to the parameters. We assume the models have equal prior probability; thus,

$$p(k) = \frac{1}{K}. \tag{22}$$

Note, that in the RJMCMC setting, the prior for model order is usually set to be a truncated Poisson distribution $p(k) \propto \Lambda^k exp(-\Lambda)/k! \times \Pi\{0 \le k \le K-1\}$ with hyperparameter Λ in order to facilitate the "Birth" and "Death" moves. We use the Jeffreys' prior for the noise variance [26]

$$p(\sigma^2) \propto \frac{1}{\sigma^2}. \tag{23}$$

Then we can write the joint posterior distribution $p(k, \boldsymbol{\theta}_k|\mathbf{y})$ as follows:

$$\begin{aligned} p(k, \boldsymbol{\theta}_k|\mathbf{y}) &\propto p(\mathbf{y}, k, \boldsymbol{\theta}_k) \\ &= p(\mathbf{y}|k, \boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k, k) \\ &\propto p(\mathbf{y}|k, \mathbf{a}_k, \mathbf{f}_k, \sigma^2)p(k, \mathbf{a}_k, \mathbf{f}_k|\sigma^2)p(\sigma^2). \end{aligned} \tag{24}$$

Assuming, that $\mathbf{f}_k$ has a uniform prior on $[0, (1/2)]^k$ and $\mathbf{a}_k$ has a zero mean normal prior, we have

$$p(k, \mathbf{a}_k, \mathbf{f}_k|\sigma^2) \propto \frac{1}{|2\pi\sigma^2\boldsymbol{\Sigma}_k|^{1/2}} \exp\left\{\frac{-\mathbf{a}_k^\top\boldsymbol{\Sigma}_k^{-1}\mathbf{a}_k}{2\sigma^2}\right\} 2^k \tag{25}$$

where $\boldsymbol{\Sigma}_k^{-1} = \delta^{-2}\mathbf{H}^\top(\mathbf{f}_k)\mathbf{H}(\mathbf{f}_k)$, with δ^2 being a hyperparameter that can be integrated out numerically if we choose for it an inverse gamma prior of the form $\mathcal{IG}(\alpha_\delta, \beta_\delta)$. (See [8, Sec. V-C] for detailed discussion.)

After integrating out $\mathbf{a}_k$ and σ^2 in (24), we obtain

$$p(k, \mathbf{f}_k|\mathbf{y}) \propto \left(\mathbf{y}^\top\mathbf{P}_k^\perp\mathbf{y}\right)^{-d_y/2} \frac{2^k}{(\delta^2+1)^k} \tag{26}$$

where

$$\mathbf{P}_k^\perp = \mathbf{I} - \mathbf{H}(\mathbf{f}_k)\mathbf{M}_k^{-1}\mathbf{H}^\top(\mathbf{f}_k)$$

and

$$\mathbf{M}_k = \mathbf{H}^\top(\mathbf{f}_k)\mathbf{H}(\mathbf{f}_k) + \boldsymbol{\Sigma}_k^{-1}.$$

In Section V-A, we show that our PMC algorithm is capable of approximating the marginalized distribution (26).

B. Detection of Number of Sources and Estimation of DOA

We have an $M \times d_y$ complex observation matrix $\mathbf{Y}$. Each column of $\mathbf{Y}$, $\mathbf{y}[n]$, is an $M \times 1$ vector representing the data received by a linear array of M sensors that can be expressed as

$$\mathbf{y}[n] = \mathbf{H}(\boldsymbol{\phi}_k)\mathbf{a}[n] + \boldsymbol{\epsilon}[n] \tag{27}$$

where $\boldsymbol{\epsilon}[n]$ is an $M \times 1$ zero mean Gaussian noise with covariance matrix $\boldsymbol{\Sigma}$, $\mathbf{a}[n]$ is a $k \times 1$ amplitude vector, and $\mathbf{H}(\boldsymbol{\phi}_k)$ is an $M \times k$ matrix whose elements can be expressed by

$$\mathbf{H}(\boldsymbol{\phi}_k)_{m,l} = \exp\{j(m-1)\phi_l\} \tag{28}$$

where $j = \sqrt{-1}$, $m = 0, 1, \cdots, M-1$, and $l = 1, \cdots, k$. The vector $\boldsymbol{\phi}_k$ is defined by $\boldsymbol{\phi}_k \triangleq [\phi_1, \ldots, \phi_k]^\top$, where ϕ_l is given by $\phi_l = \omega_0\lambda_0\sin(\varphi_l)$ with φ_l being the angle between the l^{th} incident signal and the sensor array, ω_0 the carrier frequency of the received signal, λ_0 the distance between the sensors, and v the propagation speed of the signal. In summary, each element $y_m[n]$ of $\mathbf{y}[n]$ could be expressed as

$$y_m[n] = \sum_{l=1}^{k} a_l[n]\exp\{j(m-1)\phi_l\} + \epsilon_m[n]. \tag{29}$$

We need to determine which one of the following K models generates $\mathbf{y}$:

$$\mathcal{M}_0 : \mathbf{y}[n] = \boldsymbol{\epsilon}[n]$$
$$\mathcal{M}_k : \mathbf{y}[n] = \mathbf{H}(\boldsymbol{\phi}_k)\mathbf{a}[n] + \boldsymbol{\epsilon}[n] \quad k = 1, \cdots, K-1.$$

Besides determining the model order, we need to estimate $\boldsymbol{\phi}_k$. Again we integrate out the nuisance parameters $\mathbf{A} \triangleq [\mathbf{a}[1]\ \mathbf{a}[2]\ \ldots\ \mathbf{a}[d_y]]$ and $\boldsymbol{\Sigma}$, and estimate the model order k and $\boldsymbol{\phi}_k$ using the marginalized posterior distribution $p(k, \boldsymbol{\phi}_k|\mathbf{y})$.

Define the projector on the signal subspace by [5]

$$\begin{aligned} \mathbf{P}_k &= \mathbf{H}(\boldsymbol{\phi}_k)\left(\mathbf{H}(\boldsymbol{\phi}_k)^{\mathsf{H}}\mathbf{H}(\boldsymbol{\phi}_k)\right)^{-1}\mathbf{H}(\boldsymbol{\phi}_k)^{\mathsf{H}} \\ &= \mathbf{U}_s(\boldsymbol{\phi}_k)\mathbf{U}_s^{\mathsf{H}}(\boldsymbol{\phi}_k) \end{aligned} \tag{30}$$

and let the projector on the noise subspace be

$$\begin{aligned}\mathbf{P}_k^{\perp} &= \mathbf{I} - \mathbf{P}_k \\ &= \mathbf{U}_\epsilon(\boldsymbol{\phi}_k)\mathbf{U}_\epsilon^{\mathsf{H}}(\boldsymbol{\phi}_k)\end{aligned} \tag{31}$$

where $\mathbf{U}_s(\boldsymbol{\phi}_k)$ and $\mathbf{U}_\epsilon(\boldsymbol{\phi}_k)$ are orthogonal matrices whose columns span the signal subspace and the noise subspace, respectively. Define

$$\mathbf{z}_s[n] = \mathbf{U}_s^{\mathsf{H}}(\boldsymbol{\phi}_k)\mathbf{y}[n] \tag{32}$$

$$\mathbf{z}_\epsilon[i] = \mathbf{U}_\epsilon^{\mathsf{H}}(\boldsymbol{\phi}_k)\mathbf{y}[n]. \tag{33}$$

Let $\mathbf{Z}_s \triangleq [\mathbf{z}_s[1], \cdots, \mathbf{z}_s[d_y]]$, and define $\mathbf{Z}_\epsilon$ similarly. We can show that

$$\begin{aligned}&p(\mathbf{Z}_s, \mathbf{Z}_\epsilon|\tilde{\mathbf{A}}, \boldsymbol{\phi}, k, \mathbf{W}^{-1}) \\ &\quad\simeq \pi^{-d_y k}|\mathbf{C}^{-1}|^{d_y} \\ &\qquad\times \exp\left\{-\sum_{n=1}^{d_y}(\mathbf{z}_s[n]-\tilde{\mathbf{a}}[n])^{\mathsf{H}}\mathbf{C}^{-1}(\mathbf{z}_s[n]-\tilde{\mathbf{a}}[n])\right\} \\ &\qquad\times \pi^{-d_y(M-k)}|\mathbf{W}^{-1}|^{d_y} \\ &\qquad\times \exp\left\{-\sum_{n=1}^{N}\mathbf{z}_\epsilon^{\mathsf{H}}[n]\mathbf{W}^{-1}\mathbf{z}_\epsilon[n]\right\}\end{aligned} \tag{34}$$

where $\tilde{\mathbf{A}} = [\tilde{\mathbf{a}}[1]\ \tilde{\mathbf{a}}[2]\ \cdots\ \tilde{\mathbf{a}}[d_y]]$, $\tilde{\mathbf{a}}[n] \triangleq \mathbf{U}_s^{\mathsf{H}}\mathbf{H}\mathbf{a}[n]$, $\mathbf{C} \triangleq \mathbf{U}_s^{\mathsf{H}}\boldsymbol{\Sigma}\mathbf{U}_s$, and $\mathbf{W} \triangleq \mathbf{U}_\epsilon^{\mathsf{H}}\boldsymbol{\Sigma}\mathbf{U}_\epsilon$. We will then assign the priors to the different sets of parameters and integrate out the nuisance parameters. Let [25]

$$p(k) = \frac{1}{K} \tag{35}$$

$$p(\boldsymbol{\phi}_k|k) = U[0, 2\pi]^k \tag{36}$$

$$p(\mathbf{W}^{-1}|\boldsymbol{\phi}_k, k) \propto |\mathbf{W}^{-1}|^{-(M-k)} \tag{37}$$

$$p(\tilde{\mathbf{A}}|\boldsymbol{\phi}_k, k, \mathbf{W}^{-1}) = \prod_{n=1}^{d_y}\mathcal{N}(0, \rho^2\mathbf{I}_k) \tag{38}$$

where ρ^2 is a hyperparameter that can be determined numerically (see [9, Sec. V] for discussion). Finally, after integrating out $\mathbf{W}^{-1}$ and $\tilde{\mathbf{A}}$, we get

$$p(k, \boldsymbol{\phi}|\mathbf{Z}_\epsilon) \propto \frac{\pi^{1/2(M-k)(M-k-1)}\prod_{n=1}^{M-k}\Gamma(N_y - n + 1)}{(2\pi)^k(\rho^2)^{kN_y}}|\widehat{\mathbf{R}}|^{-d_y} \tag{39}$$

where $\widehat{\mathbf{R}} \triangleq \sum_{i=1}^{d_y}\mathbf{z}_\epsilon[n]\mathbf{z}_\epsilon[n]^{\mathsf{H}}$, and $\Gamma(d_y - n + 1)$ is the Gamma function with argument $d_y - n + 1$. In Section V-B, we show that our PMC algorithm can approximate (39).

V. Numerical Results

In this section, we demonstrate the performance of the PMC algorithm by applying it respectively to the problem of sinusoids in white noise and the problem of DOA with colored noise sketched in Section IV.

A. Numerical Results for Sinusoids in White Noise

We used the following setup for the experiment. We set $d_y = 64$, $k = 2$, and $f_1 = 0.2$, $f_2 = 0.2 + 1/(l_e \times d_y)$, where $l_e = 1, 2, 3, 4$; $-\arctan(a_{s_1}/a_{c_1}) = 0$ and $-\arctan(a_{s_2}/a_{c_2}) = \pi/4$. We tested the detection performance for SNR $= 3$ dB and SNR $= 10$ dB. The signal-to-noise ratio (SNR) was defined as $10\log_{10}\{(a_{s_1}^2 + a_{c_1}^2)/2\sigma^2\}$, and both sinusoids had the same SNR.

In our simulations we assumed $K = 5$. For the PMC algorithm we generated 3000 samples in each iteration, and we ran the algorithm for a total of ten iterations. As a prior for the hyperparameter δ^2, we used $\mathcal{IG}(2, 10)$. We employed three 5×5 matrices as transition kernels for model order: $Q_1(i,i) = 0.4$, $Q_1(i,j) = 0.15$, *for* $i \neq j$; $Q_2(i,i) = 0.80$, $Q_2(i,j) = 0.05$, *for* $i \neq j$; $Q_3(i,i) = 0.92$, $Q_3(i,j) = 0.02$, *for* $i \neq j$. As discussed at the end of Section III-B, by using these transition matrices, at each iteration t of the algorithm, a majority of the model order samples $\{k_{i,j}\}_{j=1}^N$ represented the models with large total weights in the previous iteration $t-1$, while we still allocated a small portion of the samples to represent those models that have small, even negligible, total weights. The rationale for the above choice of transition kernel is as follows. Even if after resampling at iteration $t-1$ all the samples represented model k, the other models would not become extinct at t. At iteration $t = 0$, instead of sampling uniformly on $[0, 0.5]^k$, which is the prior for frequency, we chose to sample from an IF $g^{(0)}(\mathbf{f}_k)$ to make sure that the samples reach the region of interest quickly. The function $g^{(0)}(\mathbf{f}_k)$ is defined by $g^{(0)}(\mathbf{f}_k) = \mathcal{N}(\widehat{\mathbf{f}}_k, \mathbf{C}_k^{(0)})$, where $\widehat{\mathbf{f}}_k$ is a $k \times 1$ vector whose values are the frequencies of the largest k peaks of the periodogram of the data, and $\mathbf{C}_k^{(0)}$ is a $k \times k$ diagonal matrix whose diagonal elements, say $\mathbf{C}_k^{(0)}[i,i]$ represent a quarter of the width of the peak of the periodogram located at frequency $\widehat{\mathbf{f}}_k[i]$. We chose the following form of time-varying transition kernel $Q_{\theta_{k(i,t)}}(\cdot, \boldsymbol{\theta}) = \mathcal{N}(\hat{\boldsymbol{\theta}}_{k^{(i,t-1)}}, \mathbf{C}_k^{(t)})$, and let $\mathbf{C}_k^{(1)} = \mathbf{C}_k^{(0)}$. When $t > 1$, if $k = \arg\max\sum_{i=1}^N w^{(i,t-1)}\Pi_k(k^{(i,t-1)})$, we have $\mathbf{C}_k^{(t)} = ((t-1)/t)\mathbf{C}_k^{(t-1)}$; if not, then $\mathbf{C}_k^{(t)} = ((t-1)/t)\mathbf{C}_k^{(t-1)} + (1/t)\mathbf{C}_k^{(1)}$. Of course, other choices of the transition kernels are possible, but we found that our choice resulted in good performance.

The averaged estimates of marginal distribution of model order k =1, 2, 3 when SNR $= 3$ dB and $l_e = 3$ is shown in Fig. 2, from which we observe that the estimates stabilized fairly quickly. Note, that the estimates of other models were all zeros and we did not show them in this figure. We also show in Fig. 3 the averaged estimates of the frequencies along with the true values of the frequencies. It is also clear that the estimates converge fast.

We also compared the sampling efficiencies of different choices of kernels. We first proposed a set of alternative kernels for model order that *does not* satisfy the requirements we suggested in Section III-B. The alternative kernels were defined as follows: $\bar{Q}_1(i,4) = 0.4$, $\bar{Q}_1(i,j) = 0.15$, *for* $j \neq 4$; $\bar{Q}_2(i,4) = 0.80$, $\bar{Q}_2(i,j) = 0.05$, *for* $j \neq 4$; $\bar{Q}_3(i,4) = 0.92$, $\bar{Q}_3(i,j) = 0.02$, *for* $j \neq 4$. It is clear that none of these kernels obey our first requirement, i.e., dominant diagonal entries. In Fig. 4, we calculated the averaged (over 100 realizations) entropy with respect to uniformity $H^{(t)}$ defined in (9) of three PMC implementations: 1) using $Q_d(\cdot,\cdot)$ for model order k and time-varying kernel for parameters $\boldsymbol{\theta}$; 2) using $Q_d(\cdot,\cdot)$ for model order k and time-invariant kernel for parameters $\boldsymbol{\theta}$; 3) using the alternative kernel $\bar{Q}_d(\cdot,\cdot)$ for model order k

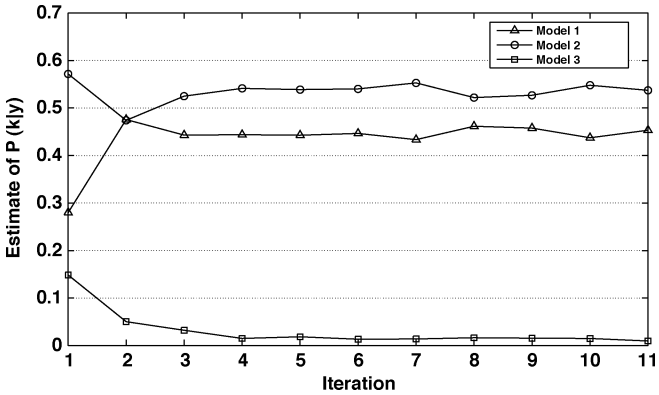

Fig. 2. Estimated $p(k|y)$ versus number of iterations.

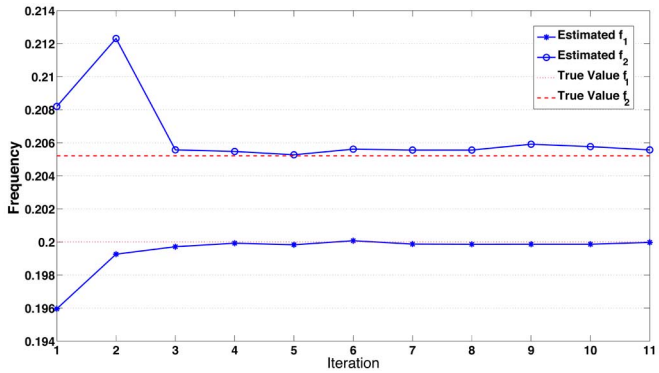

Fig. 3. Estimated f_1, f_2 versus number of iterations.

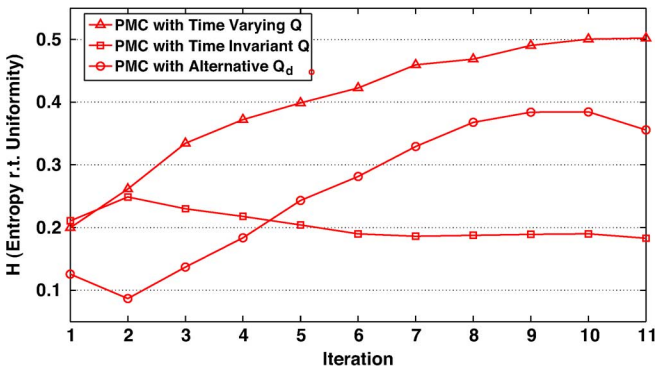

Fig. 4. Efficiency versus iteration of various PMC implementations.

and time-varying kernel for parameters $\boldsymbol{\theta}$. We can see that by employing the time-varying kernel, the sampling efficiency increased steadily with each iteration. We can also see that the implementation with the alternative kernels $\bar{Q}_d(\cdot,\cdot)$ has lower efficiency than the use of the kernels $Q_d(\cdot,\cdot)$.

We implemented the RJMCMC algorithm in the following way. The algorithm was run for 30 000 iterations with 5000 samples as burn-in period. We also used two different proposals for updating the frequency, one for local exploration and one for exploration of the "region of importance," see [8, Sec. IV-A]. Note that we chose the total number of samples generated by RJMCMC (30 000) to be large enough so that the estimated posterior can be stabilized. Also note, that this number was the same as the total number of samples used by the PMC (3000×10) [16].

We compared these two algorithms under scenarios where the SNR $= 3$ dB, 10 dB and the spacing parameter $l_e = 1, 2, 4$. For each of the different scenarios, both algorithms were run for 100 realizations, and the comparison of the detection performance is shown in Table II. The entries in the table are the number of times a particular model k was chosen out of 100 realizations. It can be seen that the performance of the two algorithms was comparable. Note that we also present the performance of the PMC algorithm that employed only one transition kernel, and the algorithm with alternative kernels $\bar{Q}_d(\cdot,\cdot)$ defined above. We observed that for these two choices of kernels, the performance degraded when the sinusoids were closely spaced and the SNR was low. The last result supports our claim that using multiple kernels for model order is beneficial, and it also supports our heuristics of intelligently choosing these predetermined kernels (as discussed in the end of Section III-B).

TABLE II
COMPARISON OF DETECTION PERFORMANCE

Algorithm	SNR	l	$k \leq 1$	$k = 2$	$k \geq 3$
PMC (multiple kernel)	3 dB	1	0	**100**	0
		2	1	**99**	0
		4	77	**23**	0
PMC (single kernel)	3 dB	1	2	**98**	0
		2	9	**90**	1
		4	89	**9**	2
PMC (alternative $\bar{Q}_d$)	3 dB	1	0	**98**	2
		2	1	**92**	7
		4	89	**11**	0
PMC (multiple kernel)	10 dB	1	0	**100**	0
		2	0	**100**	0
		4	0	**100**	0
PMC (single kernel)	10 dB	1	1	**99**	0
		2	2	**98**	0
		4	5	**95**	0
PMC (alternative $\bar{Q}_d$)	10 dB	1	0	**99**	1
		2	0	**96**	4
		4	0	**96**	4
RJMCMC	3dB	1	0	**99**	1
		2	1	**95**	4
		4	81	**19**	0
RJMCMC	10dB	1	0	**100**	0
		2	0	**98**	2
		4	1	**99**	0

We also observe that when the total number of samples was relatively small, the estimation performance of the RJMCMC deteriorated. For example, when the total number of samples was set to 10 000, and 2000 samples were used for the burn-in period, and we used 2000 samples per iteration with five iterations for PMC, the PMC outperformed the RJMCMC for low SNRs. Fig. 5 shows the estimation performance of the two algorithms under the above settings and when $l_e = 3$. It is clear that when the SNR is below 6 dB, the PMC performs better than the RJMCMC. In this figure we also plotted the CRLB [27].

B. Numerical Results for DOA

We used the following setup for the model: $d_y = 30$, $k = 2$, $\phi_1 = 20°$, $\phi_1 = 45°$, the number of sensors was $M = 5$ and the amplitudes of the signals were fixed at $a_1 = 10$, $a_2 = 10$. In order to generate a spatially colored noise, we used a second-order autoregressive process with poles $0.9\exp\{-j1.05\pi\}$ and

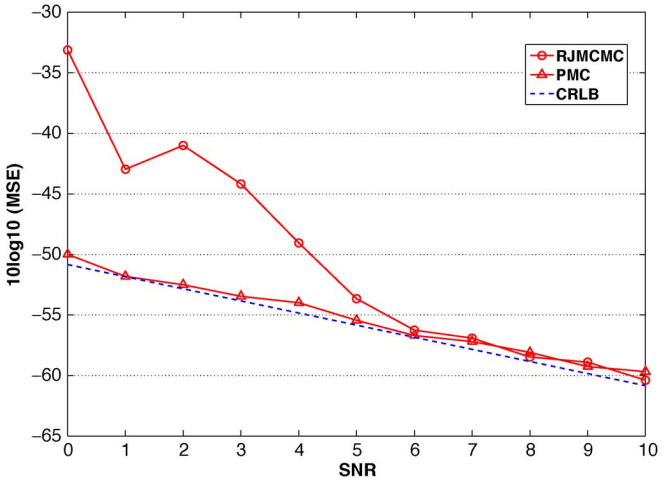

Fig. 5. MSE versus SNR of RJMCMC and PMC and comparison with the CRLB.

$0.9\exp\{-j0.90\pi\}$ whose driving noise was a circularly complex white Gaussian process with identical σ^2. The SNR was defined by $10\log 10(a_1^2/2\sigma^2)$. The hyperparameter ρ^2 was determined according to the criteria developed in Section V of [9] for different SNRs. As mentioned in [9], this setup was very difficult because the two sources were within a beamwidth of the receiver array, and the data size was very small.

For the PMC algorithm we used 1000 samples in each iteration, and we ran the algorithm for a total of four iterations. We used three 5×5 matrices as transition kernels for the model order, which were identical to those used in the previous experiment. At iteration $t = 0$, we sampled the model order and the DOAs uniformly. Note that in this situation, we did not have a natural candidate for the initial IF, as in the previous example, so we sampled the parameters from their priors (which were $U[0, 2\pi]^k$). In the subsequent iterations, we chose to use the exact time-varying transition kernel as in the previous experiment, except that $\mathbf{C}_k^{(1)} = (0.1\pi)^2 \times \mathbf{I}_{k\times k}$. Note, that $\mathbf{C}_k^{(1)}$ was preselected, as suggested in [11], when no obvious candidate was available.

We ran the RJMCMC algorithm for two settings: 4000 iterations with 1000 samples used for the burn-in period, and 15 000 iterations with 5000 samples for burn-in. In the first setting, the total number of samples generated by the RJMCMC (4000) was the same as the total number of samples used by the PMC (1000×4). Notice that in this setting, our choice of total number of samples is significantly smaller than that used in [9], which corresponds to our second setting. The purpose here was to demonstrate the instability of the RJMCMC algorithm when the sample size was small. In both settings, we used two proposals for updating the DOA, one for local exploration and one for exploring the whole parameter space. See [9, Sec. IV-A] for details.

Both algorithms were run for 100 realizations, and the detection performance of the algorithms is compared in Table III. Again, each entry of the table represents the number of times a particular model is selected in 100 realizations. It can be seen that when the total number of samples was small and the SNR was low, the performance of the RJMCMC algorithm was worse

TABLE III
COMPARISON OF PERFORMANCE OF DETECTION

Algorithm	SNR	$k \leq 1$	$k = 2$	$k \geq 3$
PMC 4,000 samples	-1 dB	44	**56**	0
	0 dB	45	**55**	0
	1 dB	37	**63**	0
	2 dB	35	**65**	0
	3 dB	14	**82**	4
RJMCMC 4,000 samples	-1 dB	66	**34**	0
	0 dB	60	**40**	0
	1 dB	48	**52**	0
	2 dB	37	**63**	0
	3 dB	27	**83**	0
RJMCMC 15,000 samples	-1 dB	48	**52**	0
	0 dB	49	**51**	0
	1 dB	40	**60**	0
	2 dB	30	**69**	1
	3 dB	15	**85**	0

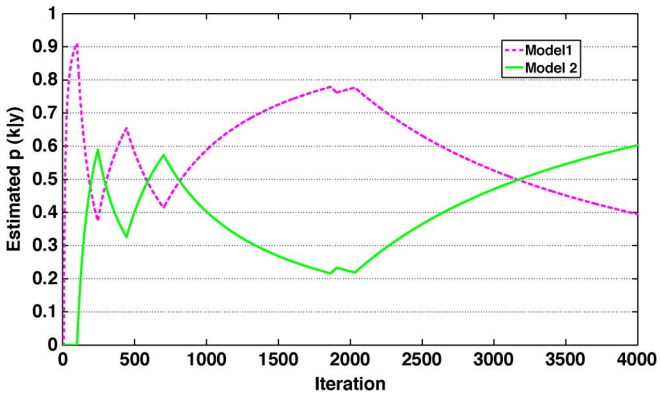

Fig. 6. Estimated $p(k|y)$ versus number of iterations obtained by the RJMCMC algorithm.

than that of the PMC algorithm. In Fig. 6, we see a typical realization of the estimates of $p(k = 1|y)$ and $p(k = 2|y)$ produced by the RJMCMC algorithm with 4000 samples and for SNR $= -1$ dB. It is clear that the estimated marginalized posterior was not stabilized within the window of 4000 samples, and the estimates based on these samples are not satisfactory. In Fig. 7, we see the averaged estimates of $p(k = 1|y)$ and $p(k = 2|y)$ for the PMC algorithm. Since the estimates of the other models were all zeros, we did not show them in the figure. It is clear that although there are only four iterations of the algorithm, the estimates improved with each iteration and they stabilized quickly.

C. Discussion

In the previous simulations we demonstrated the performance of the PMC by comparing it with that of the RJMCMC. We showed that the PMC had comparable performance with RJMCMC when the simulated sample size was large, and that the PMC outperformed the RJMCMC when the sample size was small. In both [8] and [9], the burn-in periods were determined in a heuristic fashion. We speculate that the choice of the updating proposals and the starting point of the algorithm heavily influenced the length of the burn-in period.

Another advantage of the PMC over the RJMCMC, as mentioned in Section I, is the potential for its parallel implementation. Although a thorough investigation of this topic is beyond

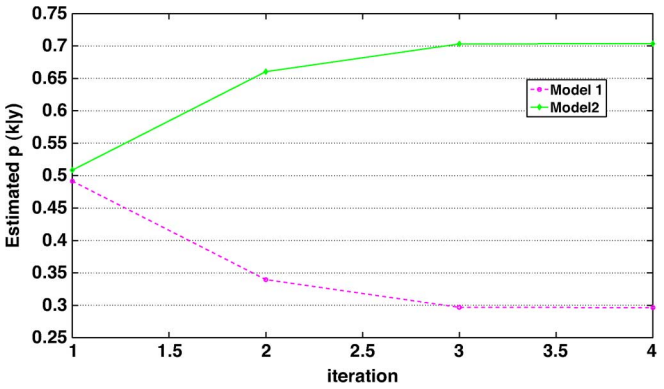

Fig. 7. Estimated $p(k|y)$ versus number of iterations obtained by the PMC algorithm.

the scope of this paper, we could easily identify several similarities between PMC and particle filtering, whose parallel implementation has been studied (see [28] for a recent development of this topic). We argue that RJMCMC is not suitable for parallel implementation mainly because each iteration of the algorithm produces a single sample and this sample is dependent on the sample produced in the previous iteration.

VI. Conclusion

We have proposed a general algorithm to carry out the Bayesian computation required for selecting the MAP model order. We have studied the convergence result of the proposed algorithm and demonstrated its performance on two typical signal processing problems. Indeed the proposed algorithm is flexible enough to approximate the posterior distribution for both problems, and it is computationally more efficient than the popular RJMCMC algorithm.

One future direction of research is to investigate theoretically the convergence rate of the proposed algorithm, which could shed new light to its behavior. It would also facilitate researchers in proposing new structures for the transition kernels for improving its rate of convergence. We also believe that a thorough study of the parallel implementation of the PMC algorithm would be of great practical interest.

Appendix A
Proof of Theorem 1

The proof is an adaptation of the proofs for PMC algorithm in [15]. First note that we have the following equations for importance sampling:

$$\begin{aligned} E\left(f(x)\right) &= \frac{\int f(x)w(x)q(x)dx}{\int w(x)q(x)dx} \\ &\approx \frac{\frac{1}{N}\sum_{i=1}^{N}\tilde{w}^{(i)}f\left(x^{(i)}\right)}{\frac{1}{N}\sum_{i=1}^{N}\tilde{w}^{(i)}} \\ &= \sum_{i=1}^{N} w^{(i)} f\left(x^{(i)}\right) \end{aligned} \tag{40}$$

where $q(x)$ is the sampling distribution and $\tilde{w}^{(i)}$ is unnormalized weight. In the following, we will set out to prove that

$$\frac{1}{N}\sum_{i=1}^{N}\tilde{w}^{(i,t)}f\left(\mathbf{x}^{(i,t)}\right) \overset{N\to\infty}{\longrightarrow} \int f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x} \tag{41}$$

which is the numerator of (40), and the convergence of the normalization constant will be automatically established when we plug in $f(x) = 1$ ($(1/N)\sum_{i=1}^{N}\tilde{w}^{(i,t)} \overset{N\to\infty}{\longrightarrow} 1$) [15], and (18) would follow.

We then state a Lemma. The proof of this lemma can be found in [29]. Let "$\to_p$" denote convergence in probability.

Lemma 1: Let (Ω, A) be a measurable space. Assume that the following conditions hold:

1) A sequence $\{\xi^{(i)}\}_{i=1}^{N}$ is independent given G_N, where G_N is a σ algebra in A.
2) $\{\sum_{i=1}^{N} E[\xi^{(i)}|G_N]\}$ is bounded in probability, e.g.,
$$\lim_{a\to\infty}\sup_{N>1} p\left(\sum_{i=1}^{N} E\left[\xi^{(i)}|G_N\right] \geq a\right) = 0. \tag{42}$$
3) $\forall b > 0$, $\sum_{i=1}^{N} E[\xi^{(i)}\Pi\{|\xi^{(i)}| > b\}|G_N] \to_p 0$.

Then,

$$\sum_{i=1}^{N}\left(\xi^{(i)} - E\left(\xi^{(i)}|G_N\right)\right) \to_p 0. \tag{43}$$

We verify the conditions on Lemma 1 to prove the convergence of the PMC algorithm.

We first state our assumptions. The function $f(x)$ is absolutely integrable under $p(x|y)$, which is to say

$$\int |f(x)|\, p(xy)dx < \infty. \tag{44}$$

Also the kernels $Q_d(\cdot,\cdot)$ and $Q_{\theta_k}(\cdot,\cdot)$ do not take value of 0 (or that $1/Q_d(\cdot,\cdot)$ and $1/Q_{\theta_k}(\cdot,\cdot)$ are both finite).

Let

$$\begin{aligned} \xi^{(i)} &= \frac{1}{N}\tilde{w}^{(i,t)}f\left(\mathbf{x}^{(i,t)}\right) \\ &= \frac{1}{N}\frac{p\left(\mathbf{x}^{(i,t)}|\mathbf{y}\right)f\left(\mathbf{x}^{(i,t)}\right)}{Q_{d^{(i,t)}}\left(k^{*(i,t-1)},k^{(i,t)}\right)Q_{\theta_{k^{(i,t)}}}\left(\boldsymbol{\theta}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}\right)} \end{aligned} \tag{45}$$

where $\mathbf{x}^{(i,t)}$ is the vector $[k^{(i,t)}\ \boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)^\top}]$. We let $G_N = \sigma\{\{\mathbf{x}^{*(i,t-1)}\}_{i=1}^{N}, \{\alpha_d^{(t)}\}_{d=1}^{D}\}$, which is, essentially, a σ algebra generated by the sequence $\{\mathbf{x}^{*(i,t-1)}\}_{i=1}^{N}$ and $\{\alpha_d^{(t)}\}_{d=1}^{D}$.

Notice that in the proposed algorithm the step $t = 0$, is just conventional importance sampling, so the proof for this case comes from the law of large numbers of importance sampling. Based on this observation, we check all the three conditions in Lemma 1, for $t \geq 1$. We use induction and that the convergence in (18) is established for $t-1$.

1) At iteration t, the $k^{(i,t)}$'s are i.i.d. and drawn from $Q_{d^{(i,t)}}(k^{*(i,t-1)},k^{(i,t)})$, and the $d^{(i,t)}$'s are i.i.d. and

generated from $\mathcal{M}(\alpha_1^{(t)},\ldots,\alpha_D^{(t)})$. Thus, we have that $\{k^{(i,t)}\}_{i=1}^N$ are independent from each other conditionally on $k^{*(i,t-1)}$ and $\alpha_d^{(t)}$. We also have that each $\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}$ is drawn from $Q_{\theta_{k^{(i,t)}}}(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)})$. Then, the independence of $\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}$ conditional on $\{\mathbf{x}^{*(i,t-1)}\}_{i=1}^N$ comes from the independence of $k^{(i,t)}$ (notice, that $\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)}$ is the estimate of $\boldsymbol{\theta}$ for the model order $k^{(i,t)}$ and is obtained from $\{\mathbf{x}^{(i,t-1)}\}_{i=1}^N$ at iteration $t-1$).

2) We have

$$\begin{aligned}
E\left[\xi^{(i)}|G_N\right]
&= E\left[\frac{1}{N}\tilde{w}^{(i,t)} f\left(\mathbf{x}^{(i,t)}\right)|G_N\right] \\
&= \frac{1}{N}\int \frac{p\left(\mathbf{x}^{(i,t)}|\mathbf{y}\right) f\left(\mathbf{x}^{(i,t)}\right)}{Q_{d^{(i,t)}}\left(k^{*(i,t-1)},k^{(i,t)}\right) Q_{\theta_{k^{(i,t)}}}\left(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}\right)} \\
&\qquad \times Q_{d^{(i,t)}}\left(k^{*(i,t-1)},k^{(i,t)}\right) \\
&\qquad \times Q_{\theta_{k^{(i,t)}}}\left(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}\right) d\mathbf{x}^{(i,t)} \\
&= \frac{1}{N}\int p\left(\mathbf{x}^{(i,t)}|\mathbf{y}\right) f\left(\mathbf{x}^{(i,t)}\right) d\mathbf{x}^{(i,t)}. \qquad (46)
\end{aligned}$$

The integral in the above result is bounded for all i and we get

$$\sum_{i=1}^N E\left[\xi^{(i)}|G_N\right] = \frac{1}{N}\sum_{i=1}^N \int p\left(\mathbf{x}^{(i,t)}|\mathbf{y}\right) f\left(\mathbf{x}^{(i,t)}\right) d\mathbf{x}^{(i,t)} < \infty.$$

The last inequality comes from the assumption that $f(\cdot)$ is absolutely integrable with respect to (w.r.t.) $p(\mathbf{x}|\mathbf{y})$.

3) From the condition 3) of the lemma, we have the following, shown in the first equation at the bottom of the page. Notice, that the expectation is taken on the random variable $d^{(i,t)}$ and $\mathbf{x}^{(i,t)}$, respectively. Since

$$E_{x,d}\left(f(x,d)\right) = E_d(E_{x|d}\left(f(x,d)\right) \qquad (47)$$

the aforementioned long equation becomes, as in (46), shown in the second equation at the bottom of the page.

We simplify the equation by denoting $p(\mathbf{x}^{(i,t)}|\mathbf{y})f(\mathbf{x}^{(i,t)})$ as $F(\cdot)$ and $(1/N)(p(k^{(i,t)},\theta_{k^{(i,t)}}^{(i,t)}|\mathbf{y})$ $f(\mathbf{x}^{(i,t)})/Q_d(k^{*(i,t-1)},k^{(i,t)})Q_{\theta_{k^{(i,t)}}}(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}))$ as $M(\cdot)$. Notice that $F(\cdot)$ is a function of $\mathbf{x}^{(i,t)}$, and $M(\cdot)$ is a function of $\mathbf{x}^{(i,t)}$, d, $k^{*(i,t-1)}$, and $\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)}$, respectively. Then the above equation becomes

$$\frac{1}{N}\sum_{i=1}^N\sum_{d=1}^N \alpha_d^{(t)} \int F(\cdot)\Pi\left\{|M(\cdot)|>\delta\right\} d\mathbf{x}^{(i,t)}. \qquad (48)$$

Notice that $\int F(\cdot)\Pi\{|M(\cdot)| > \delta\}d\mathbf{x}^{(i,t)}$ is a function of $k^{*(i,t-1)}$, d, and $\{\widehat{\boldsymbol{\theta}}_k^{(t-1)}\}_{k=1}^K$. Evidently, we have the following inequality:

$$\begin{aligned}
&\int F(\cdot)\Pi\left\{|M(\cdot)|>\delta\right\} d\mathbf{x}^{(i,t)} \\
&\le \int |F(\cdot)|\,\Pi\left\{|M(\cdot)|>\delta\right\} d\mathbf{x}^{(i,t)} \\
&\le \int |F(\cdot)|\, d\mathbf{x}^{(i,t)} = \int \left|p\left(\mathbf{x}^{(i,t)}|\mathbf{y}\right) f\left(\mathbf{x}^{(i,t)}\right)\right| d\mathbf{x}^{(i,t)} \\
&< \infty. \qquad (49)
\end{aligned}$$

$$\begin{aligned}
&\sum_{i=1}^N E\left[\xi^{(i)}\Pi\left\{\left|\xi^{(i)}\right|>\delta\right\}|G_N\right] \\
&= \frac{1}{N}\sum_{i=1}^N E\left(\tilde{w}^{(i,t)} f\left(\mathbf{x}^{(i,t)}\right)\Pi\left\{\left|\frac{1}{N}\tilde{w}^{(i,t)} f\left(\mathbf{x}^{(i,t)}\right)\right|>\delta\right\}|G_N\right) \\
&= \frac{1}{N}\sum_{i=1}^N E\left(\frac{p\left(\mathbf{x}^{(i,t)}|\mathbf{y}\right) f\left(\mathbf{x}^{(i,t)}\right)}{Q_{d^{(i,t)}}\left(k^{*(i,t-1)},k^{(i,t)}\right) Q_{\theta_{k^{(i,t)}}}\left(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}\right)}\right. \\
&\qquad \left.\times\,\Pi\left\{\frac{1}{N}\left|\frac{p\left(\mathbf{x}^{(i,t)}|\mathbf{y}\right) f\left(\mathbf{x}^{(i,t)}\right)}{Q_{d^{(i,t)}}\left(k^{*(i,t-1)},k^{(i,t)}\right) Q_{\theta_{k^{(i,t)}}}\left(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}\right)}\right|>\delta\right\}|G_N\right)
\end{aligned}$$

$$\frac{1}{N}\sum_{i=1}^N\sum_{d=1}^N \alpha_d^{(t)} \int p\left(k^{(i,t)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}|\mathbf{y}\right) f\left(\mathbf{x}^{(i,t)}\right)\Pi\left\{\left|\frac{1}{N}\frac{p\left(k^{(i,t)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}|\mathbf{y}\right) f\left(\mathbf{x}^{(i,t)}\right)}{Q_{d^{i,t}}\left(k^{*(i,t-1)},k^{(i,t)}\right) Q_{\theta_{k^{(i,t)}}}\left(\widehat{\boldsymbol{\theta}}_{k^{(i,t)}}^{(t-1)},\boldsymbol{\theta}_{k^{(i,t)}}^{(i,t)}\right)}\right|>\delta\right\} d\mathbf{x}^{(i,t)}$$

Consequently, we have that $\int F(\cdot)\Pi\{|M(\cdot)| > \delta\}d\mathbf{x}^{(i,t)}$ is a function of $k^{*(i-1,t)}$, and it takes a finite value (it is also, as mentioned above, a function of d and $\{\widehat{\boldsymbol{\theta}}_k^{(t-1)}\}_{k=1}^K$). Evidently, the above function is integrable under $p(k|\mathbf{y})$. Let us denote the integral $\int F(\cdot)\Pi\{|M(\cdot)| > \delta\}d\mathbf{x}^{(i,t)}$ by $h(k^{*(i-1,t)})$. Recall that $k^{*(i,t-1)}$ is a resampled particle, so the weights are normalized. By induction on (18), we have

$$\begin{aligned}
&\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\int F(\cdot)\Pi\left\{|M(\cdot)|>\delta\right\}d\mathbf{x}^{(i,t)}\\
&=\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}h\left(\tilde{k}^{(i-1,t)}\right)\\
&\to_p \lim_{N\to\infty}\int h(k)p(k|\mathbf{y})dk\\
&=\lim_{N\to\infty}\sum_{k=1}^{K}h(k)p(k|\mathbf{y})\\
&=\lim_{N\to\infty}\sum_{k=1}^{K}\int F(\cdot)\Pi\left\{|M(\cdot)|>\delta\right\}d\mathbf{x}^{(i,t)}p(k|\mathbf{y})\\
&\le\lim_{N\to\infty}\int|F(\cdot)|\sum_{k=1}^{K}\Pi\left\{|M(\cdot)|>\delta\right\}p(k|\mathbf{y})d\mathbf{x}^{(i,t)}\\
&=\lim_{N\to\infty}\int_{E_1}|F(\cdot)|\sum_{k=1}^{K}\Pi\left\{|M(\cdot)|>\delta\right\}p(k|\mathbf{y})d\mathbf{x}^{(i,t)}\\
&\quad+\int_{E_2}|F(\cdot)|\sum_{k=1}^{K}\Pi\left\{|M(\cdot)|>\delta\right\}p(k|\mathbf{y})d\mathbf{x}^{(i,t)} \qquad (50)
\end{aligned}$$

where $E_1 \triangleq \{\mathbf{x}^{(i,t)} : |f(\mathbf{x}^{(i,t)})| < \infty\}$, and $E_2 \triangleq \{\mathbf{x}^{(i,t)} : |f(\mathbf{x}^{(i,t)})| = \infty\}$ and we have $E_1 \cap E_2 = \emptyset$. As N goes to infinity, we can show that both integrals go to 0. We omit the detail of this claim due to space limit.

Finally, we have the following convergence:

$$\frac{1}{N}\sum_{i=1}^{N}\int F(\cdot)\Pi\left\{|M(\cdot)|>\delta\right\}d\mathbf{x}^{(i,t)}\to_p 0 \qquad (51)$$

which implies

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{d=1}^{D}\alpha_d^{(t)}\int F(\cdot)\Pi\left\{|M(\cdot)|>\delta\right\}d\mathbf{x}^{(i,t)}\to_p 0 \qquad (52)$$

because $\alpha_d^{(t)} <= 1$ and $\sum_{d=1}^{D}\alpha_d^{(t)} = 1$.

Thus, we verified the third condition, and the conclusion follows, namely,

$$\frac{1}{N}\sum_{i=1}^{N}\tilde{w}^{(i,t)}f\left(\mathbf{x}^{(i,t)}\right)\overset{N\to\infty}{\longrightarrow}_p\int f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x}$$

■

REFERENCES

[1] P. Stoica and Y. Selen, "Model order selection: A review of information criterion rules," *IEEE Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, Jul. 2004.

[2] P. M. Djurić, "Simultaneous detection and frequency estimation of sinusoidal signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Minneapolis, MN, 1993, pp. 53–56.

[3] P. M. Djurić, "A model selection rule for sinusoids in white Gaussian noise," *IEEE Trans. Signal Process.*, vol. 44, no. 7, pp. 1744–1751, Jul. 1996.

[4] P. M. Djurić, "Asymptotic map criteria for model selection," *IEEE Trans. Signal Process.*, vol. 46, no. 10, pp. 2726–2735, Oct. 1998.

[5] C.-M. Cho and P. M. Djurić, "Bayesian detection and estimation of cisoids in colored noise," *IEEE Trans. Signal Process.*, vol. 43, no. 12, pp. 2943–2951, Dec. 1996.

[6] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 4, pp. 711–732, 1995.

[7] C. Andrieu, P. M. Djurić, and A. Doucet, "Model selection by Markov chain Monte Carlo computations," *Signal Process.*, vol. 81, pp. 19–37, 2001.

[8] C. Andrieu and A. Doucet, "Joint Bayesian model selection and estimation of noise sinusoids via reversible jump MCMC," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2667–2676, Oct. 1999.

[9] J. R. Larocque and J. P. Reilly, "Reversible jump MCMC for joint detection and estimation of sources in colored noise," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 231–240, Feb. 2002.

[10] R. Iba, "Population-based Monte Carlo algorithms," *J. Comput. Graphical Statist.*, vol. 7, pp. 175–193, 2000.

[11] O. Cappé, J. M. Marin, and C. P. Robert, "Population Monte Carlo," *J. Comput. Graph. Statist.*, vol. 13, no. 4, pp. 907–929, 2003.

[12] D. Rubin, "Comment on 'The calculation of posterior distributions by data augmentation' by M. A. Tanner and W. H. J. Wong," *J. Amer. Statist. Assoc.*, vol. 82, p. 543, 1987.

[13] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer, 2005.

[14] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert, "Minimum variance importance sampling via population Monte Carlo," *ESAIM: Probab. Statist.*, vol. 11, pp. 427–447, 2007.

[15] R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert, "Convergence of adaptive mixtures of importance sampling schemes," *Ann. Statist.*, vol. 35, 2007.

[16] G. Celeux, J.-M. Marin, and C. P. Robert, "Iterated importance sampling in missing data problems," *Comput. Statist. Data Anal.*, vol. 50, pp. 3386–3404, 2006.

[17] M. F. Bugallo, M. Hong, and P. M. Djurić, "Marginalized population Monte Carlo," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Taipei, Taiwan, 2009.

[18] *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York: Springer, 2001.

[19] M. Shafi, P. J. Smith, and H. Gao, "Quick simulation: A review of importance sampling techniques in communications systems," *J. Sel. Areas Commun.*, vol. 15, pp. 397–412, 1997.

[20] M. West, "Approximating posterior distribution by mixtures," *J. R. Statist. Soc. B*, vol. 55, no. 2, pp. 409–422, 1993.

[21] O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert, "Population Monte Carlo for ion channel restoration," *J. Comput. Graph. Statist.*, vol. 13, pp. 907–929, 2004.

[22] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Process. Mag.*, vol. 20, no. 5, pp. 19–38, Sep. 2003.

[23] D. B. Rubin, "Using the SIR algorithm to simulate posterior distributions," in *Bayesian Statistics 3*. New York: Oxford Univ. Press, 1988, pp. 395–402, (with discussion).

[24] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. New York: Springer, 2001.

[25] K. M. Wong, J. P. Reilly, Q. Wu, and S. Qiao, "Estimation of the directions of arrival of signals in unknown correlated noise, Part I: The map approach and its implementation," *IEEE Trans. Signal Process.*, vol. 40, no. 8, pp. 2007–2017, Aug. 1992.

[26] G. E. P. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*. New York: Wiley, 1973.

[27] S. M. Kay, *Modern Spectral Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[28] S. Hong and P. M. Djurić, "High-throughput scalable parallel resampling mechanism for effective redistribution of particles," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1144–1155, Mar. 2006.

[29] R. Douc and E. Moulines, "Limit theorems for properly weighted samples with applications to sequential Monte Carlo," TSI, Telecom, Paris, France, 2005, Tech. Rep.

Mingyi Hong (S'09) received the B.E. degree in communications engineering from Zhejiang University, Hangzhou, China, in 2005 and the M.S. degree in electrical engineering from Stony Brook University, Stony Brook, NY, in 2007. He is currently pursuing the Ph.D. degree in the Department of Systems and Information Engineering, University of Virginia, Charlottesville.

His research interests are primarily in the fields of statistical signal processing, wireless communications, and optimization theory. Currently, he is working in Monte Carlo methods with emphasis in estimation and detection theory. He is also doing research regarding dynamic spectrum allocation in the cognitive radio network.

Mónica F. Bugallo (M'04) received the Ph.D. degree in computer engineering from University of A Coruña, A Coruña, Spain, in 2001.

From 1998 to 2001, she was with the Department of Electronics and Systems, University of A Coruna, where she worked in interference cancellation applied to multiuser communication systems. In 2002, she joined the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, where she is currently an Assistant Professor and teaches courses in digital communications and information theory. Her research interests lie in the area of statistical signal processing and its applications to different disciplines including communications and biology.

Petar M. Djurić (M'90–SM'99–F'06) received the B.S. and M.S. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1981 and 1986, respectively, and the Ph.D. degree in electrical engineering from the University of Rhode Island, Kingston, in 1990.

From 1981 to 1986, he was a Research Associate with the Institute of Nuclear Sciences, Vinča, Belgrade. Since 1990, he has been with Stony Brook University, Stony Brook, NY, where he is Professor in the Department of Electrical and Computer Engineering. He works in the area of statistical signal processing, and his primary interests are in the theory of signal modeling, detection, and estimation and application of the theory to a wide range of disciplines.

Prof. Djurić has been invited to lecture at universities in the United States and overseas and has served on numerous committees for the IEEE. During 2008–2009, he was Distinguished Lecturer of the IEEE Signal Processing Society. He was the Area Editor for Special Issues of the *Signal Processing Magazine* and Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He has also been on the Editorial Boards of many IEEE and EURASIP professional journals. In 2007, he received the IEEE Signal Processing Magazine Best Paper Award.