Stony Brook University

POPULATION MONTE CARLO SAMPLING FOR HIGH DIMENSIONAL PROBLEMS

A Dissertation Presented

by

Bingxin Shen

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

August 2011

Stony Brook University

The Graduate School

Bingxin Shen

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Mónica F. Bugallo - Dissertation Advisor
Associate Professor, Department of Electrical and Computer Engineering

Petar M. Djurić - Chairperson of Defense
Professor, Department of Electrical and Computer Engineering

Sangjin Hong
Associate Professor, Department of Electrical and Computer Engineering

Wei Zhu
Professor, Department of Applied Mathematics & Statistics

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

POPULATION MONTE CARLO SAMPLING

FOR HIGH DIMENSIONAL PROBLEMS

by
Bingxin Shen
Doctor of Philosophy
in
Electrical Engineering
Stony Brook University
2011

Many real-world data analysis problems involve estimating unknowns and approximating their posterior distributions when only partial or inaccurate/noised observations are available. In most cases, the system models are non-linear and/or non-Gaussian with high dimensions, where the posterior distributions cannot be obtained analytically. For the last few decades, many approximation schemes have been proposed to solve this problem. One group of tools favored in theory and practice are Monte Carlo (MC)-based methods. Population Monte Carlo (PMC) is one of the methods of the MC family for batch processing of data.

PMC algorithms iterate on a set of samples and weights. The proposal distributions are updated at each iteration by learning from the performances of the previous proposal distributions compared to the target distribution. The target distribution is often the a posteriori distribution of a set of unknowns of interest given observed data and the employed model. The estimation quality and convergence efficiency rely on many factors including the number and "quality" of the generated samples.

In problems with a high dimensional state space, the PMC implementation is very challenging due to the necessity of large numbers of samples. In this thesis the focus is on researching and advancing the PMC methodology towards its adequate and robust performance in scenarios of high dimensional systems.

In some of these problems, some of the unknown parameters are conditionally linear given the remaining parameters. For those cases, marginalized PMC (MPMC) is proposed

to lower the computational cost by only generating samples of the nonlinear parameters and marginalizing the remaining linear parameters. This approach is based on the well-known Rao-Blackwell theorem.

The computational efficiency of the PMC method can be further improved by the use of a distributed structure. To that end, we propose a novel method referred to as multiple PMC where the state space of interest is partitioned into several subspaces with lower dimensions and handled by a set of parallel PMC estimators. Each PMC estimator updates the weights of the samples and the importance functions, if necessary, using information from the other PMC estimators.

As with every method that uses importance sampling, the crucial factor for good performance of the method for PMC is the choice of generating functions of the particles. We also propose an alternative method where the generating functions are *alternating conditionals*, thereby mimicking the idea behind Gibbs sampling. With this approach, one can generate particles in high dimensions more efficiently.

We test and demonstrate the proposed approaches on the classical problem of estimating the frequencies of multiple sinusoids. The simulation results show the accuracy of the estimates and the feasibility of the methods. The performances of the proposed methods are compared to that of other conventional approaches. Computational complexity and convergence properties are also studied.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AWGN	Additive White Gaussian Noise
CDF	Cumulative Density Function
CRLB	Cramér-Rao Lower Bound
CWGN	Complex White Gaussian Noise
EFF	Effective Sample Size
FFT	Fast Fourier Transformation
flop	floating-point operation
HMM	Hidden Markov Model
IF	Importance Function
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MLE	Maximum Likelihood Estimator
MLLFE	Mean LikeLihood Frequency Estimator
MSE	Mean Square Error
MMSE	Minimum Mean-Squre Estimator
MPMC	Marginalized Population Monte Carlo
MultiPMC	Multiple Population Monte Carlo
MultiMPMC	Multiple Marginalized Population Monte Carlo
NMR	nuclear magnetic resonance
PF	Particle Filter
PCRLB	posterior Cramér-Rao Lower Bound
PMC	Population Monte Carlo
RB	Rao-Blackwel
SMC	Sequential Monte Carlo
SNR	Signal-to-Noise Ratio

ACKNOWLEDGMENTS

This work has been supported by Electrical and Computer Engineering Department, Stony Brook University, National Science Foundation under CCF-0515246 and the Office of Naval Research under Award N00014-09-1-1154. I would like to thank them for their generous support.

I would like to thank my dearest parents Changxi and Xuezhen, and my husband Guozheng for their love, encouragement and support through my academic endeavor.

I would also like to thank my advisors Professor Mónica F. Bugallo and Professor Petar M. Djurić. They have been really great advisors for the past four years. Their enthusiasm for research and the vision for the future have been an inspiration. I would like to thank Professor Sangjin Hong, Professer Wei Zhu and Professor Nancy Mendell for their guidance, suggestions and advices during my graduate study and research work.

I am grateful to Dr. Takai Helio and the whole MARIACHI research group for the enthusiasm and support. Finally, I would like to thank my friends: Zejie, Mingyi, Ting, Yao, Yunlong, Shishir, Cagla, Li, Jinghui, Xiaoying, Shuo, Chunmei for the help and happy hours during these past four years. All of you have made Stony Brook a great memory for me.

CHAPTER 1

Introduction

1.1 Motivation

Modern science and engineering advancement involves dealing with problems related to high-dimensional systems. Image retrieval [69, 73], financial tick-by-tick data analysis [32, 41, 50], biological data processing [35, 40], or weather prediction [26, 29] are just few examples of high dimensional systems of interest (see Figure 1.1).

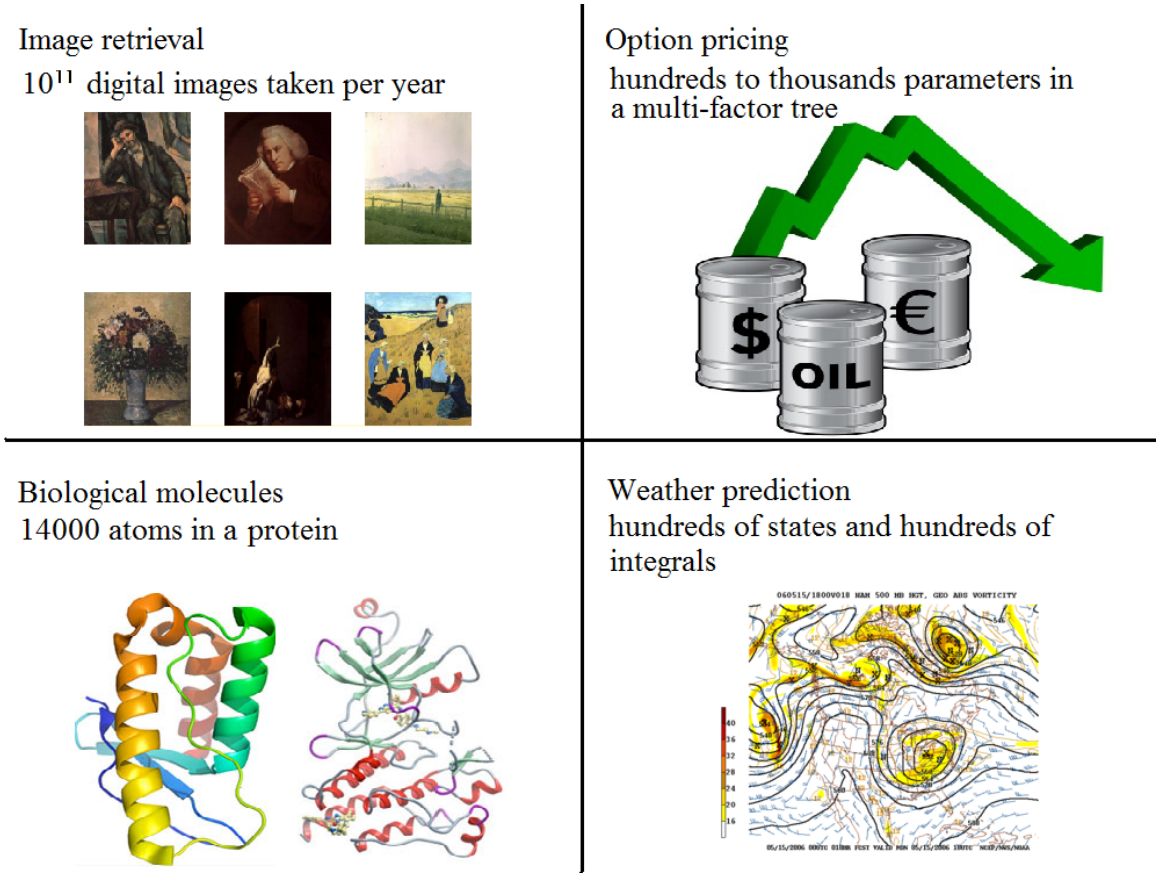


Figure 1.1: Examples of high dimensional problems.

Image retrieval deals with searching for images with certain content within a large digital database [69, 73]. It can be used for security such as face recognition, media retrieval application like Google's image search project, or medical treatment with nuclear magnetic resonance (NMR) images of a patient. Unlike text data, images form a much more complicated space. For instance 10^{11} digital images taken into Google's image database form an extremely large state space if no data compression or feature extraction are involved. Traditional processing procedures cannot handle such large database.

In finance, a typical multi-factor tree (usually the joint product of a number of binomial trees) may involve hundreds of parameters in valuing and analyzing complex instruments, portfolios and investments [32]. The traditional numerical approaches are usually infeasible or impossible to deal with high-dimensional integrals in option pricing problems [41, 50].

Data processing is widely used in the biological problems, for instance the development of molecular diagnostics. However, it is extremely hard to sample and process the information using traditional methods for large molecular systems, which can have up to $14,000$ atoms [35]. In dealing with biological molecule identification problems [40], one faces difficulties due to model mismatch and the nature of the biological sequence. The resulting problems represent inference in high dimensional, highly multimodal spaces,

Many modern geophysical problems, such as weather forecasting [26] are characterized by extremely high-dimensional systems, and they pose difficult challenges for assimilation of system information and observations. A typical weather prediction problem may deal with hundreds of states and involve computations of hundreds of integrals.

Processing and inferring information from the previously described systems involve additional challenges and difficulties that traditional signal processing techniques fail to meet. Therefore, new approaches need to be developed and studied.

1.1.1 Monte Carlo methods

Many real-world data analysis and signal processing problems involve the estimation of unknown quantities or parameters from some given observations or measurements [23, 28, 45, 64, 68]. An estimator attempts to approximate the unknown parameters using the observations and it is desirable to obtain estimators exhibiting optimality and minimum average error. However, in many situations it is infeasible or impossible to obtain such estimators, especially when dealing with high-dimensional systems.

A class of computational algorithms which are often used in simulating complex physical and mathematical systems is the Monte Carlo (MC) family [23, 34, 55, 56, 59, 62]. MC methods employ repeated random sampling to compute their results.They are especially useful in studying high-dimensional systems, characterized by significant uncertainty in their inputs, complicated boundary conditions, non-linearities and non-Gaussianities [23].

MC methods have been applied in many areas of computational mathematics and basically can be divided into two sub-groups: methods that operate in sequential mode and those which work in batch mode. The sequential Monte Carlo (SMC) methods [17, 23], also known as particle filters (PFs), deal with real time applications. Markov chain Monte Carlo (MCMC) and population Monte Carlo (PMC) belong to the group of batch methods.

In some applications, data estimation needs to be performed online or in real time. In these cases, observations/measurements are processed in a sequential manner. For data estimation in a linear Gaussian state-space model, it is possible to obtain an analytical solution, given by the well known and widely used Kalman filter [42, 72]. If the data are modeled as a partially observed finite Markov chain, it is also possible to derive an exact analytical expression to compute the posterior distribution, which is known as the hidden Markov model (HMM) filter [4, 61]. When the system is non-linear and/or non-Gaussian, one can obtain approximate solutions using grid-based filters [2, 15], or extended Kalman filters [47] or may be Gaussian sum approximations [2, 49]. However, these methods are extremly difficult to implement and are computationally too expensive for high dimensional systems. SMC methods constitute an

important alternative for estimating unknown parameters in such applications. SMC methods are flexible, easy to implement, parallelisable and applicable in very general settings [16,17,23].

Unlike online data processing that deals with interactive inputs/outputs, batch data processing collects input data into batches and processes them in an offline mode. Except in few special cases, including linear Gaussian state space models, it is infeasible to employ traditional methods to obtain an analytical estimation solution. One group of tools that has been favored includes batch mode MC methods such as the MCMC and PMC algorithms [1, 38, 62]. The research presented in this work focuses on high-dimensional batch mode data processing.

1.1.2 Challenges

Over the last few decades, research has been focused in data gathering and data processing methods. The information technology is growing fast and affecting more and more different aspects of society. Data processing, estimation and prediction methods play a vital role in the advancement of scientific research, medical investment, engineering development, and commercial endeavors.

Many real-world problems are described by high dimensional nonlinear and non-Gaussian complex models where standard computational methods are difficult to implement. MC batch methods suffer too in their application to this type of problems due to the necessity of generating very large number of samples as the number of unknowns (dimension) of the problem increases.

Figure 1.2 shows the CPU computation time on a personal computer of a PMC algorithm in an example of frequency estimation of sinusoidal signal in noise with increasing dimensionality of the vector of unknown parameters (i.e. the number of unknown frequencies). The number of iterations and the number of samples used per parameter were fixed. From the figure, it is clear that as the dimension of parameters to be estimated increases, the corresponding CPU time increases exponentially. In fact, the PMC algorithm needs more iterations to converge which further increases the computational cost and lengthens the convergence time. Figure 1.3 shows

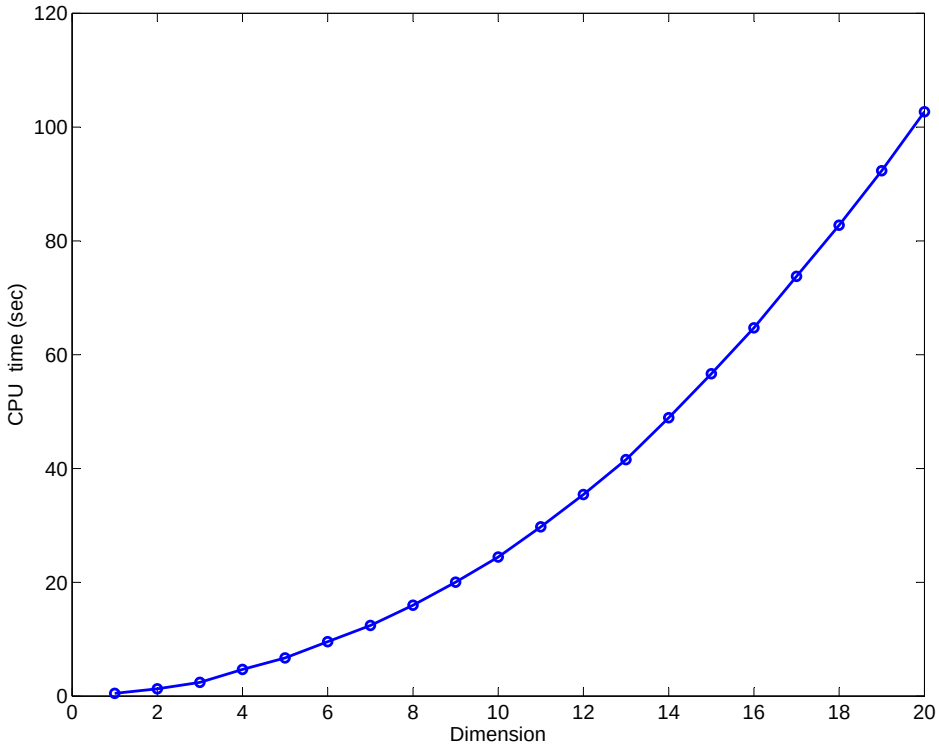


Figure 1.2: CPU time cost for a traditional PMC algorithm vs. dimensionality of the system.

the number of iterations required by the PMC method with increasing dimensionality of the vector of parameters. It can be observed that the PMC algorithm needs a very large amount of iterations to reach a similar level of accuracy as the dimension increases.

In high dimensional applications, the computational cost increases exponentially and the applicability of these methods becomes questionable. Therefore, dealing with problems related to high dimensional spaces is an important area of research.

1.2 Batch Monte Carlo methodologies

Batch data processing collects input data into batches and processes them without strict deadlines for completing the processing.

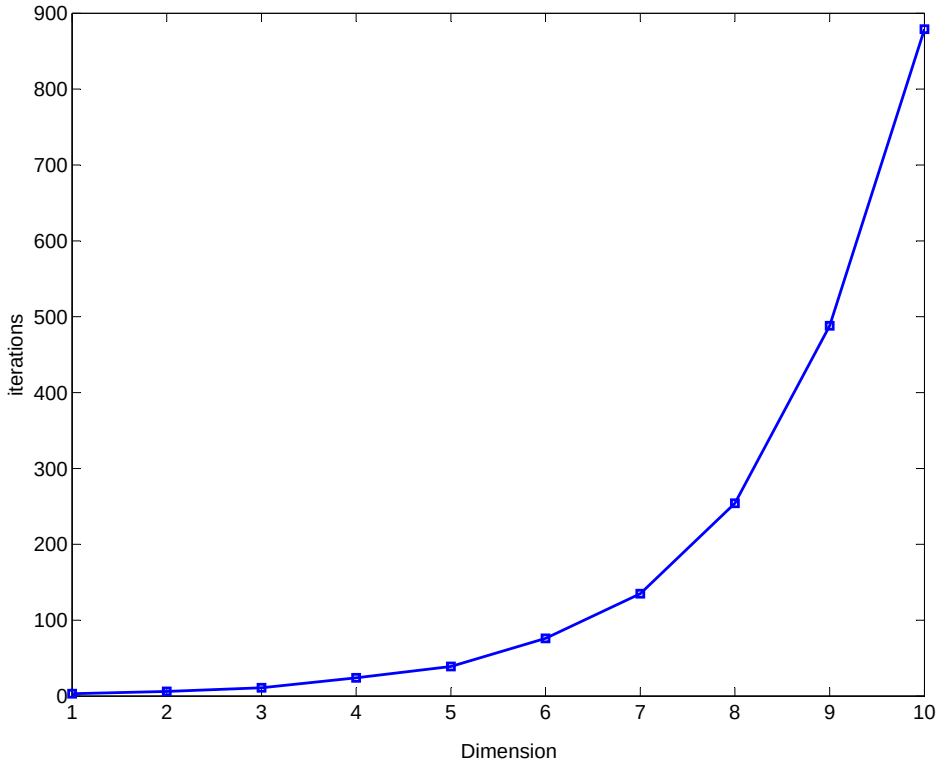


Figure 1.3: Number of iterations required by a traditional PMC method vs dimensionality of the system.

1.2.1 Markov-chain Monte Carlo vs. population Monte Carlo sampling

Within the MC methodology, MCMC sampling is a class of algorithms for sampling from probability distributions based on constructing a Markov chain towards the target distribution [1]. The states/samples follow a random walk in the state space and the likelihoods are computed to determine the acceptance or rejection of these new states/samples, shown as in Figure 1.4. The states/samples of the chains after a number of steps/iterations are used to construct an approximation of the target distribution. The most common application of MCMC algorithms is to numerically calculate multi-dimensional integrals. However, the main problems related to the application of MCMC algorithms are the choice of the unknown number of drawn samples to claim convergence [62]; the selection of appropriate initial conditions that

will lead the chain to converge to the distribution [54, 62]; and the parallel implementation of the algorithm [1].

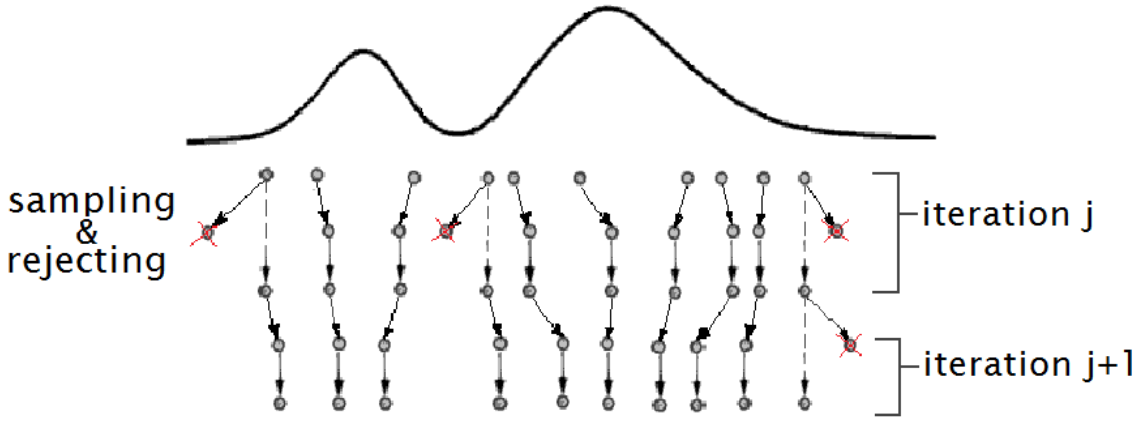


Figure 1.4: The MCMC method.

An alternative to MCMC sampling is the PMC algorithm, which does not need tests for convergence if samples are drawn from good proposal distributions. PMC methods are inspired from particle systems and were introduced to handle and approximate rapidly changing target distributions like those found in signal and image processing [23, 33]. The PMC methodology primarily handles fixed but complex target distributions by building a sequence of increasingly better proposal distributions.

The PMC algorithm produces a set of samples at each iteration that are used for the representation of vector of parameters. The iterative structure allows for adaptivity and convergence towards the target distribution. PMC algorithms have been applied to a number of fields, such as scientific computing [55], statistical physics [38], statistical sciences [62], and image rendering [51].

As shown in Figure 1.5, at each iteration the set of random samples are drawn from a family of proposal distributions, called importance functions (IFs) or proposal distributions. A weight for each sample is calculated according to the importance criterion. Finally resampling is carried out before moving to the next iteration to avoid degeneracy of the samples [63].

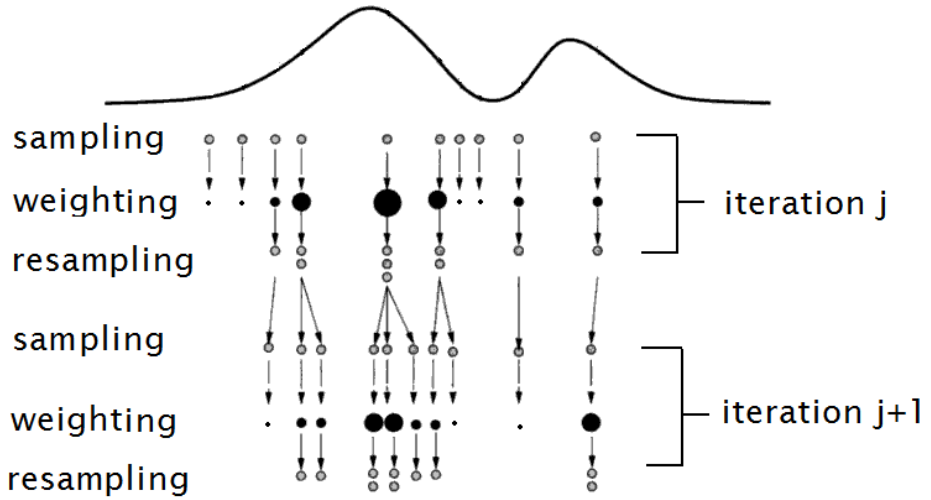


Figure 1.5: The PMC method.

At the end, the samples from all the iterations are used to form unbiased estimates of summations/integrals under that distribution. PMC sampling is an adaptive algorithm, since the proposal distribution is updated at each iteration by learning from the performance of the previous proposal distributions compared to the target one. The PMC algorithm is presented in detail in Section 2.2.

PMC sampling has similarities but also differences with respect to MCMC sampling [38]. Both methods are useful tools for the calculation of multi-dimensional summations/integrals. MCMC is easy to implement, however, since the algorithm draws samples and moves them around the equilibrium distribution in relatively small steps, it might take a long time to explore the space of interest [62]. PMC employs the resampling/reweighting frame, which updates the weights by learning from previous proposals and target distributions. The advantage of PMC sampling over MCMC sampling is that the scheme is unbiased at every iteration and therefore can be stopped at any time. PMC sampling is also more robust than MCMC sampling on the initialization of parameters [10]. For these reasons, in this work we focus on the PMC method and devise variants of its standard implementation to improve its performance and applicability in high dimensional scenarios.

1.2.2 State-of-art

The PMC methodology was originally summarized and named by Iba in [38]. The algorithm was first applied for the calculation of the elements of an inverse matrix developed by von Neumann and others [25, 34].Another reference of the PMC algorithm is Metropolis and Ulam [56], where a solver of the Schrödinger equation by random walkers is proposed. According to [14], one of the earliest studies of simple algorithm in statistical science is found in [36].

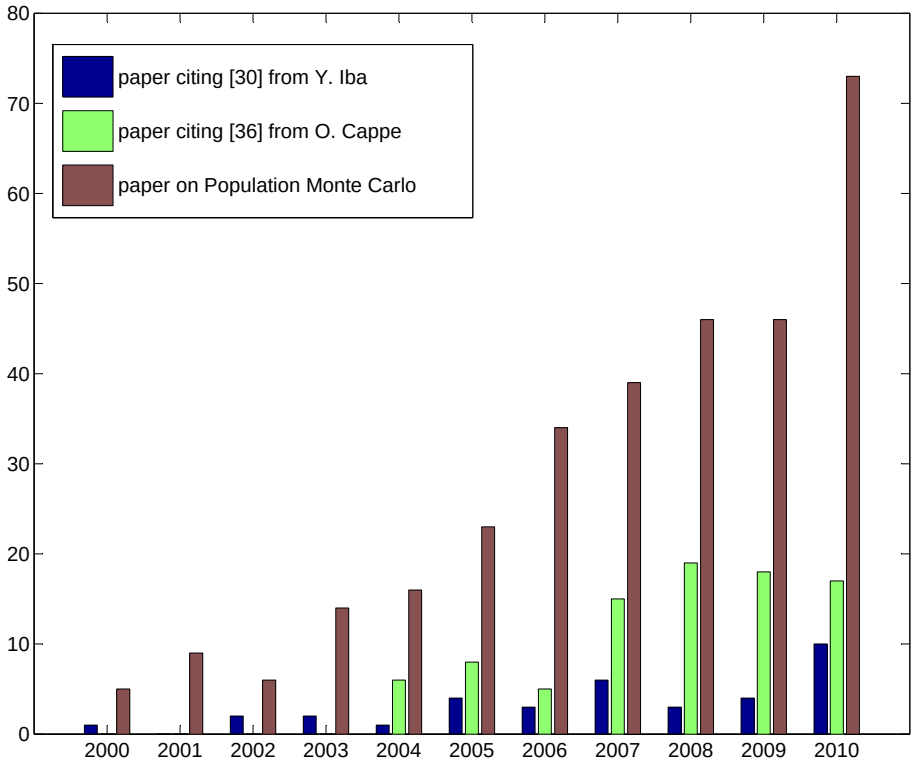


Figure 1.6: Research papers on PMC.

Since the publication of [38], there has been an increasing interest and ongoing research on the topic. A comparison of PMC and MCMC sampling was published in [10]. There are 36 research papers citing [38], 88 papers citing [10], and 311 papers on the topic of the PMC

algorithm or its application between years 2000 and 2010 (see Figure 1.6). An overview of the general PMC algorithm via computation of the products of non-negative sparse matrices is given in [10, 38, 55].

The robustness regarding initialization of the PMC algorithm is shown in [10]. The PMC algorithm based on iterated and adaptive importance sampling for static models was proposed. Models were based on a static setting where the target distribution did not change/evolve over time. PMC algorithms were also shown to be progressively adapted to a target distribution with a diminishing Kullbak divergence in [21]. In [10], analysis of the ion channel model of Hodgson is illustrated using an importance sampling scheme based on a hidden Markov representation, and the comparison of PMC and MCMC is provided.

A PMC scheme is applied to missing data problems in [12]. Instead of using a constant IF or a sequence of IFs, other IFs that depend on both the iteration and the sample index are proposed. Advantages of the new PMC scheme are also illustrated for problems of settings of increasing difficulty, where the missing part can not be simulated or approximated through completion devices. A comparison with MCMC for missing data problems is presented. PMC sampling has better performance than the MCMC method in an example with the stochastic volatility model.

The PMC algorithm was applied to reduce the asymptotic variance for the function of interest in [22]. Variance reduction has always been a critical issue of MC methods. In [22], a set of IFs is iteratively optimized to minimize asymptotic variance. An example of computation of the price of a European option in the Cox-Ingersoll-Ross model is used to explain the implementation of the proposed iterative scheme.

A factored sampling method named condensation was applied to robot vision in [39]. The method was employed to interpret static images. A PMC approach to analyze complex traits was presented in [7]. It can help plant geneticists and breeders in exploiting the marker and phenotypic data on pedigreed populations as available from ongoing breeding programs. The PMC

method was also combined with sequential Monte Carlo chain particle and its effectiveness was illustrated over an example for object tracking in video sequences [5].

The PMC algorithm was applied to a detection system of aerosol-released threats outdoors [3], where it characterized the uncertainty associated with the threat and optimized the detector placement scheme.

PMC sampling was used to improve the motif discovery methods [6] which play pivotal roles in deciphering the genetic regulatory codes in protein sequences. It was shown that PMC method overcame the conventional Expectation Maximization (EM) method's main drawback of being trapped in local optima. In another application, PMC sampling was used in micro simulations in heavy demography data processing [74], where it was able to extract more information than any other traditional method.

1.3 Contributions

As stated along this chapter, high dimensional state space makes the PMC implementation very challenging due to the necessity of very large amounts of samples and the corresponding heavy computation cost. In fact, the PMC algorithm needs not only many more samples but also many more iterations to converge in such cases.

In some of the high dimensional problems, part of the unknown parameters are conditionally linear given the remaining parameters. The marginalized PMC (MPMC) is proposed to lower the computational cost by only generating samples of the nonlinear parameters and marginalizing the remaining linear parameters [8]. This approach is based on the well-known Rao-Blackwell theorem [11].

The computational efficiency of the PMC method can be further improved by the use of a distributed structure [66]. In this thesis, we propose a novel method referred to as Multiple PMC (MultiPMC) where the state space of interest is partitioned into several subspaces with lower dimensions and handled by a set of parallel PMC estimators. Each PMC estimator updates the weights of the samples and the IFs, and if necessary, uses information from the other PMC

estimators. A similar structure used for sequential Monte Carlo methods applied to the problem of target tracking can be found in [19]. A related approach to ours is the one from [24], where the intended application is in speaker recognition. A finite mixture of Gaussians is decomposed into subproblems, which are easier to work with. Then missing data are introduced, and samples are drawn from posteriors. We note, however, that drawing directly from posteriors is often infeasible. In this thesis, we employ the PMC algorithm to make the generation of samples easy.

In all cases discussed above, the IFs update along with iteration number and sample index. However, for simplicity and easy implementation, only Gaussian distributions are used as proposal functions. In almost all applications, a good set of initial samples and proposal functions are important for good performance. In this dissertation we explore these issues and test the effects of different proposal functions on the convergence of estimates.

In particular, we propose that the generating functions be *alternating conditionals*, thereby mimicking the idea behind Gibbs sampling [31]. With this approach, it is expected, that one can efficiently generate particles in high dimensions [20]. We demonstrate the performance of the proposed approach on the problem of frequency estimation of multiple sinusoids.

In summary, in this work we contribute to the advancement of PMC sampling for high dimensional systems by introducing the following innovations:

- **marginalized PMC**, which lowers the dimension of the unknown spaces by marginalization of the linear parameters and therefore decreases the computational cost;
- **multiple PMC**, which partitions the high dimensional space of unknowns into lower dimensional spaces and handles the subproblems with a set of parallel PMC estimators and therefore improves accuracy and reduces complexity;
- **Gibbs PMC**, which efficiently generates samples/particles according to alternating conditionals and achieves better performance.

1.4 Thesis organization

The remaining of this document is organized as follows. Chapter 2 states the mathematical formulation of the problem and describes in detail the standard PMC; and Chapter 3 explained how to choose IFs. The proposed variants, named marginalized PMC, multiple PMC, and Gibbs PMC algorithms are presented in Chapter 4, 5 and 6, respectively. Theoretical issues related to computational complexity of the algorithms is discussed in Chapter 7. Several examples are used to compare the implementation of the studied and proposed algorithms in Chapter 8. Simulation results and performance analysis are also discussed. Finally, Chapter 9 concludes the dissertation with some final thoughts and future work.

CHAPTER 2

Population Monte Carlo algorithms

PMC algorithms are a group of statistical methods based on adaptive importance sampling. They generates samples from proposal distributions that are updated at each iteration by learning from the performances of the previous proposal distributions compared to the target distribution. The target distribution is often the a posteriori distribution of a set of unknowns of interest given observed data and the employed model.

2.1 Mathematical formulation of the problem

We observe a set of data $\mathbf{y}$ which are modeled according to

$$\mathbf{y} = h(\mathbf{x}, \mathbf{v}) \tag{2.1}$$

where $\mathbf{y} \in R^{d_y \times 1}$ (or $C^{d_y \times 1}$) is a vector of observations, $\mathbf{x} \in R^{d_x \times 1}$ is a vector of unknowns, $\mathbf{v} \in R^{d_v \times 1}$ (or $C^{d_v \times 1}$) is a noise vector with a known parametric distribution (typically $d_v = d_y$), and $h : R^{d_x} \times R^{d_v} \to R^{d_y}$ (or $h : R^{d_x} \times C^{d_v} \to C^{d_y}$) is a known function of the unknowns and the noise.

Note that for additive noise case, the general model can be rewritten as

$$\mathbf{y} = h(\mathbf{x}) + \mathbf{v}. \tag{2.2}$$

For simplicity and easiness in the explanation of the proposed algorithms we constrained this work to the additive noise case. Extension to other types of noise may be devised.

For the unknowns, we assume that we have the a priori distribution $\pi(\mathbf{x})$, and that given the noise probability distribution, we can write the conditional distribution $p(\mathbf{y}|\mathbf{x})$. Given the observation vector $\mathbf{y}, \pi(\mathbf{x})$, and $p(\mathbf{y}|\mathbf{x})$, we want to compute the posterior distribution $p(\mathbf{x}|\mathbf{y})$,

which can be written as

$$p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}) \tag{2.3}$$

where $\propto$ symbolizes proportionality. We refer to $p(\mathbf{x}|\mathbf{y})$ as our target distribution. In some cases, we may not be interested in the complete posterior of $\mathbf{x}$, and instead, only in the posterior of some subset of $\mathbf{x}$.

Example:

Consider an example of estimating a set of real sinusoids in noise. The model

$$\mathbf{y} = h(\mathbf{x}) + \mathbf{v} \tag{2.4}$$

where the observation $\mathbf{y} \in R^{d_y \times 1}$ is a $d_y \times 1$ vector, and can be written as

$$y_t = \sum_{k=1}^{K} a_k \cos(2\pi f_k t) + b_k \sin(2\pi f_k t) + v_t, \quad t = 1, 2, ..., d_y$$

The unknown parameter $\mathbf{x} \in R^{3K \times 1}$ vector, where K is the number of sinusoids to be estimated. The parameters a_k and b_k are the amplitudes of the cosine and sine components, respectively, of the k-th sinusoid whose frequency is f_k. The function $h(\cdot)$ is a nonlinear function of the parameters in $\mathbf{x}$; and $\mathbf{v} \in R^{d_y \times 1}$ is a white Gaussian noise vector with a covariance matrix $\mathbf{C}_v$.

2.2 The PMC algorithm

The PMC algorithm approximates a stationary target distribution by an iterative sampling procedure. The underlying principle of PMC is importance sampling [23, 68]. The importance sampling has primarily been used for numerical integration and later in particle filtering [23]. In estimation theory, a standard problem is the estimation of unknowns given the

observations and the problem model. The objective is often to get the point estimates of the unknowns or to provide their posterior distributions.

Let the unknowns be a vector denoted by $\mathbf{x}$, and the observations be a vector presented by $\mathbf{y}$. A commonly used point estimator is the minimum mean-square error estimator (MMSE), which is defined as

$$\eta_x = \int \mathbf{x} p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \tag{2.5}$$

where $p(\mathbf{x}|\mathbf{y})$ is the posterior of $\mathbf{x}$. If we can draw samples from the posterior,

$$\mathbf{x}^{(m)} \sim p(\mathbf{x}|\mathbf{y}), \quad m = 1, 2, ..., M$$

where M is the total number of independently drawn samples, then we can compute the integral in equation (2.5) according to classical Monte Carlo integration by

$$\hat{\eta}_x \simeq \frac{1}{M} \sum_{m=1}^{M} \mathbf{x}^{(m)}. \tag{2.6}$$

This estimate will converge to the true value by the strong law of large numbers [37,45]. Moreover, for large M one can expect

$$\frac{\hat{\eta}_x - \eta_x}{\sigma_{\hat{\eta}_x}} \sim \mathcal{N}(0, 1)$$

where

$$\sigma_{\hat{\eta}_x} = \sqrt{\frac{1}{M} \sum_{m=1}^{M} (x^{(m)} - \hat{\eta}_x)^2}.$$

However, samples usually can not be drawn directly from the posterior $p(\mathbf{x}|\mathbf{y})$ in practice. Alternatively, samples can be generated from another probability distribution $q(\mathbf{x})$, called

importance function, and the estimate is computed as

$$\begin{aligned}\eta_x &= \int \mathbf{x} p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \\ &= \int \mathbf{x} \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \\ &\simeq \frac{1}{M} \sum_{m=1}^{M} \frac{\mathbf{x}^{(m)} p(\mathbf{x}^{(m)}|\mathbf{y})}{q(\mathbf{x}^{(m)})}.\end{aligned}$$

When $q(\mathbf{x})$ satisfies some conditions, it can be shown that by the strong law of large numbers, this estimate also converges to the true mean of the posterior [13, 60].

The above approximation is often modified to

$$\eta_x \simeq \sum_{m=1}^{M} w^{(m)} \mathbf{x}^{(m)} \tag{2.7}$$

where $w^{(m)}$ denotes the weight of sample $\mathbf{x}^{(m)}$, i.e.,

$$w^{(m)} \propto \frac{p(\mathbf{x}^{(m)}|\mathbf{y})}{q(\mathbf{x}^{(m)})} \tag{2.8}$$

and

$$\sum_{m=1}^{M} w^{(m)} = 1. \tag{2.9}$$

Here we remark that the goal of PMC is much more ambitious than simply obtaining point estimates. With PMC one approximates the posterior distribution of the vector of unknowns $\mathbf{x}$ given the set of observations $\mathbf{y}$, i.e., $p(\mathbf{x}|\mathbf{y})$. This distribution contains all the possible information about $\mathbf{x}$ and from it, one can obtain the MMSE, the maximum a posteriori (MAP), or other point estimates.

PMC employs an iterated and adaptive importance sampling scheme. It also uses resampling as do particle filtering methods to prevent sample degeneration [23, 28], where samples with small weights are most likely removed and ones with large weights are replicated.

The resampling method [30, 63] is an extension of the importance sampling method that convert a set of weighted samples $\left\{\mathbf{x}^{(m)}, \mathbf{w}^{(m)}\right\}$ to a set of unweighted samples $\left\{\tilde{\mathbf{x}}^{(m)}\right\}$.

The goal of resampling is to eliminate the samples/particles with low weights while distributing more particles in more probable target regions. The decision of resampling can be made based on many methods/rules [11]. The most standard and efficient approach is multinomial sampling from $\mathbf{x}^{(m)}$ with probabilities proportional to the importance weights $\mathbf{w}^{(m)}$. We replace the weighted samples

$$\left\{\mathbf{x}^{(m)}\right\} = \left\{x^{(1)},\ x^{(2)},\ ...,\ x^{(M)}\right\}$$

with unweighted samples

$$\left\{\tilde{\mathbf{x}}^{(m)}\right\} = \left\{\tilde{x}^{(1)},\ \tilde{x}^{(2)},\ ...,\ \tilde{x}^{(M)}\right\}$$

where

$$\tilde{x}^{(i)} = x^{(l_i)},\ \ 1 \leq i \leq M$$

and the $\{(m)\}$ follows the multinomial distribution

$$\left\{x_{(1)},\ x_{(2)},,\ ...,\ x_{(M)},\right\} \propto \mathcal{M}(M,\ w^{(1)},\ w^{(2)},\ ...,\ w^{(M)})$$

i.e.,

$$p(l_i = k|x^{(1)},\ x^{(2)},\ ...,\ x^{(M)}) \propto w^{(k)},\quad 1 \leq i \leq M,\ 1 \leq k \leq M.$$

Figure 2.1 shows two simple examples of sampling, weighting and resampling procedures. The solid black curve represents the target distribution and the dashed red line represents the proposal, i.e. IF. If one samples from the IF of a uniform distribution (i. e. the proposal in the upper example), after weighting and resampling, the new set of samples will approximate the target distribution. If one samples from the IF which is the same as the target distribution (i.e. the proposal in the lower example), the set of samples will get equally weights and the new

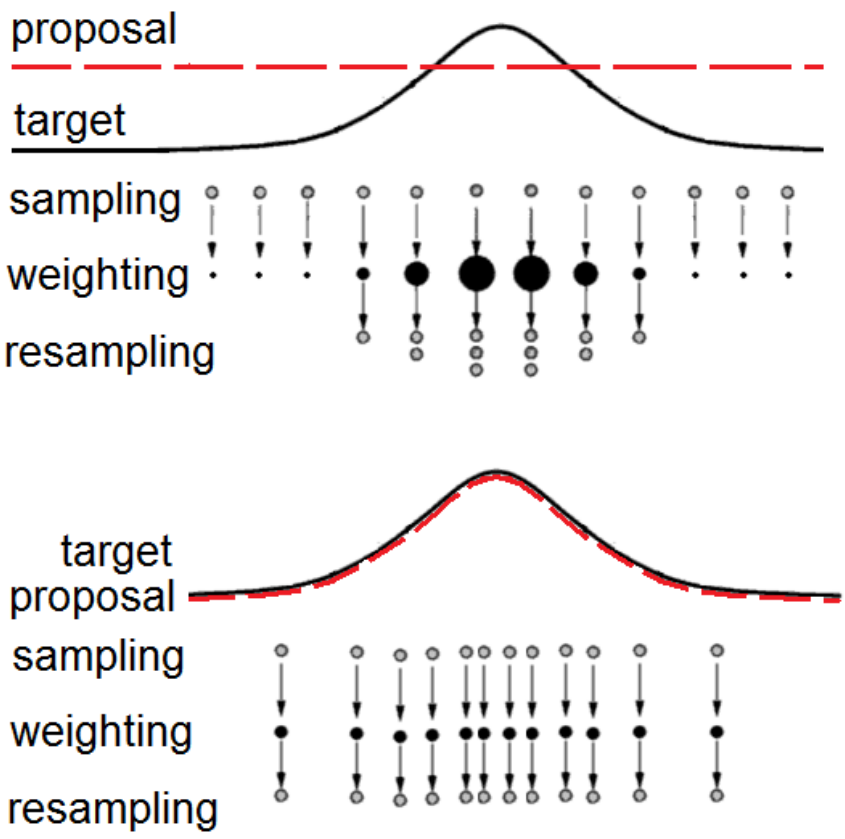


Figure 2.1: Examples of sampling, weighting and resampling.

set is the same as the original one. The whole set of samples approximate the target distribution as well. These two simple example shows the idea of the sampling, weighting and resampling.

After all, the PMC method can be summarized as follows. Let j denote the iteration number, $j = 1, 2, ..., J$, and let m represent the index of the particle, $m = 1, 2, ..., M$.

Algorithm for PMC:

`Step 1.` Choose an importance function $q_j^{(m)}(\mathbf{x})$;

`Step 2.` Draw samples $\mathbf{x}_j^{(m)}$ from $q_j^{(m)}(\mathbf{x})$;

`Step 3.` Compute the weights of the samples by

$$\tilde{w}_j^{(m)} = \frac{p(\mathbf{x}_j^{(m)}|\mathbf{y})}{q_j^{(m)}(\mathbf{x}_j^{(m)})}; \tag{2.10}$$

`Step 4.` Normalize the weights according to

$$w_j^{(m)} = \frac{\tilde{w}_j^{(m)}}{\sum_{k=1}^{M} \tilde{w}_j^{(k)}}; \tag{2.11}$$

`Step 5.` Resample the samples according to their weights;

`Step 6.` If more iterations are needed, set $j = j + 1$, and go back to step 1;

`Step 7.` At the end, obtain the MMSE estimate from all available samples by

$$\tilde{\mathbf{x}} = \frac{\sum_{j=1}^{J} \sum_{m=1}^{M} w_j^{(m)} \mathbf{x}_j^{(m)}}{\sum_{j=1}^{J} \sum_{m=1}^{M} w_j^{(m)}}.$$

Each iteration of the PMC algorithm produces a set of samples approximately simulated from the target distribution but the iterative structure allows for adaptivity toward the target distribution. Since the validation is based on importance sampling principles, dependence on the past samples can be arbitrary and the approximation to the target is unbiased at each iteration and does not require convergence times nor stopping rules [27]. The PMC algorithm therefore can be stopped at each iteration as the sampling is statistically unbiased. Note that all available weighted samples (from all available iterations) are used to approximate the posterior $p(\mathbf{x}|\mathbf{y})$.

Example:

Consider an example of estimating a bimodal distribution in noise. We had 10 observations from

$$\mathbf{y} \sim 0.375\mathcal{N}(\theta_1, \sigma_1^2) + 0.625\mathcal{N}(\theta_2, \sigma_2^2)$$

where $\theta_1 = 0.24$, $\theta_2 = 0.265$ and $\sigma_2^2 = \sigma_1^2 = 0.002$. The objective is to estimate θ_1 and θ_2 and to approximate the bimodal distribution. The prior of these two modes are $\theta_1 \sim \mathcal{N}(0, 5)$ and $\theta_2 \sim \mathcal{N}(0, 5)$.

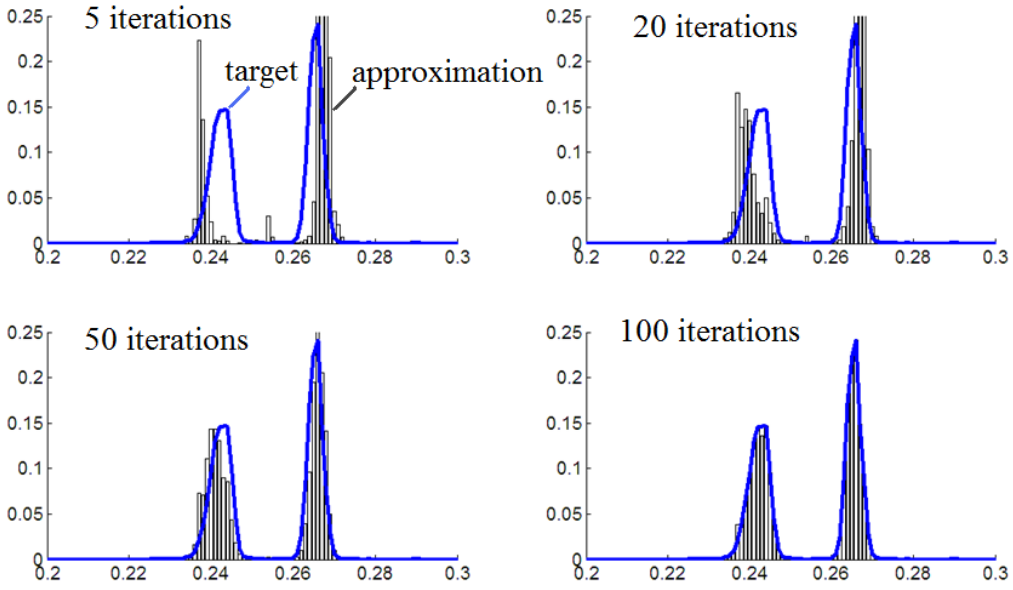


Figure 2.2: Approximation of a bimodal distribution.

We used $5,000$ samples per iteration when implementing the PMC algorithm to this simple example. We used normal distributions as IFs and we draw samples

$$\mathbf{x}_j^{(m)} \sim \mathcal{N}(\mathbf{x}_{j-1}^{(m)}, 0.002)$$

Figure 2.2 shows the approximations along iterations. From the approximated distribution, one can easily get the point estimates for θ_1 and θ_2.

Note that, a total number of $500,000$ samples were generated when applying the standard PMC to estimate the above two-dimensional problem. As the dimension of the parameter space increases, the use of PMC becomes challenging because it requires generation of a large number of samples. Other techniques will be introduced in this work to improve the PMC sampling efficiency or lower the dimension of the space of unknowns.

CHAPTER 3

Importance Functions

PMC algorithm is a statistical method and is used for generation of samples approximately from a target distribution. The method is iterative in nature and is based on the principle of importance sampling [59, 63].

3.1 How to choose importance functions

The idea of PMC is to apply the importance sampling iteratively, that is, to obtain a approximation of a target distribution, in our case a posterior $p(\mathbf{x}|\mathbf{y})$, iteratively. Suppose that one starts sampling from a proposal/IF $q_1(\mathbf{x})$

$$x_1^{(m)} \sim q_1(\mathbf{x}).$$

These samples $x_1^{(m)}$ from the support of a discrete random measure

$$\chi_1 = \left\{x_1^{(m)}, w_1^{(m)}\right\}_{m=1}^{M}$$

where $w_1(m)$ are the normalized weights associated to the samples

$$w_1^{(m)} \propto \frac{p(\mathbf{x}_1^{(m)}|\mathbf{y})}{q_1(\mathbf{x}_1^{(m)})}$$

where

$$\sum_{m=1}^{M} w_1^{(m)} = 1.$$

Once the first random measure χ_1 is obtained, based on its support one can construct a better importance function $q_2(\mathbf{x})$ followed by the generation of a new set of samples $x_2^{(m)}$ from it and association of new weights $w_2^{(m)}$ to the samples. Thereby, one obtains χ_2 and continues

to construct in a similar fashion χ_3, χ_4, and so on. Note that, all available samples $(x)_j^{(m)}$, $j = 1, 2, ..., J$ are used to construct χ_{J+1} from which one proposes $q_{J+1}(\mathbf{x})$. These random measures χ_j approximate the target distribution $p(\mathbf{x}|\mathbf{y})$ and can be used for computing estimates of integrals under that distribution.

3.2 Importance functions with single transition kernel

The standard PMC implements the importance sampling using normal distribution with a single-kernel proposal. Initially one samples from a proposal/IF $q_1(\mathbf{x})$

$$x_1^{(m)} \sim q_1(\mathbf{x})$$

followed by weighting and resampling stated as in Chapter 2. In each later iteration, a new sample $\mathbf{x}_j^{(m)}$ is draw from a normal distribution

$$\mathbf{x}_j^{(m)} \sim q_j^{(m)}(\mathbf{x}) = \mathcal{N}(\mathbf{x}_{j-1}^{(m)}, v)$$

where v is a constant variance.

The performance of the PMC algorithm relies on the choice of the variance v of the transition kernel. It might take a long time for the PMC sampler to approach the target area with a relative small variance. However, small variance helps to better explore the details around the target area. To improve the sampling efficiency, one might be interested in the study of dynamic variance or multiple variances.

3.3 Importance functions with multiple transition kernels

In this work, we used a scheme of multiple IFs (i.e. multiple transition kernels) to adaptively improve the sampling efficiency. In each iteration, a new sample $\mathbf{x}_j^{(m)}$ is draw from

$$\mathbf{x}_j^{(m)} \sim \mathcal{N}(\mathbf{x}_{j-1}^{(m)}, v_n), \quad n \in \{1, 2, ..., N\}$$

where $\{v_1, v_2, ..., v_N\}$ are a set of pre-selected varicances. Between different iterations, the algorithm can change the structure of the transition kernels to ensure that the subsequent sampling procedure is carried out more efficiently. An algorithm to adaptively choose transition kernels was proposed in [21] and it shown that the asymptotic variance of the estimates decreased by applying the proposed scheme. A fixed number of pre-selected transition kernels was used in [9]. The efficiency of the algorithm were demonstrated by an example with the posterior being a mixture Gaussian distribution. It was shown that the produced samples by the algorithm accurately approximates the distribution.

In this work, we propose and use multiple IFs as follows. Assume that the total number of iterations was J, and the total number of samples per iteration was M. At initial iteration $j = 1$, the IF $q_j^{(m)}(\mathbf{x})$ was chosen as follows.

For one problem of sinusoids in noise, we used complex normal distributions for unknowns with initial mean at $\mathbf{x}_0$, which is obtained by Yule-Walker methods [67] (other preprocessing methods also available), and variance that was randomly chosen from a set of multiple transition kernels [9]

$$\mathbf{v}_x = [v_{x,1},\ v_{x,2},\ ...,\ v_{x,N}]^\top.$$

At initial step, each variance was selected with probability of $\alpha_{x,d} = \frac{1}{N}$, with $d = 1,\ 2,\ ,...,\ N$.

For example, if we use five transition kernels for an unknown frequency who has prior as $x \sim \mathcal{U}[0, 1]$, it could be

$$\mathbf{v}_x = {\sigma_x}^2 \times [1^2, 0.1^2, 0.01^2, 0.001^2, 0.0001^2]^\top$$

with a constant ${\sigma_x}^2 = 0.5^2$. The transition kernel with the largest variance makes the samples easily explore the whole range of the frequency at any time; while the other transition kernels allow more detailed exploration in the space of the unknown frequency as the samples moving towards the target area.

Initially, each variance was selected with probability of $\alpha_{x,d} = \frac{1}{5}$, with $d = 1,\ 2,\ , ...,\ 5$. Then the algorithm with multiple IFs iterate as follows:

1. At iteration $j = 1$, we used complex normal distributions as IFs for the unknowns $\mathbf{x}^{(m)}$, with mean at $\mathbf{x}_0$ and variance randomly chosen from $\mathbf{v}_x$ randomly with probability of $\alpha_{x,d} = \frac{1}{5}$. At iteration $j > 1$, we used complex normal distributions as IFs for the unknowns $\mathbf{x}^{(m)}$, with mean at $\mathbf{x}_{j-1}^{(m)}$ and variance that was randomly chosen from $\mathbf{v}_x$ with probability of updated $\alpha_{x,d}$.

2. Draw samples $\mathbf{x}_j^{(m)}$ from $q_j^{(m)}(\mathbf{x})$.

3. Compute the weights of the samples by

$$\tilde{w}_j^{(m)} \propto \frac{p(\mathbf{x}_j^{(m)}|\mathbf{y})}{q_j^{(m)}(\mathbf{x}_j^{(m)})};$$

4. Normalize the weights according to

$$w_j^{(m)} = \frac{\tilde{w}_j^{(m)}}{\Sigma_{k=1}^{M} \tilde{w}_j^{(k)}}.$$

5. Resample the samples according to their weights.

6. Update $\alpha_{x,d}$ with $d = 1,\ 2,\ , ...,\ N$

$$\alpha_{x,d} = \sum_{m=1}^{M} w_j^{(m)} \mathbf{I}_d \left(v_{x,j}^{(m)} = v_{x,d} \right)$$

where $\mathbf{I}_d\left(v_{x,j}^{(m)} = v_{x,d}\right)$ is an indication function

$$\mathbf{I}_d\left(v_{x,j}^{(m)} = v_{x,d}\right) = \begin{cases} 1, & when\ v_{x,j}^{(m)} = v_{x,d} \\ 0, & when\ v_{x,j}^{(m)} \neq v_{x,d} \end{cases}.$$

In order to keep every variance valid after each iteration, re-scaling was employed to $\alpha_{x,d}$ to ensure that the minimum weight for each available variance is $\frac{1}{4N}$.

Example:

Consider an example of estimating a sinusoid in noise. We had $d_y = 20$ observations from

$$y_t = a\cos(2\pi ft + \phi) + v_t,$$
$$t = 1, 2, ..., 20.$$

The unknown parameter $\mathbf{x} = [a, \phi, f] = [1, 0, 0.5]$ is a 3×1 vector, where a_k is the amplitude of the cosine, ϕ is the phase and f is the frequency, respectively. The white Gaussian noise vector $\mathbf{v} \in R^{20\times 1}$ has known covariance matrix $\mathbf{C}_v = \sigma_v^2\mathbf{I}$, where $\mathbf{I} \in R^{20\times 20}$ is an identity matrix.

We applied the standard PMC and the PMC with multiple IFs to this estimation problem. The standard PMC used IFs as

$$q(\mathbf{x})_j^{(m)} \sim \mathcal{N}(\mathbf{x}_{j-1}^{(m)}, 0.005^2)$$

with a constant variance for all samples. The PMC with multiple IFs used a set of pre-selected variances as

$$\mathbf{v} = [v_1, v_2, ..., v_5] = 0.5^2 \times [1^2, 0.1^2, 0.01^2, 0.001^2, 0.0001^2]^\top$$

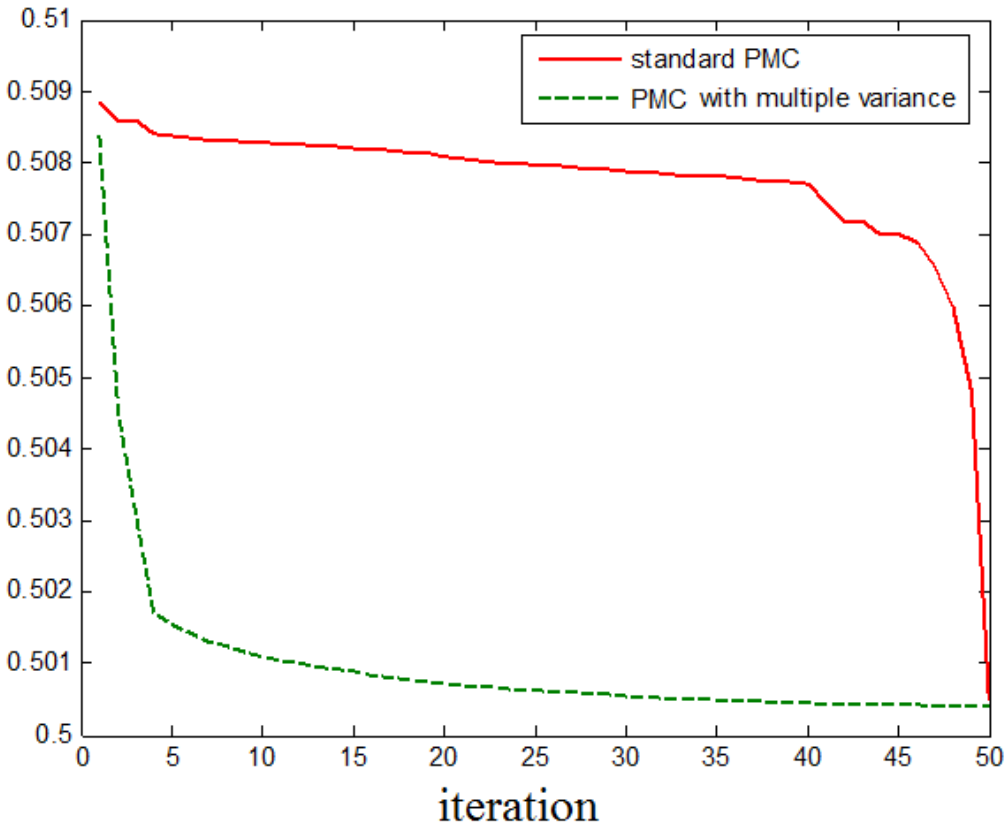


Figure 3.1: Performance of standard PMC and the PMC with multiple IFs.

and the IFs are

$$q(\mathbf{x})_j^{(m)} \sim \mathcal{N}(\mathbf{x}_{j-1}^{(m)}, v_n), \quad n \in \{1, 2, ..., 5\}.$$

Both methods used 500 samples for each iteration and the results are shown in Figure 3.1. It took much less iterations for the PMC with multiple IFs to reach the target area.

The above example shows that the multiple choices of IFs with adaptive weights helps the PMC algorithms to explore the space of unknowns more efficiently.

CHAPTER 4

Marginalized Population Monte Carlo algorithms

The PMC methods are of great interest in many non-linear problems where standard computational methods are difficult to implement. However, when the dimension of the parameter space increases, the use of PMC is also challenging due to the requirement of generation of a large number of samples. In many high-dimensional problems, some of the unknown parameters are conditionally linear given the remaining parameters. By marginalizing part of the unknowns, MPMC algorithm is expected to lower the computational complexity.

4.1 Mathematical formulation of the problem

We assume that the model of the data is

$$\mathbf{y} = h(\mathbf{x}_n) + A(\mathbf{x}_n)\mathbf{x}_l + \mathbf{v} \tag{4.1}$$

where the observation vector is $\mathbf{y} \in R^{d_y \times 1}$ and the unknown parameter vector is $\mathbf{x} \in R^{d_x \times 1}$. The vector $\mathbf{x}$ is composed of nonlinear parameters $\mathbf{x}_n \in R^{d_{x_n} \times 1}$ and linear parameters $\mathbf{x}_l \in R^{d_{x_l} \times 1}$, where $d_x = d_{x_n} + d_{x_l}$, and the prior density of $\mathbf{x}$ is given by $p(\mathbf{x}_n, \mathbf{x}_l)$. As in (2.2), $h(\cdot)$ is a nonlinear function of the parameters $\mathbf{x}_n$; $A(\mathbf{x}_n)$ is a matrix of functions of the nonlinear parameter $\mathbf{x}_n$ and has dimension $d_y \times d_{x_l}$; and $\mathbf{v}$ is a $d_y \times 1$ noise vector with a known probability distribution $p(\mathbf{v})$.

Example:

Consider the problem of frequency estimation of sinusoids in noise. A modified version of the model is given by:

$$\mathbf{y} = h(\mathbf{x}) + \mathbf{v} = \mathbf{A}(\mathbf{x}_n)\mathbf{x}_l + \mathbf{v}$$

More specifically,

$$\begin{aligned} y_t &= \sum_{k=1}^{K} a_k \cos(2\pi f_k t) + b_k \sin(2\pi f_k t) + v_t \\ t &= 1, 2, ..., d_y \end{aligned}$$

The unknown vector of parameters is $\mathbf{x} \in R^{3K\times 1}$, which is composed of nonlinear parameters $\mathbf{x}_n \in R^{K\times 1}$ and linear parameters $\mathbf{x}_l \in R^{2K\times 1}$, where K is the number of sinusoids. The parameters a_k and b_k are the amplitudes of the cosine and sine components, respectively, of the k-th sinusoid whose frequency is f_k. The function $h(\cdot)$ is a nonlinear function of $\mathbf{x}$; $\mathbf{A}(\mathbf{x}_n) \in R^{d_y\times 2K}$ is a matrix of functions of the nonlinear parameters $\mathbf{x}_n$; and $\mathbf{v} \in R^{d_y\times 1}$ is a white Gaussian noise vector with covariance matrix $\mathbf{C}_v$.

The parameters to be estimated are $\mathbf{x} = [\mathbf{x}_l; \mathbf{x}_n]$, where

$$\mathbf{x}_l = [a_1, b_1, a_2, b_2, ..., a_K, b_K]^T$$
$$\mathbf{x}_n = [f_1, f_2, ..., f_K]^T.$$

The prior of the amplitudes and the frequencies are considered independent, i.e.,

$$p(\mathbf{x}) = p(\mathbf{x}_l, \mathbf{x}_n) = p(\mathbf{x}_l)p(\mathbf{x}_n).$$

4.2 The MPMC algorithm

MPMC employs a scheme where PMC is only applied to the nonlinear parameters, while the linear parameters are integrated out (marginalized) [8]. This approach is analogous to the well-known Rao-Blackwell (RB) theorem used in particle filtering/sequential Monte Carlo [11]. Previous work on this subject can be found in [12], where RB was applied to marginalization of missing data through numerical integration. In this work, we consider the general problem of having conditionally linear parameters, assume certain structure of distributions that allow for analytical integration and apply sampling only for the nonlinear parameters of the model. We argue that the MPMC method deals with lower dimension of space of unknowns than PMC,

therefore it can construct new proposals much more easily. We expect that the marginalization can achieve an improved computational efficiency.

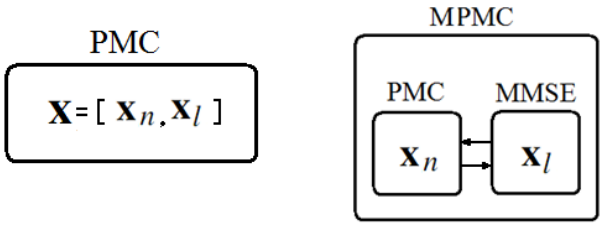


Figure 4.1: PMC and MPMC samplers.

As shown in Figure 4.1, in the PMC algorithm, PMC sampling is applied to the whole unknown vector $\mathbf{x} = [\mathbf{x}_n, \mathbf{x}_l]$; in the MPMC algorithm, PMC sampling is only applied to the nonlinear paratmeters $\mathbf{x}_n$ and the linear ones $\mathbf{x}_l$ are handled by other analytical methods, for instance MMSE estimator. Specifically, at iteration j, one only generates samples of the nonlinear parameters, $\mathbf{x}_{n,j}^{(m)}$. The corresponding weights to these samples are

$$w_{n,j}^{(m)} \propto \frac{p(\mathbf{x}_{n,j}^{(m)}|\mathbf{y})}{q_{n,j}^{(m)}(\mathbf{x}_{n,j}^{(m)})}. \tag{4.2}$$

The numerator $p(\mathbf{x}_{n,j}^{(m)}|\mathbf{y})$ is the marginalized posterior of $\mathbf{x}_n$, which satisfies

$$p(\mathbf{x}_{n,j}^{(m)}|\mathbf{y}) \propto \int p(\mathbf{y}|\mathbf{x}_l, \mathbf{x}_{n,j}^{(m)}) p(\mathbf{x}_l, \mathbf{x}_{n,j}^{(m)}) d\mathbf{x}_l. \tag{4.3}$$

Based on $\mathbf{x}_{n,j}^{(m)}$, one can simply use the MMSE criterion to estimate the corresponding $\mathbf{x}_l$.

The proposed MPMC algorithm is summarized as follows. Let j denote the iteration number, $j = 1, 2, ...$, and let m denote the index of the particle, $m = 1, 2, ..., M$.

Algorithm for MPMC:

`Step 1.` Choose an importance function $q_{n,j}^{(m)}(\mathbf{x}_{n,j})$;

`Step 2.` Draw samples $\mathbf{x}_{n,j}^{(m)}$ from $q_{n,j}^{(m)}(\mathbf{x}_{n,j})$;

`Step 3.` Based on $\mathbf{x}_{n,j}^{(m)}$, use the MMSE criterion to estimate the corresponding $\mathbf{x}_l$;

`Step 4.` Compute the weights of the samples by

$$\tilde{w}_{n,j}^{(m)} = \frac{p(\mathbf{x}_{n,j}^{(m)}|\mathbf{y})}{q_{n,j}^{(m)}(\mathbf{x}_{n,j}^{(m)})}; \tag{4.4}$$

`Step 5.` Normalize the weights according to

$$w_{n,j}^{(m)} = \frac{\tilde{w}_{n,j}^{(m)}}{\sum_{k=1}^{M} \tilde{w}_{n,j}^{(k)}}; \tag{4.5}$$

`Step 6.` Resample the samples according to their weights;

`Step 7.` If more iterations are needed, set $j = j + 1$, and go back to step 1;

`Step 8.` At the end, obtain the MMSE estimate of the nonlinear parameters from all available samples by

$$\tilde{\mathbf{x}}_n = \frac{\sum_{j=1}^{J} \sum_{m=1}^{M} w_{n,j}^{(m)} \mathbf{x}_{n,j}^{(m)}}{\sum_{j=1}^{J} \sum_{m=1}^{M} w_{n,j}^{(m)}}.$$

The estimate of the linear parameters $\tilde{\mathbf{x}}_l$ can be obtained using the MMSE criterion. Note that all available weighted samples (from all iterations) are used to approximate the posterior $p(\mathbf{x}|\mathbf{y})$.

CHAPTER 5

Multiple Population Monte Carlo algorithms

The computational efficiency of the PMC method can be further improved by the use of a distributed structure [66]. We propose a novel method referred to as Multiple PMC (MultiPMC) where the state space of interest is partitioned into several subspaces with lower dimensions and handled by a set of parallel PMC samplers [19, 24]. Each PMC sampler updates the weights of the samples and the importance functions, if necessary, using information from the other PMC samplers. A similar structure used for sequential Monte Carlo methods applied to the problem of target tracking can be found in [19]. A related approach to ours is the one from [24], where the intended application is in speaker recognition. A finite mixture of Gaussians is decomposed into subproblems, which are easier to work with. Then missing data are introduced, and samples are drawn from posteriors. We note, however, that drawing directly from posteriors is often infeasible. Here, we employ PMC algorithms to make the generation of samples easy.

The partitioning of the state often depends on the problem [52]. By decomposing the original problem, one can considerably reduce the computational complexity. In the next section we apply the idea of multiple samplers to the standard PMC, then we merge the concept of multiple samplers with MPMC to come up with a more simplified scheme.

5.1 Multiple PMC

We will further assume that the model in equation (2.2) can be partitioned into K subproblems as follows:

$$\mathbf{y} = \sum_{k=1}^{K} h_k(\mathbf{x}_k) + \mathbf{v} \tag{5.1}$$

where $[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_K]$ forms the unknown vector $\mathbf{x}$ in the general model described by (2.2).[1]

[1] Equation (5.1) represents only one particular case where MultiPMC can be applied.

Example:

Take again the problem of estimating a set of sinusoids in noise. The problem can be modeled as

$$y_t = \sum_{k=1}^{K} \tilde{A}_k e^{i2\pi f_k t} + v_t, \qquad t = 1, 2, ..., d_y$$

where $i = \sqrt{-1}$; $\tilde{A}_k = A_k e^{i\phi_k} \in C$ is the complex amplitudes of the k-th component whose frequency is $f_k \in [0, 1]$, and v_t is white Gaussian noise. The parameters to be estimated are $\mathbf{x} = [\tilde{A}_1, f_1, ..., \tilde{A}_K, f_K]$, a $3K$-dimension vector of unknowns. Assume that we know the total number of sinusoids K. The problem can be easily decomposed as K subproblems and handled by K PMC samplers, where the k-th sampler will target the unknowns $\mathbf{x}_k = [\tilde{A}_k, f_k]$ of dimension 3.

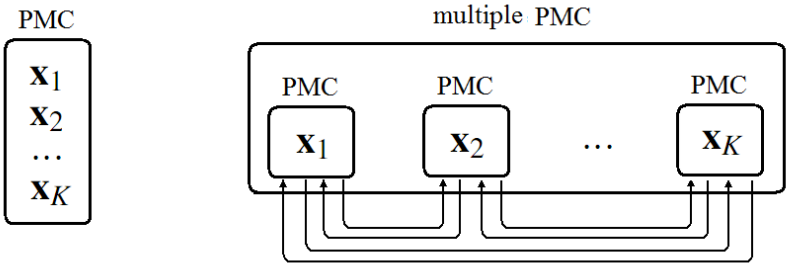


Figure 5.1: PMC and multiple PMC samplers.

Once the problem is partitioned, we assign each unknown vector $\mathbf{x}_k$ a PMC sampler, shown in Figure 5.1. The sample generation/propagation and resampling step of each PMC sampler can be implemented as the algorithm stated in Section 2.2. The key question is the weight updating of the samples for $\mathbf{x}_k$ based on the other PMC samplers. Ideally, the weight

update $\tilde{w}_{k,j}^{(m)}$ should be carried out by

$$\tilde{w}_{k,j}^{(m)} = \frac{p(\mathbf{x}_{k,j}^{(m)}|\mathbf{x}_{-k,j}, \mathbf{y})}{q_{k,j}^{(m)}(\mathbf{x}_j^{(m)})} \tag{5.2}$$

where $\mathbf{x}_{-k,j}$ contains the true values of all unknowns except $\mathbf{x}_{k,j}$. This form of update requires the knowledge of $\mathbf{x}_{-k,j}$, which is not available. Here we propose to implement the updates as

$$\tilde{w}_{k,j}^{(m)} = \frac{p(\mathbf{x}_{k,j}^{(m)}|\tilde{\mathbf{x}}_{-k,j}, \mathbf{y})}{q_{k,j}^{(m)}(\mathbf{x}_{k,j}^{(m)})} \tag{5.3}$$

where $\tilde{\mathbf{x}}_{-k,j}$ are the most recent estimated values of all the unknowns except $\mathbf{x}_{k,j}$

$$\tilde{\mathbf{x}}_{-k,j} = \tilde{\mathbf{x}} \setminus \tilde{\mathbf{x}}_{k,j} = [\tilde{\mathbf{x}}_{1,j}^\top, \tilde{\mathbf{x}}_{2,j}^\top, ..., \tilde{\mathbf{x}}_{k-1,j}^\top, \tilde{\mathbf{x}}_{k+1,j}^\top, ..., \tilde{\mathbf{x}}_{K,j}^\top]^\top$$

and

$$\tilde{\mathbf{x}}_{k,j} = \sum_{m=1}^{M} w_{k,j}^{(m)} \mathbf{x}_{k,j}^{(m)} \tag{5.4}$$

where $w_{k,j}^{(m)}$ is the normalized weight.

If the estimations from (5.4) are not available yet, another possibility is to use

$$\tilde{\mathbf{x}}_{k,j-1} = \sum_{m=1}^{M} w_{k,j-1}^{(m)} \mathbf{x}_{k,j-1}^{(m)}. \tag{5.5}$$

In each iteration, the PMC samplers use the exchanged estimates to compute their weights in an alternating way. For good performance of the method, we propose that in each iteration the implementation order of the PMC samplers is selected randomly. Note that the exchanging information among the PMC samplers is not limited to the estimates. Suppose all the samples/particles in one of the subspace form up a distribution with multiple modes, this very PMC sampler should provide all these modes to the rest samplers instead of only one estimate. In other words, if the samples/particles in one of the subspace has several clusters, this PMC sampler should deliver at least efficient information one estimate from each cluster to the rest PMC samplers.

Let j denote the iteration number, $j = 1, 2, ...,$ k denote the k-th PMC sampler, and m represent the index of the particle in each sampler, $m = 1, 2, ..., M$. At the j-th iteration, the MultiPMC method can be summarized as follows.

Algorithm for multiple PMC:

`Step 1.` Randomly choose the order of PMC samplers. Let the order be $l_1, l_2, ..., l_k, ..., l_K$.

`Step 2.` For the l_k-th PMC sampler, choose the importance function $q_{l_k,j}^{(m)}(\mathbf{x}_{l_k,j}^{(m)})$, $m = 1, 2, ..., M$; and draw samples $\mathbf{x}_{l_k,j}^{(m)}$ from $q_{l_k,j}^{(m)}(\mathbf{x}_{l_k,j}^{(m)})$;

`Step 3.` Compute the weights of the samples by

$$\tilde{w}_{l_k,j}^{(m)} = \frac{p(\mathbf{x}_{l_k,j}^{(m)}|\tilde{\mathbf{x}}_{-l_k}, \mathbf{y})}{q_{l_k,j}^{(m)}(\mathbf{x}_{l_k,j}^{(m)})}$$

`Step 4.` Normalize the weights according to

$$w_{l_k,j}^{(m)} = \frac{\tilde{w}_{l_k,j}^{(m)}}{\Sigma_{k=1}^{M} \tilde{w}_{l_k,j}^{(m)}}; \tag{5.6}$$

`Step 5.` Resample the samples according to their weights;

`Step 6.` Obtain the most recent estimated values for the l_k-th sampler by

$$\tilde{\mathbf{x}}_{k,j} = \sum_{m=1}^{M} w_{k,j}^{(m)} \mathbf{x}_{k,j}^{(m)};$$

`Step 7.` When all K samplers finish the above steps and if more iterations are needed, let j equal $(j + 1)$ and go back to `step 1`;

`Step 8.` At the end, the estimated parameters are

$\tilde{\mathbf{x}}_j = [\tilde{\mathbf{x}}_{1,j}^\top, \tilde{\mathbf{x}}_{2,j}^\top, ..., \tilde{\mathbf{x}}_{K,j}^\top]^\top$. Note that all available weighted samples (from all samplers and all iterations) are used to approximate the posterior $p(\mathbf{x}|\mathbf{y})$.

5.2 Multiple MPMC

The distributed structure of multiple samplers can also be applied to the MPMC methods, shown in Figure 5.2.

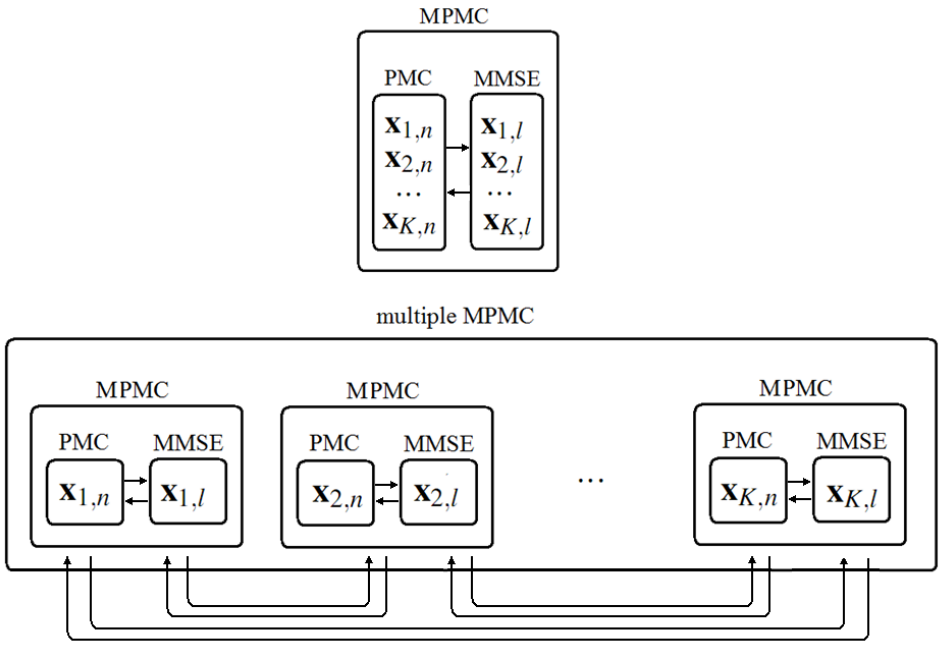


Figure 5.2: MPMC and multiple MPMC samplers.

If we modify (4.1) for a model of type (5.1), we can write

$$\mathbf{y} = \sum_{k=1}^{K} (h_k(\mathbf{x}_{k,n}) + A_k(\mathbf{x}_{k,n})\mathbf{x}_{k,l}) + \mathbf{v}, \tag{5.7}$$

where $[\mathbf{x}_{1,n}, \mathbf{x}_{1,l}, \mathbf{x}_{2,n}, \mathbf{x}_{2,l}, ..., \mathbf{x}_{K,n}, \mathbf{x}_{K,l}]$ form the unknown vector $\mathbf{x}$ in the general model described by (4.1).

Example:

Consider the example of estimating the set of sinusoids in noise:

$$y_t = \sum_{k=1}^{K} \tilde{A}_k e^{i2\pi f_k t} + v_t, \qquad t = 1, 2, ..., d_y$$

where $\tilde{A}_k \in C$ is the complex amplitude of the k-th component whose frequency is $f_k \in [0, 1]$, and v_t is white Gaussian noise. The parameters to be estimated are $\mathbf{x} = [\tilde{A}_1, f_1, ..., \tilde{A}_K, f_K]$ of dimension$3K$. Assume that we know the total number of sinusoids K. The problem can be easily decomposed as K subproblems and handled by K PMC samplers, where the k-th sampler will target the unknowns $\mathbf{x}_k = [\tilde{A}_k, f_k]$ of dimension 3. Note that the k-th frequency f_k constitutes a nonlinear parameter, and the k-th complex amplitude $\tilde{A}_k$ is conditionally linear parameters.

Assume that we know the total number of sinusoids K. The problem can be easily decomposed as K subproblems and handled by K MPMC samplers, where the k-th sampler targets the nonlinear parameter $\mathbf{x}_{k,n} = f_k$ of dimension 1.

We assign each unknown vector $\mathbf{x}_{k,n}$ an MPMC sampler, and use the MMSE criterion to estimate the corresponding marginalized linear unknowns $\mathbf{x}_{k,l}$. The sample generation/propagation and resampling step of each MPMC sampler can be implemented in the usual way. The proposed weight update is implemented by

$$\tilde{w}_{k,n,j}^{(m)} = \frac{p(\mathbf{x}_{k,n,j}^{(m)} | \tilde{\mathbf{x}}_{-k,n}, \mathbf{y})}{q_{k,n,j}^{(m)}(\mathbf{x}_{k,n,j}^{(m)})} \tag{5.8}$$

where $\tilde{\mathbf{x}}_{-k,n}$ represents the most recent estimated values of all nonlinear unknowns except $\mathbf{x}_{k,n}$, i.e.,

$$\tilde{\mathbf{x}}_{-k,n} = \tilde{\mathbf{x}}_n \setminus \tilde{\mathbf{x}}_{k,n} = [\tilde{\mathbf{x}}_{1,n}^\top, \tilde{\mathbf{x}}_{2,n}^\top, ..., \tilde{\mathbf{x}}_{k-1,n}^\top, \tilde{\mathbf{x}}_{k+1,n}^\top, ..., \tilde{\mathbf{x}}_{K,n}^\top]^\top$$

and

$$\tilde{\mathbf{x}}_{k,n} = \sum_{m=1}^{M} w_{k,n,j}^{(m)} \mathbf{x}_{k,n,j}^{(m)}. \tag{5.9}$$

If the estimate in equation (5.9) is not available yet, one can use

$$\tilde{\mathbf{x}}_{k,n} = \sum_{m=1}^{M} w_{k,n,j-1}^{(m)} \mathbf{x}_{k,n,j-1}^{(m)}. \tag{5.10}$$

The implementation order of each MPMC sampler is randomized at each iteration as before. Let j denote the iteration number, $j = 1, 2, ...$, let k denote the k-th MPMC sampler, and let m represent the index of the particle in each sampler, $m = 1, 2, ..., M$. At the j-th iteration the MultiMPMC method can be summarized as follows.

Algorithm for multiple MPMC:

`Step 1.` Randomly choose the order of MPMC samplers. Let the order be $l_1, l_2, ..., l_k, ..., l_K$.

`Step 2.` For the l_k-th MPMC sampler, choose the importance function $q_{l_k,n,j}^{(m)}(\mathbf{x}_{l_k,n,j}^{(m)})$, $m = 1, 2, ..., M$; and draw nonlinear samples $\mathbf{x}_{l_k,n,j}^{(m)}$ from $q_{l_k,n,j}^{(m)}(\mathbf{x}_{l_k,n,j}^{(m)})$;

`Step 3.` Compute the weights of the samples by

$$\tilde{w}_{l_k,n,j}^{(m)} = \frac{p(\mathbf{x}_{l_k,n,j}^{(m)} | \tilde{\mathbf{x}}_{-l_k,n}, \mathbf{y})}{q_{l_k,n,j}^{(m)}(\mathbf{x}_{l_k,n,j}^{(m)})}$$

`Step 4.` Normalize the weights according to

$$w_{l_k,n,j}^{(m)} = \frac{\tilde{w}_{l_k,n,j}^{(m)}}{\Sigma_{k=1}^{M} \tilde{w}_{l_k,n,j}^{(m)}}; \tag{5.11}$$

`Step 5.` Resample the samples according to their weights;

`Step 6.` Obtain the most recent estimated values for the l_k-th sampler by

$$\tilde{\mathbf{x}}_{k,n} = \sum_{m=1}^{M} w_{k,n,j}^{(m)} \mathbf{x}_{k,n,j}^{(m)}$$

and use the MMSE criterion to obtain the estimated values of the linear parameters $\tilde{\mathbf{x}}_{k,l}$;

`Step 7.` If necessary, move to the $(j+1)$-th iteration when all K samplers finish the above steps.

`Step 8.` At the end, the estimate of the nonlinear parameters are

$$\tilde{\mathbf{x}}_n = [\tilde{\mathbf{x}}_{1,n}^\top, \tilde{\mathbf{x}}_{2,n}^\top, ..., \tilde{\mathbf{x}}_{K,n}^\top]^\top$$

and the estimate of the linear parameters $\tilde{\mathbf{x}}_l$ are obtained by the MMSE method. Note that all available weighted samples (from all samplers and all iterations) are used to approximate the posterior $p(\mathbf{x}|\mathbf{y})$.

CHAPTER 6

Population Monte Carlo a la Gibbs sampling

The key principle for constructing the approximations with any of the previously described PMC algorithms is importance sampling, which is a technique for estimating properties of a particular distribution with samples generated from a different distribution.

As with every method that uses importance sampling, the crucial factor for good performance of the method is the choice of generating functions of the particles. We propose that the generating functions be *alternating conditionals*, thereby mimicking the idea behind Gibbs sampling [31]. With this approach, it is expected, that one can generate particles in high dimensions more efficiently. Figure 6.1 shows a simple example of generation of a two-dimensional particle C from particle A with standard sampling and Gibbs sampling. The traditional way samples in a two dimensional space, and it sample particle C from A directly with the knowledge of the joint distribution. The Gibbs approach samples with known constraints (usually from the marginal distributions of each dimension of the unknowns) alternatingly, first sampling particle B from A and then sampling particle C from B. We expect to get samples of better "quality" with this proposed Gibbs PMC sampler.

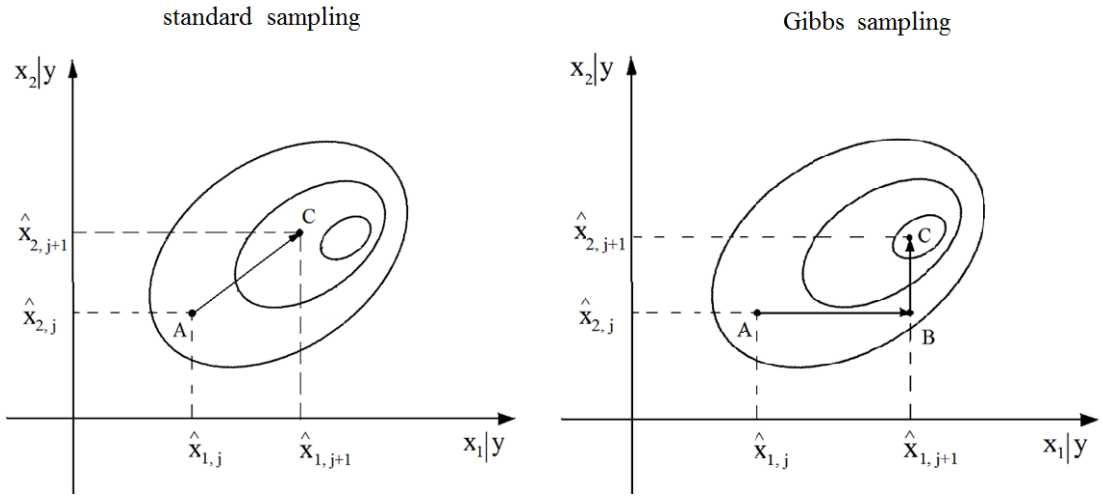


Figure 6.1: Standard sampling vs Gibbs sampling.

Gibbs sampling is an algorithm for generation of particles that represent samples from the joint probability distribution of two or more unknowns [31]. The particles have equal weights and they approximate the joint distribution or are used for computing integrals under the joint distribution. Gibbs sampling belongs to the larger class of MCMC methods and is often used for Bayesian inference [30].

In MCMC methods, at iteration j, we generate M particles. The m-th particle of $\mathbf{x}$,

$$\mathbf{x}_j^{(m)} = [x_{1,j}^{(m)}, x_{2,j}^{(m)}, \cdots, x_{d_x,j}^{(m)}]^\top$$

is constructed as follows: we draw $x_{k,j}^{(m)} \sim q_j(x_k)$, $k = 1, 2, \cdots, d_x$, where $q(x_k)$ is a proposal function [30]. We either accept/reject these proposals individually in parallel MCMC sampling or we reject the complete $\mathbf{x}_j^{(m)}$ globally. Gibbs sampling is a special type of MCMC sampling where each $x_{k,j}^{(m)}$ is sampled from a conditional distribution $q(x_k|\mathbf{x}_{-k,*})$, where $\mathbf{x}_{-k,*}$ is the vector of all the parameters in $\mathbf{x}$ except for x_k and the remaining conditioning parameters are at their current values (i.e., we use the last drawn values for the conditioning parameters).

Irrespectively of which MCMC approach we use, we have issues with convergence assessment. This problem, however, can be completely put away if we introduce importance sampling. In other words, if a particle $\mathbf{x}^{(m)}$ is obtained from $q(\mathbf{x})$, but we want it to come from $p(\mathbf{x})$, then we need to assign the particle an importance weight given by

$$\tilde{w}^{(m)} = \frac{p(\mathbf{x}^{(m)})}{q(\mathbf{x}^{(m)})}. \tag{6.1}$$

The weights and the particles form a random measure, $\chi = \{\mathbf{x}^{(m)}, w^{(m)}\}_{m=1}^{M}$, where the $w^{(m)}$s are normalized weights and M denotes the total number of samples. Here we reiterate that in PMC, we implement the generation of particles in iterations. For example, in iteration one, we get the random measure χ_1, in iteration two, the random measure χ_2 and so on. The objective is that, as we proceed with iterations, we improve the accuracy of the approximation. To that end, for obtaining better generating functions, one can use the approximations from the previous

iterations. One way of exploiting the previous iteration is to employ resampling (another operation that is common in particle filtering) [18]. That is, we construct new generating functions by using particles from the previous iteration that are picked based on their weights.

Here we propose a general approach for constructing generating functions for the PMC method [20]. We draw the particles of particular unknowns from a conditional distribution, where the conditioning is on the remaining unknowns. We basically mimic the Gibbs sampling idea, where as explained, we replicate the same steps except that our conditionals are not obtained from the target distribution. Note that in PMC, we assume that we *cannot* generate from the conditionals of the target distribution, and therefore we work with a different joint distribution, but one that allows for easy drawing of particles.

We now describe the specific steps of the proposed scheme. In iteration $j = 0$, we initialize the particle streams by drawing them from the prior $\pi(\mathbf{x})$. We draw M particles, and to each of them we assign the weights according to

$$\tilde{w}_0^{(m)} = p(\mathbf{y}|\mathbf{x}_0^{(m)}). \tag{6.2}$$

We assume now that at iteration $j-1$, we have the particles and the weights $\chi_{j-1} = \{\mathbf{x}_{j-1}^{(m)}, w_{j-1}^{(m)}\}_{m=1}^{M}$. We also recall that $\mathbf{x}_{j-1}^{(m)} = [x_{1,j-1}^{(m)}, x_{2,j-1}^{(m)}, \cdots, x_{d_x,j-1}^{(m)}]^\top$. The particles in the $j-$th iteration are obtained as follows:

Algorithm for Gibbs PMC:

`Step 1.` Randomly choose the order of generation of the parameters in $\mathbf{x}_j^{(m)} = [x_{1,j}^{(m)}, x_{2,j}^{(m)}, \cdots, x_{d_x,j}^{(m)}]^\top$. Let the order be $l_1, l_2, \cdots, l_{d_x}$.

`Step 2.` For $m = 1, 2, \cdots, M$, choose a particle for conditioning based on the normalized weights of the particles. Let the selected particle be with index λ_m. Then generate new

particles according to

$$x_{l_1,j}^{(m)} \sim q_{l_1,j}\left(x_{l_1}|x_{l_2,j-1}^{(\lambda_m)}, x_{l_3,j-1}^{(\lambda_m)}, \cdots, x_{l_{d_x},j-1}^{(\lambda_m)}\right)$$

for $n = 2, 3, \cdots, d_x - 1$,

$$x_{l_n,j}^{(m)} \sim q_{l_n,j}\left(x_{l_n}|x_{l_1,j}^{(m)}, \cdots x_{l_{n-1},j}^{(m)}, \ \ x_{l_{n+1},j-1}^{(\lambda_m)} \cdots, x_{l_{d_x},j-1}^{(\lambda_m)}\right)$$

and

$$x_{l_{d_x},j}^{(m)} \sim q_{l_{d_x},j}\left(x_{l_{d_x}}|x_{l_1,j}^{(m)}, x_{l_2,j}^{(m)}, \cdots, x_{l_{d_x-1},j}^{(m)}\right).$$

`Step 3.` Compute the weights of the samples by

$$\tilde{w}_j^{(m)} = \frac{p\left(\mathbf{y}|\mathbf{x}_j^{(m)}\right)}{\prod_{n=1}^{d_x} q_{l_n}(x_{l_n}^{(m)})}.$$

`Step 4.` Normalize the weights according to

$$w_j^{(m)} = \frac{\tilde{w}_j^{(m)}}{\Sigma_{k=1}^{M}\tilde{w}_j^{(k)}}; \tag{6.3}$$

`Step 5.` Resample the samples according to their weights;

`Step 6.` If more iterations are needed, set $j = j + 1$, and go back to step 1.

`Step 7.` At the end, obtain the estimated values from all available samples by

$$\tilde{\mathbf{x}} = \frac{\sum_{j=1}^{J}\sum_{m=1}^{M} w_j^{(m)}\mathbf{x}_j^{(m)}}{\sum_{j=1}^{J}\sum_{m=1}^{M} w_j^{(m)}}.$$

Note that all available weighted samples (from all samplers and all iterations) are used to approximate the posterior $p(\mathbf{x}|\mathbf{y})$.

The computed weights are stored as they were obtained by the last expression. They are normalized at the end of the algorithm with the weights of the particles from all the iterations in order to get the best possible approximation of the distribution of interest. However, the weights from Step 3 are also separately normalized before restarting the next iteration, so that one can use the normalized weights . The method can stop at any iteration.

We note that if we cannot generate $\mathbf{x}_0^{(m)}$ from $\pi(\mathbf{x})$, we can use a convenient generating function $q(\mathbf{x})$, and therefore the initial weights of the particles are

$$\tilde{w}_0^{(m)} = \frac{p(\mathbf{y}|\mathbf{x}_0^{(m)})}{q(\mathbf{x}_0^{(m)})}. \tag{6.4}$$

Example:

Consider an example of estimating a bivariate normal distribution. We had 4 observations from

$$\mathbf{y} \sim \mathcal{N}(\mathbf{x}, \mathbf{C}) = \mathcal{N}\left(\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

The unknown parameter $\mathbf{x} = [\theta_1, \theta_2]^{\mathrm{T}} = [0, 0]^{\mathrm{T}}$, and $\rho = 0.8$ is known.

We applied the standard PMC and the Gibbs sampling to this problem. Both methods used 4 samples per iteration and 500 iterations. The initial 4 samples are point $(-1, -1)$, $(-1, 1)$, $(1, -1)$, and $(1, 1)$. The results are shown in Figure 6.2. As seen it took less iterations for the Gibbs sampler to reach the target area. The quality of samples generated by the Gibbs methods are better than the standard PMC.

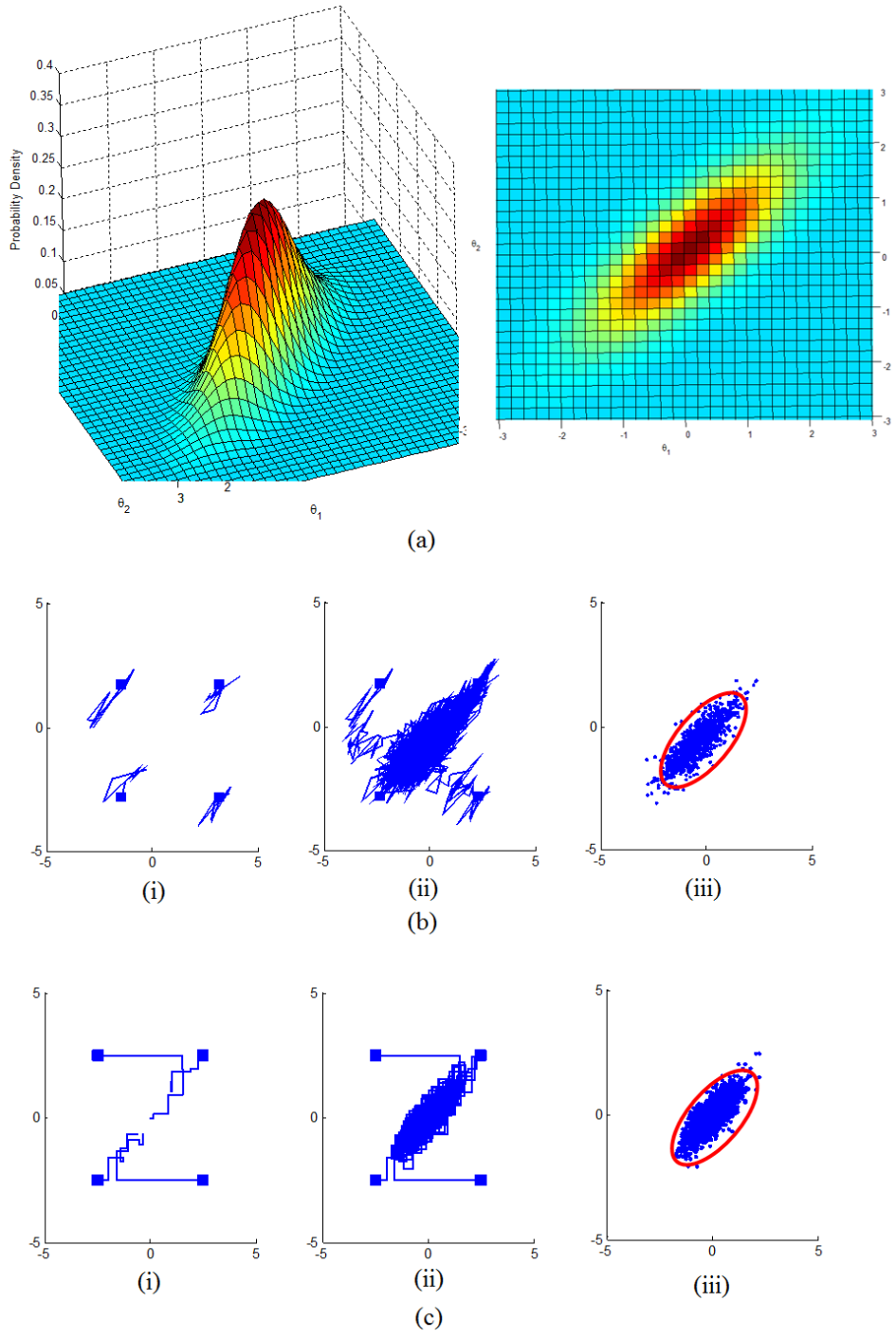


Figure 6.2: (a) The bivariate normal distribution; (b) PMC sampling: (i) first 10 iterations, (ii) 500 iterations, (iii) second half of samples; (c) Gibbs sampling: (i) first 10 iterations, (ii) 500 iterations, (iii) second half of samples.

CHAPTER 7

Numerical complexity

7.1 Summary

Consider the example of estimating a set of complex sinusoids in noise, where the problem is modeled as:

$$y_t = \sum_{k=1}^{K} \tilde{A}_k e^{i2\pi f_k t} + v_t, \qquad t = 1, 2, ..., d_y$$

where $\tilde{A}_k \in C$ is the complex amplitudes of the k-th component whose frequency is $f_k \in [0, 1]$, and v_t is white Gaussian noise. The parameters to be estimated are $\mathbf{x} = [\tilde{A}_1, f_1, ..., \tilde{A}_K, f_K]$ with dimension $3K$ (note that $\tilde{A}_k$ has a real and an imaginary component). Assume that we know the total number of sinusoids K. The problem can be easily decomposed as K subproblems and handled by K PMC samplers, where the k-th sampler will target the unknowns $\mathbf{x}_k = [\tilde{A}_k, f_k]$ with dimension 3. Note that the k-th frequency f_k constitutes a nonlinear parameter, and the k-th complex amplitude $\tilde{A}_k$ is conditionally linear parameters.

algorithm	**PMC**	**MPMC**	**multiple PMC**	**multiple MPMC**
type of samplers used	PMC	MPMC	PMC	MPMC
number of samplers used	1	1	K	K
dimension of space	3K	K	3	1
parameters for which samples are generated	$\mathbf{x} = [\tilde{A}_1, f1,$ $..., \tilde{A}_K, f_K]$	$\mathbf{x}_n$ $= [f1, ..., f_K]$	$\mathbf{x}_k$ $= [\tilde{A}_k, f_k]$	$\mathbf{x}_{k,n}$ $= f_k$
summary	1 sampler 3K parameters	1 sampler K parameters	K sampler 3 parameters	K sampler 1 parameters

Table 7.1: Dimension of parameter space explored by variant PMC algorithms.

The dimension of the space of unknowns considered by each of the discussed PMC algorithm is shown in Table 7.1. As seen, MPMC reduces the dimension of space from the original $3K$ to K. The distributed structure further brings down the dimension of the problem and therefore decrease the computational cost.

As discussed in Chapter 6, the proposed Gibbs PMC is intednded to generate better "quality" particles from the alternating conditionals, and therefore should further improve the efficiency of PMC algorithms in high dimensional applications.

7.2 Numerical complexity

The computational complexities of PMC, MPMC, MultiPMC and MultiMPMC are explored in this section with the example presented in Section 7.1. The number of floating-point operations (flops) for each algorithm are obtained from a theoretical point of view. A flop is defined as one addition, subtraction, multiplication, or division of two floating-point numbers. However, there are certain operations in the algorithms that can not be simply or easily measured in flops, for example, the cost of generating random numbers, resampling samples, and calculating nonlinear functions. However, it is still possible to evaluate the absolute time that each algorithm requires. For instance, the computational complexities of generating samples and resampling are proportional to the number of samples used for that purpose. These will be referred to as the equivalent flop complexity [43].

Table 7.2: Complexity of PMC

step	flops/EF
choose IF	$d_x(M-5)+dx\cdot M\cdot c_2$
sample generation	$dx\cdot M\cdot c_3$
weight computation	$3c_1+c_5+2c_8+d_y^3+4d_y^2+2d_y+d_x^3+4d_x^2+6d_x$
weight normalization	$2M-1$
resampling	$M\cdot\log M$

Table 7.3: Complexity of MPMC

step	flops/EF
choose IF	$d_{x_n}(M-5)+d_{x_n}\cdot M\cdot c_2$
sample generation	$d_{x_n}\cdot M\cdot c_3$
calculation of $\mathbf{x}_l$ using MMSE	$M\cdot c_6$
weight computation	$M\cdot[3(4d_{x_n}+c_1)+2+c_5]$
weight normalization	$2M-1$
resampling	$M\cdot\log M$

Table 7.4: Operations involved in the PMC and MPMC implementations

EF	computation	notes
c_1	$exp(\cdot)$	nonlinear function
c_2	$\cdot\sim\mathcal{U}[a,b]$	generating random number
c_3	$\cdot\sim\mathcal{N}(\mu,\sigma^2)$	generating random number
c_4	comparison	nonlinear function
c_5	$h(\cdot)$	nonlinear function in equation (2.2) and (4.1)
c_6	MMSE	nonlinear function
c_7	$A(\cdot)$	nonlinear function in equation (4.1)
c_8	$\sqrt{\cdot}$	nonlinear function

The complexity from a single iteration of PMC algorithm is shown in Table 7.2. The computational costs of the general and main steps are calculated based on a problem as in (2.2):

$$\mathbf{y}=h(\mathbf{x})+\mathbf{v}$$

with d_y observations and d_x unknowns and employing M samples/particles at each iteration. The complexity from a single iteration of MPMC algorithm is shown in Table 7.3. The computational costs are evaluated based on an equivalent problem as in (4.1) with d_y observations, d_{x_n} nonlinear unknowns, d_{x_l} linear unknowns, and M samples/particles at each iteration. The equivalent flops c_i, $i=1,\ 2,\ ...,\ 8$ are explained in Table 7.4. For the sinusoid example above,the estimated coefficients are $c1=1$, $c2=2$, $c3=5$, $c4=1$, $c5=dy(K+3+c1)$, $c6=30$, $c7=dy(K+1)$, and $c8=1$ (on a Dell Optiplex 360 with 32-bit operating system,

Inter dual 2.40-GHz CPU, and 4-GB memory). Note that, the complexity of MultiPMC and MultiMPMC can be easily evaluated when treated as several PMCs and MPMCs, respectively.

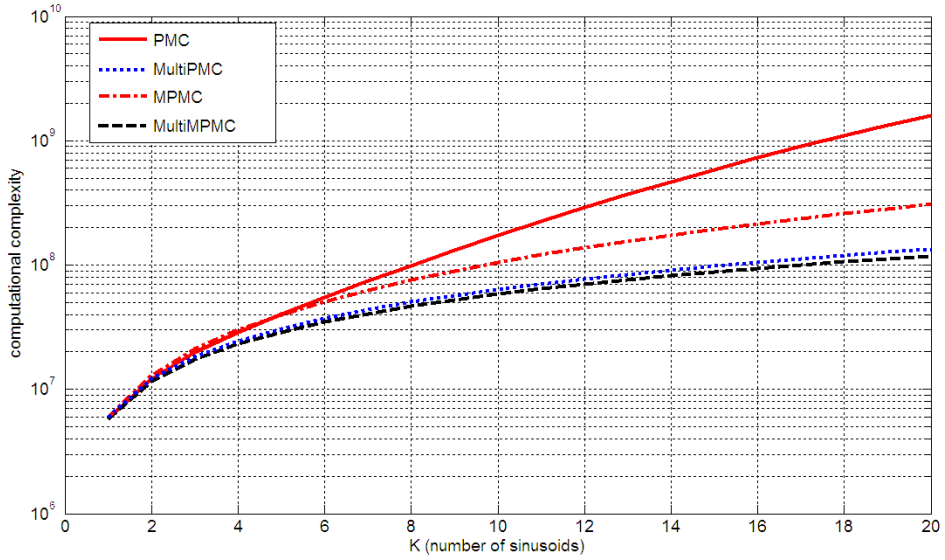


Figure 7.1: Computational complexity vs number of sinusoids.

The numerical complexities of standard PMC, MPMC, multiple PMC and multiple MPMC algorithms from one iteration are shown in Figure 7.1. The analyzed problem consists of estimating frequencies, amplitudes and phases of K sinusoids. We used $d_y = 25$, $d_x = 3K$, and for MPMC $d_{x_n} = K$ and $d_{x_l} = 2K$. Each sampler had a total of $200 \cdot K$ samples/particles at each iteration.

As shown in the figure, as the dimension of the unknowns increases, the complexity is decreased when the linear parameters are marginalized. In other words, for a given computational complexity cost, more samples can be used in MPMC than in the general PMC. By marginalizing the linear parameters, we can choose either to decrease the complexity and therefore increase the efficiency of the algorithm, or increase the number of samples and therefore increase the accuracy of the methods.

The advantages of MultiPMC and MultiMPMC are also shown is Figure 7.1. For the standard PMC method, the computational complexity increase exponentially as the number of

dimensions goes up. The MPMC method is more efficient as it marginalizes the nonlinear parameter and decreases dimension of the space of unknowns. Moreover, the distributed structures partition the problem into smaller dimensional subproblems, and breaks the exponential increase in complexity into linear like increase.

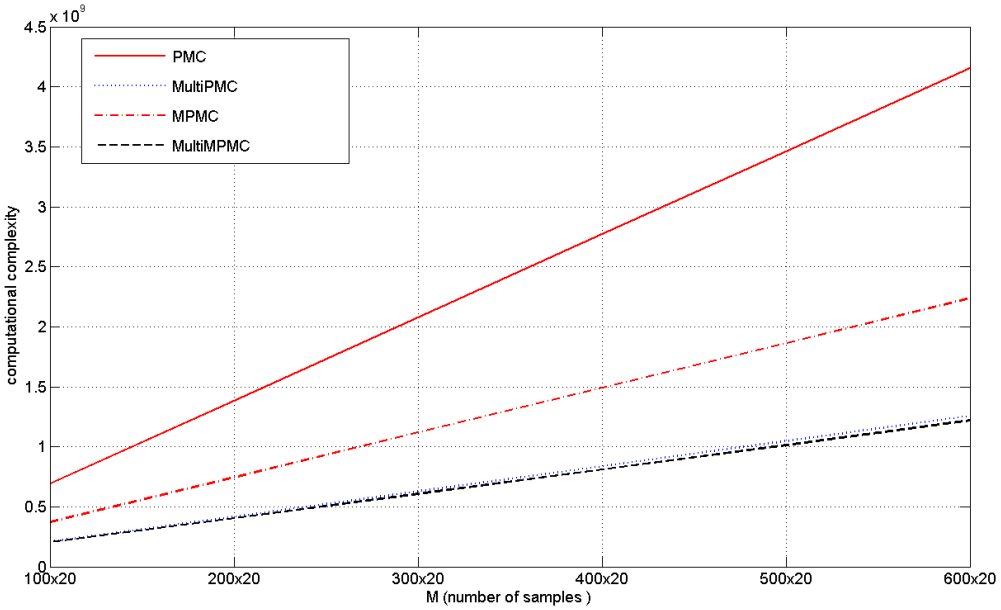


Figure 7.2: Computational complexity vs number of samples.

The numerical complexities of the algorithms from one iteration vs. the number of the samples used are shown in Figure 7.2. We used $d_y = 25$, $K = 20$, $d_x = 3K$, and for MPMC $d_{x_n} = K$ and $d_{x_l} = 2K$. We used a total number of samples/particles from $100 \cdot K$ to $600 \cdot K$ for each sampler. As shown in the figure, as the number of the samples used increases, the complexity basically increases linearly for each sampler. From the figure we can conclude that MPMC is more computational efficient than the standard PMC sampler. The distributed structure has lowest computational cost. Note that, all the discussion in this chapter are based on the numerical computation complexity. In the next chapter we will show by simulation that the multiple MPMC can be both efficient and accurate at the same time.

CHAPTER 8

Computer simulations

The proposed methods were tested on the problem of frequency estimation of multiple sinusoids. Simulation results showed the accuracy of the estimates and the feasibility of the methods. The new proposed PMC algorithms showed remarkable improvement with respect to other conventional approaches.

8.1 Example I: Estimation of frequencies of multiple real sinusoids

A problem of frequency estimation of real sinusoids in noise is demonstrated in this section. The PMC and MPMC algorithms were applied to solve the problem. Both methods were compared regarding the performance and the complexity.

8.1.1 The problem

The model considered for this first experiment follows the expression

$$\mathbf{y} = h(\mathbf{x}) + \mathbf{v} = \mathbf{A}(\mathbf{x}_n)\mathbf{x}_l + \mathbf{v} \tag{8.1}$$

where the observation vector is $\mathbf{y} \in R^{d_y \times 1}$ and the unknown parameter vector is $\mathbf{x} \in R^{d_x \times 1}$. The vector $\mathbf{x}$ is composed of nonlinear parameters $\mathbf{x}_n \in R^{d_{x_n} \times 1}$ and linear parameters $\mathbf{x}_l \in R^{d_{x_l} \times 1}$, where $d_x = d_{x_n} + d_{x_l}$, and the prior density of $\mathbf{x}$ is given by $p(\mathbf{x}_n, \mathbf{x}_l)$. As in (2.2), $h(\cdot)$ is a nonlinear function of the parameters $\mathbf{x}_n$; $A(\mathbf{x}_n)$ is a matrix of functions of the nonlinear parameter $\mathbf{x}_n$ and has dimension $d_y \times d_{x_l}$; and $\mathbf{v}$ is a $d_y \times 1$ noise vector with a known probability distribution $p(\mathbf{v})$.

More specifically, the above model was written as

$$\begin{aligned} y_t &= \sum_{k=1}^{K} a_k \cos(2\pi f_k t) + b_k \sin(2\pi f_k t) + v_t \\ t &= 1, 2, ..., d_y \end{aligned}$$

where a_k and b_k were the amplitudes of the cosine and sine components, respectively, of the k-th sinusoid whose frequency was f_k; and v_t was a white Gaussian noise.

The parameters to be estimated are $\mathbf{x} = [\mathbf{x}_l; \mathbf{x}_n]$, where

$$\mathbf{x}_l = [a_1, b_1, a_2, b_2, ..., a_K, b_K]^T$$
$$\mathbf{x}_n = [f_1, f_2, ..., f_K]^T.$$

The prior of the amplitudes and the frequencies are considered independent, i.e.,

$$p(\mathbf{x}) = p(\mathbf{x}_l, \mathbf{x}_n) = p(\mathbf{x}_l)p(\mathbf{x}_n).$$

For the prior of the frequencies, a constant over the region $0 < f_1 < f_2 < ... < f_K < 0.5$ was adopted. A zero-mean Gaussian distribution was proposed as the prior for the amplitudes.

In the experiment, we had $d_y = 20$ observations with $K = 2$ sinusoids generated according to the described model. The state $\mathbf{x}$ had six dimensions

$$\begin{aligned} \mathbf{x} &= [a_1, b_1, a_2, b_2, f_1, f_2] \\ &= [2\cos(\pi/6), 2\sin(\pi/6), 2\cos(-\pi/5), 2\sin(-\pi/5), 0.24, 0.24 + \delta] \end{aligned}$$

where $\delta = \frac{1}{2d_y}$. Note that δ represented the difference between two frequencies and was two times smaller than the resolution of the classical periodogram. The priors used for amplitudes were zero-mean Gaussian distributions

$$a_1, b_1, a_2, b_2 \sim \mathcal{N}(0, 5).$$

The priors for frequency were uniform distributions

$$f_1, f_2 \sim \mathcal{U}[0, 0.5] \text{ with constraint } f_1 < f_2.$$

The noise was drawn from the distribution

$$v \sim \mathcal{N}(0, \sigma_v),$$

and the value of the variance was defined by using the Signal-to-Noise Ratio (SNR) as follows

$$SNR = \frac{A^2}{2\sigma^2},$$

where SNR was measured in dB, A was the amplitude of the sinusoid, and σ^2 was the noise power.

8.1.2 Validation of PMC sampler

An amount of $M = 1,000$ particles were generated from a initial set of IFs. The initial proposed IF for the amplitudes were zero-mean Gaussian distributions, with predetermined variance vector $\mathbf{v} = 2 \times [1^2, 0.1^2, 0.01^2, 0.001^2, 0.0001^2]$. The initial IFs for the frequencies had preselected means at the maximum value of the fast Fourier transformation (FFT) of observation $\mathbf{y}$, and predetermined variance vector as $\mathbf{v} = 0.2 \times [1^2, 0.1^2, 0.01^2, 0.001^2, 0.0001^2]$. At the initial step, for each parameter, amplitude or frequency, a variance from the available five was assigned randomly to a particle with probability of $\frac{1}{5}$. After each iteration, the weights of the available variances for each parameter were updated separately according to the performance of the particles as stated in 3.3. Updated IFs had means located at the previous particles and variance coming from the predetermined variance vector with updated weights. In order to keep every variance valid after each iteration, re-scaling was employed to ensure that the minimum weight for a single variance was 0.05.

Two runs of results are shown in Figure 8.1 and Figure 8.2, one with *good* initial IFs for the frequency components and the other with *bad* initial IFs. The true values of states were

$$\mathbf{x} = [a_1, b_1, a_2, b_2, f_1, f_2] = [1.7321, 1, 1.6180, -1.1756, 0.24, 0.265].$$

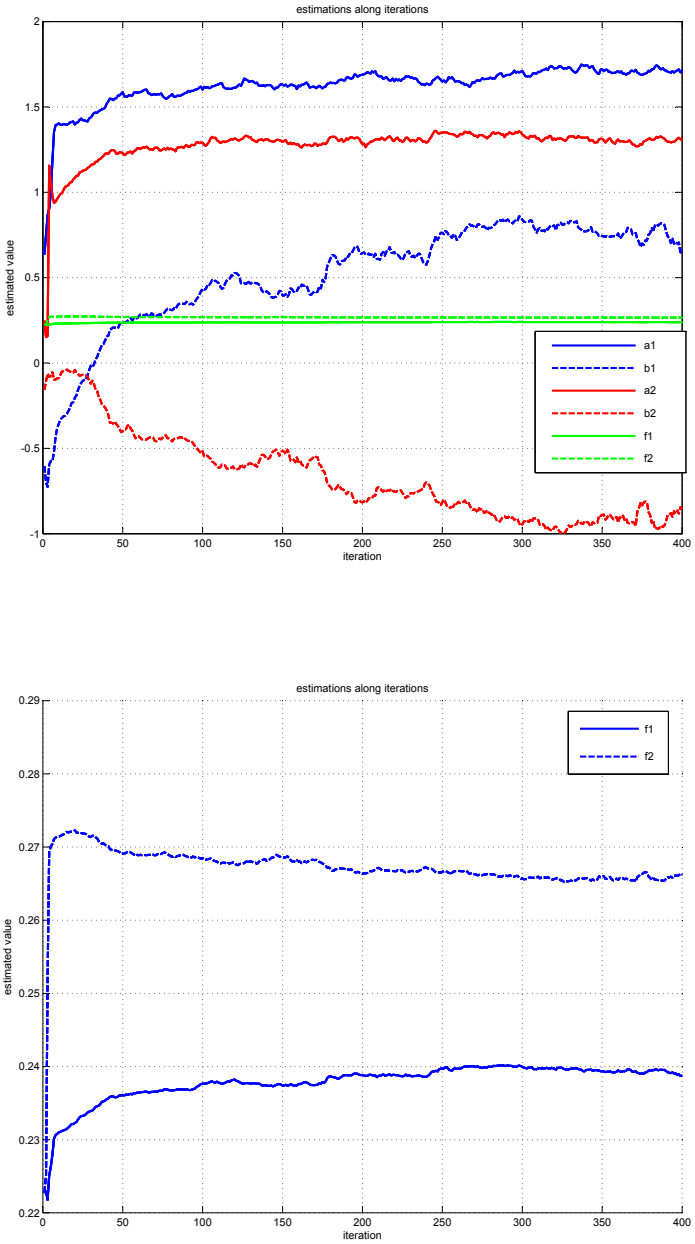


Figure 8.1: Example 1: Estimates vs iterations using the PMC algorithm with *good* initial IFs.

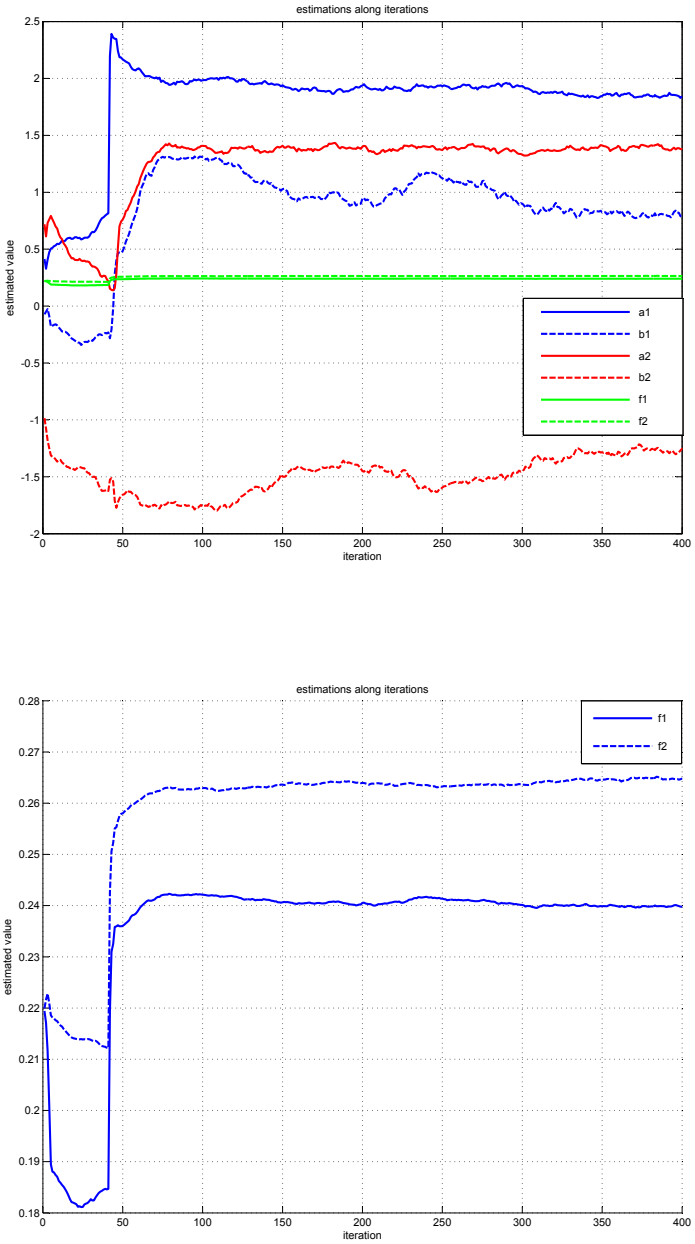


Figure 8.2: Example 2: Estimates vs iterations using the PMC algorithm with *bad* initial IFs.

As seen, PMC sampler is robust and is able to move the samples towards the true value of the unknowns with *good* or *bad* initial IFs. However, it takes up to 50 iterations to bring the samples close to the true values of the unknowns and therefore the efficiency needs to be improved.

8.1.3 Validation of MPMC sampler

The unknown parameters were separated into linear and nonlinear components

$$\mathbf{x}_l = [a_1, b_1, a_2, b_2] = [2\cos(\pi/6), 2\sin(\pi/6), 2\cos(-\pi/5), 2\sin(-\pi/5)],$$
$$\mathbf{x}_n = [f_1, f_2] = [0.24, 0.24 + \delta],$$

where $\delta = \frac{1}{2d_y}$.

As discussed in Section 4, solving the integral in Equation (4.3) is the key step for implementation of the MPMC algorithm. In this example, it can be readily shown that

$$p(\mathbf{x}_{n,j}^{(m)}|\mathbf{y}) \propto \frac{exp(-\frac{1}{2}\mathbf{y}^T(\mathbf{C}_v + \mathbf{A}_j\mathbf{C}_{x_l}\mathbf{A}_j^T)^{-1}\mathbf{y})}{|\mathbf{C}_v + \mathbf{A}_j\mathbf{C}_{x_l}\mathbf{A}_j^T|^{\frac{1}{2}}}, \tag{8.2}$$

where $\mathbf{C}_v = \sigma_v\mathbf{I}$, and $\mathbf{C}_{x_l} = 5\mathbf{I}$.

At each iteration in MPMC, samples of nonlinear parameters were generated according to the IF, and then linear parameters were determined via MMSE [45]. An amount of $M = 1,000$ particles for the nonlinear parameters were generated with initial Gaussian distributions, where the means were preselected at the maximum frequency of the FFT of the observations $\mathbf{y}$, and the frequency predetermined variance vector was $\mathbf{v} = 0.2\times[1^2, 0.1^2, 0.01^2, 0.001^2, 0.0001^2]$. At the initial step, for each nonlinear/frequency parameter, a variance from the available five was assigned randomly to a particle with probability of $\frac{1}{5}$, as stated in 3.3. After each iteration, the weights of available variances for each parameter were updated separately according to the performance of the particles. Updated importance functions had means located at the previous particles and variance coming from the predetermined variance vector with updated weights. In order to keep every variance valid after each iteration, re-scaling was employed to ensure that the minimum weight for a single variance was 0.05.

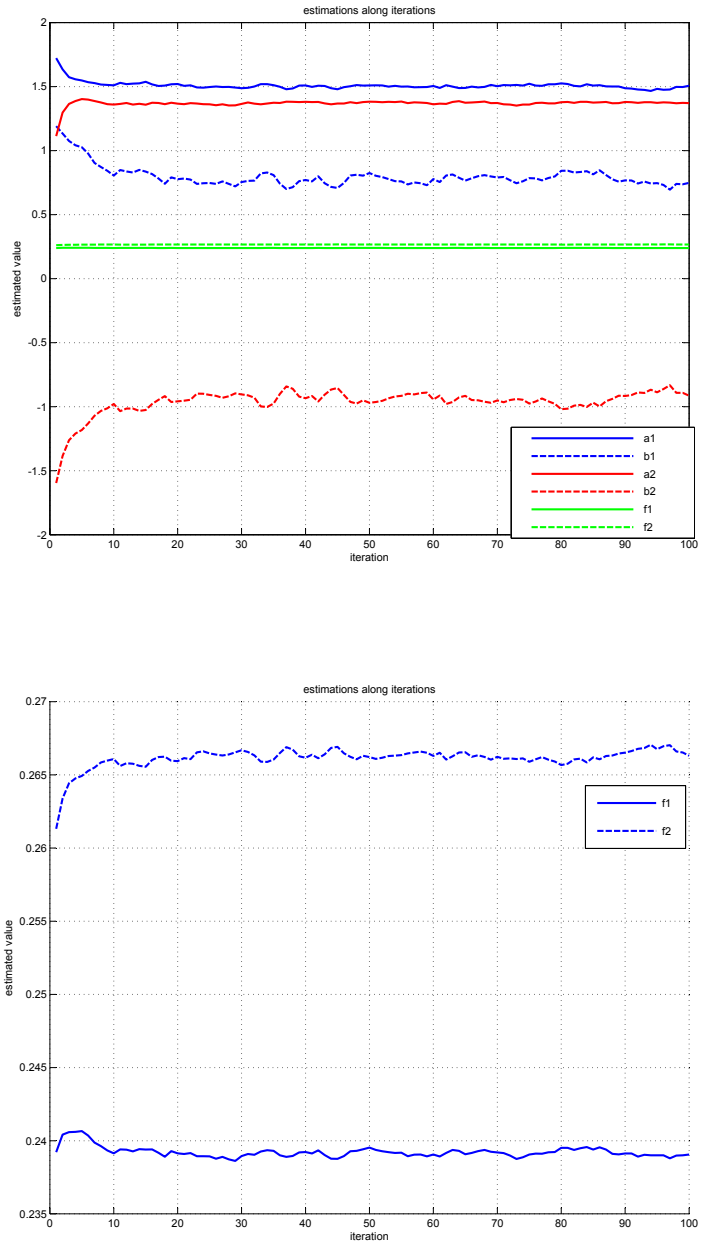


Figure 8.3: Example 3: Estimates vs iterations using the MPMC algorithm with *good* initial IFs..

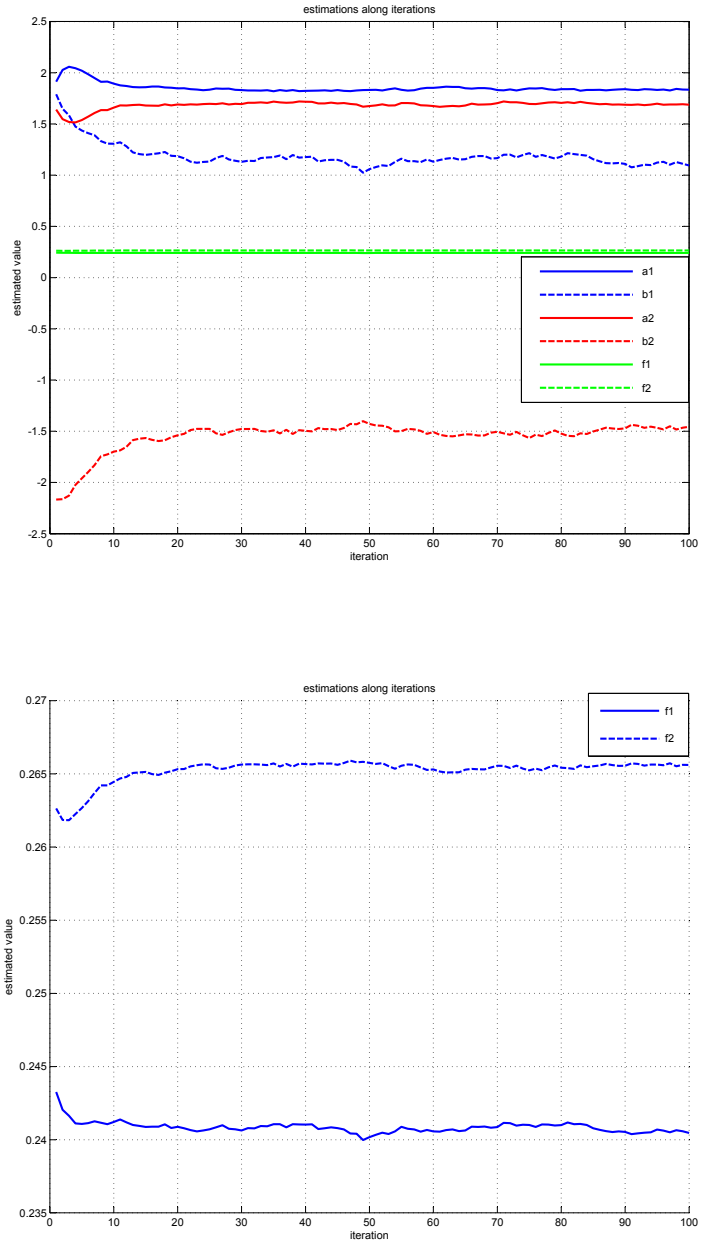


Figure 8.4: Example 4: Estimates vs iterations using the MPMC algorithm with *bad* initial IFs..

The performance of two realization of MPMC sampler with *good* and *bad* initial IFs are shown in Figure 8.3 and Figure 8.4. The true values of states were

$$\mathbf{x}_l = [a_1, b_1, a_2, b_2] = [1.7321, 1, 1.6180, -1.1756],$$
$$\mathbf{x}_n = [f_1, f_2] = [0.24, 0.265].$$

MPMC sampler is both robust and efficient. It moves the samples towards the true values within only a few iterations regardless the proposals, much faster than the PMC. It required less computations as it marginalized two-third of the unknowns. Furthermore, the quantified comparisons between these two algorithms are presented in the next section.

8.1.4 Comparisons between PMC and MPMC sampler

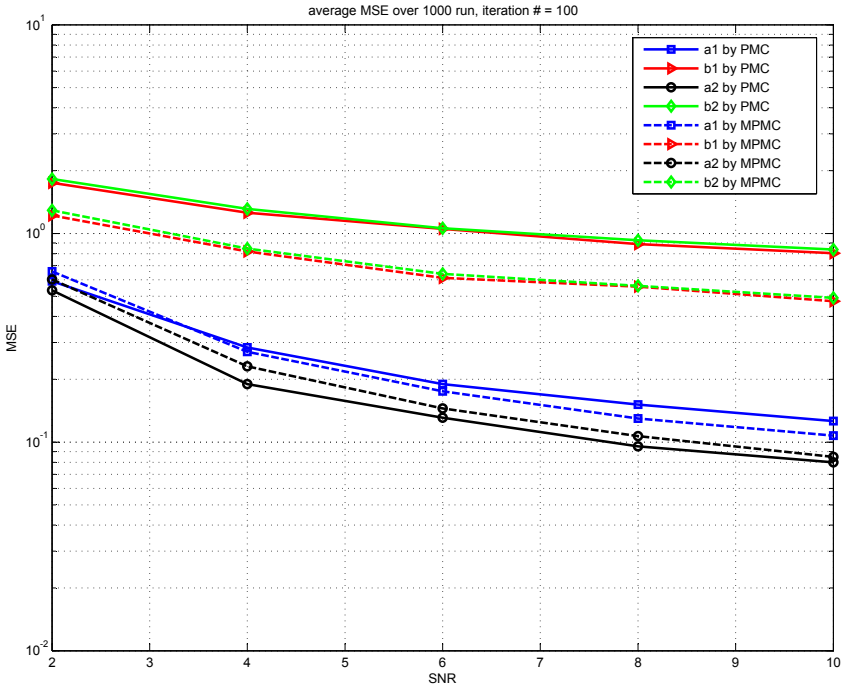
The performance of the algorithms to this problem was quantified by using the mean square error (MSE) of the parameters to be estimated, which was calculated as

$$MSE = \frac{1}{K}\sum_{k=1}^{K}(\hat{\mathbf{x}}^k - \mathbf{x})^2, \tag{8.3}$$

where k represented the k-th run of the algorithm, $\hat{\mathbf{x}}^k$ denoted the estimates obtained in k-th run, and $\mathbf{x}$ was the true value of the state. All the MSE plots were averaged over $K = 10$ runs for both methods.

Figures 8.5 depicts the MSE vs SNR using both algorithms with 100 iterations, averaged over 1,000 runs. As seen in the Figure, with 100 iterations, the MPMC performs better when the SNR ranges from 2dB to 10dB. For the frequency parameters, the MPMC provide accurate estimates even at low SNRs. For example, MPMC has lower MSE with SNR=4dB than the PMC does at SNR=10dB.

All in all, the MPMC required less computations, which is an advantage in a high-dimensional system. The MPMC is more robust and starts with a set of initial estimates close to their real values. Besides, the MPMC has the advantage of better performance at lower SNRs.



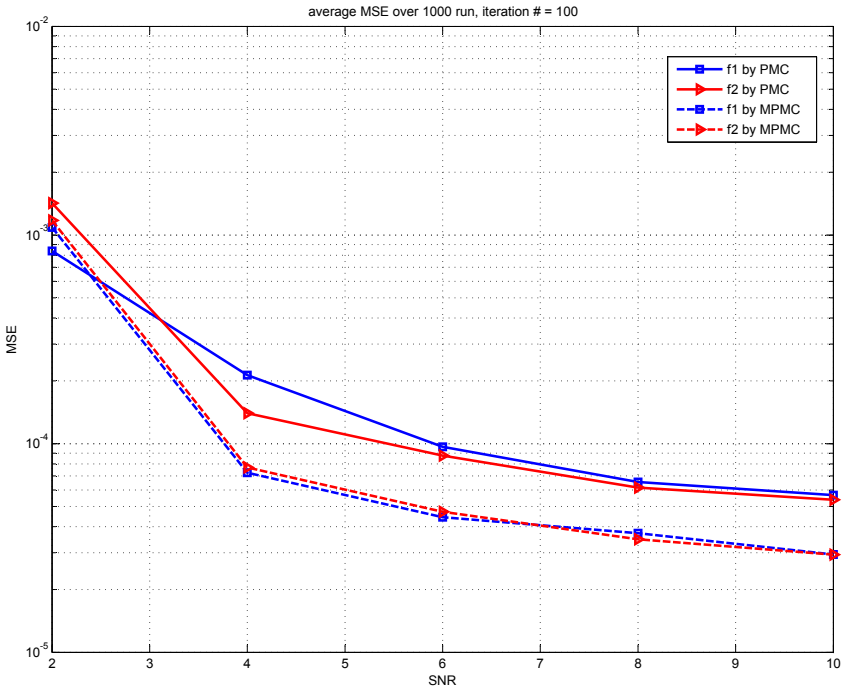


Figure 8.5: Comparison: MSE of estimates vs SNR using the PMC and MPMC algorithms for iteration=100, averaged over 1,000 runs.

8.2 Example II: Estimation of frequencies of one complex sinusoid

The PMC and MPMC algorithms were tested in a problem of frequency estimation of sinusoids in noise, and the results showed their convergence to the Cramér-Rao lower bound.

8.2.1 The problem

In this section, the model described in previous section was employed:

$$\mathbf{y} = h(\mathbf{x}) + \mathbf{v} \tag{8.4}$$

where the observation vector is $\mathbf{y} \in R^{d_y \times 1}$ and the unknown parameter vector is $\mathbf{x} \in R^{d_x \times 1}$. The vector $\mathbf{x}$ is composed of nonlinear parameters $\mathbf{x}_n \in R^{d_{x_n} \times 1}$ and linear parameters $\mathbf{x}_l \in R^{d_{x_l} \times 1}$, where $d_x = d_{x_n} + d_{x_l}$, and the prior density of $\mathbf{x}$ is given by $p(\mathbf{x}_n, \mathbf{x}_l)$. As in (2.2), $h(\cdot)$ is a nonlinear function of the parameters $\mathbf{x}_n$; $A(\mathbf{x}_n)$ is a matrix of functions of the nonlinear parameter $\mathbf{x}_n$ and has dimension $d_y \times d_{x_l}$; and $\mathbf{v}$ is a $d_y \times 1$ noise vector with a known probability distribution $p(\mathbf{v})$.

More specifically, in this section we consider

$$y_t = \tilde{A}_k e^{i(2\pi f_1 t)} + v_t, \quad t = 1, 2, ..., d_y \tag{8.5}$$

where $\tilde{A}_1 \in C$ was the complex amplitude of the frequency $f_1 \in [0, 1]$, and v_t was white Gaussian noise. The parameters to be estimated are $\mathbf{x} = [\tilde{A}_1, f_1]$ with dimension 3 (note that $\tilde{A}_1$ has a real and imaginary component).

8.2.2 Cramér-Rao lower bound

Consider the data model in Equation 8.5 in complex white Gaussian noise (CWGN). The parameter v_t is a CWGN process with variance σ^2. The parameter vector to be estimated is

$$\mathbf{x} = [\tilde{A}_1, f_1, ..., \tilde{A}_K, f_K]$$

and it can be rewritten as

$$\mathbf{x} = [A_1, \phi_1, f_1, ..., A_K, \phi_K, f_K]$$

note that all parameters in this vector are real and $\tilde{A}_k = A_k e^{i\phi_k}$. The Fisher information matrix is calculated as [45]

$$[\mathbf{I(x)}]_{ij} = \frac{2}{\sigma^2}[\sum_{t=1}^{d_y} \frac{\partial y_t^*}{\partial \mathbf{x}_i} \frac{\partial y_t}{\partial \mathbf{x}_j}] \tag{8.6}$$

where $[\mathbf{I(x)}]_{ij}$ is the (i, j) element of $[\mathbf{I}]$ matrix, $\mathbf{x}_i$ is the i_{th} parameter to be estimated in $\mathbf{x}$, and y_t^* is the conjugate transpose of y_t. The CRLB is found as the (i, i) position of the inverse of the Fisher information matrix

$$var(\hat{\mathbf{x}}_i) \geq [\mathbf{I}^{-1}(\mathbf{x})]_{ij}. \tag{8.7}$$

Take one frequency component with unknown amplitude and phase as an example, where

$$y_t = A_1 e^{i(2\pi f_1 t + \phi_1)} + v_t, \quad t = 1, 2, ..., d_y$$

and the parameter vector to be estimated is

$$\mathbf{x} = [A_1, \phi_1, f_1].$$

The partial derivatives are calculated as

$$\begin{aligned}
\frac{\partial y_t}{\partial \mathbf{x}_1} &= \tfrac{\partial y_t}{\partial A_1} &&= exp[j(2\pi f_1 t + \phi_1)] \\
\frac{\partial y_t}{\partial \mathbf{x}_2} &= \tfrac{\partial y_t}{\partial \phi_1} &&= jA \cdot exp[j(2\pi f_1 t + \phi_1)] \\
\frac{\partial y_t}{\partial \mathbf{x}_3} &= \tfrac{\partial y_t}{\partial f_1} &&= j2\pi t \cdot exp[j(2\pi f_1 t + \phi_1)].
\end{aligned}$$

The Fisher information matrix is

$$\mathbf{I(x)} = \frac{2}{\sigma^2} \begin{bmatrix} d_y & 0 & 0 \\ 0 & d_y \cdot A^2 & A^2 \sum_{t=1}^{d_y} 2\pi t \\ 0 & A^2 \sum_{t=1}^{d_y} 2\pi t & A^2 \sum_{t=1}^{d_y} (2\pi t)^2 \end{bmatrix}.$$

Upon inversion we have

$$\begin{aligned}
var(\hat{A}_1) &\geq \quad \tfrac{\sigma^2}{2d_y} \\
var(\hat{\phi}_1) &\geq \quad \tfrac{\sigma^2(2d_y-1)}{A^2 d_y(d_y+1)} \\
var(\hat{f}_1) &\geq \quad \tfrac{6\sigma^2}{(2\pi)^2 A^2 d_y(d_y^2-1)}.
\end{aligned}$$

Note that for complex data the SNR is defined as $SNR = \frac{A^2}{\sigma^2}$, and the bounds reduce to

$$\begin{aligned}
var(\hat{A}_1) &\geq \quad \tfrac{\sigma^2}{2d_y} \\
var(\hat{\phi}_1) &\geq \quad \tfrac{(2d_y-1)}{SNR \cdot d_y(d_y+1)} \\
var(\hat{f}_1) &\geq \quad \tfrac{6}{(2\pi)^2 \cdot SNR \cdot d_y(d_y^2-1)}.
\end{aligned}$$

8.2.3 Validation of PMC sampler with an example of one frequency with known amplitude and phase

In the first step experiment, we had $d_y = 20$ observations with $K = 1$ sinusoids generated according to the model described. In this step we treated $\tilde{A}_1 = 1$ as known parameters. The state **x** had only one dimension

$$\mathbf{x} = [f_1],$$

The complex noise were drawn from the distribution

$$v \sim \mathcal{N}(0, \sigma_v),$$

or equivalently

$$real(v) \sim \mathcal{N}(0, \tfrac{\sigma_v}{\sqrt{2}}), \text{ and } imag(v) \sim \mathcal{N}(0, \tfrac{\sigma_v}{\sqrt{2}}),$$

and the value of the variance was defined by using the SNR as follows

$$SNR = \tfrac{A^2}{{\sigma_v}^2},$$

where SNR was measured in dB, A was the amplitude of the sinusoid, and σ^2 was the noise power.

An amount of $M = 500$ particles were generated from an initial importance function. The initial IFs for frequency had preselected means at the maximum value of the FFT of observation **y**, and used multiple-kernel IFs as stated in Chapter 3. The predetermined variance vector was $\mathbf{v} = \sigma_0 \times [1^2, 0.1^2, 0.01^2, 0.001^2, 0.0001^2]$.

At the initial step, for the frequency, a variance from the available five was randomly assigned to a particle with probability of $\frac{1}{5}$. After each iteration, the weights of available variances for each parameter were updated separately according to the performance of the particles. Updated IFs had means located at the previous particles and variance coming from the predetermined variance vector with updated weights. In order to keep every variance valid after

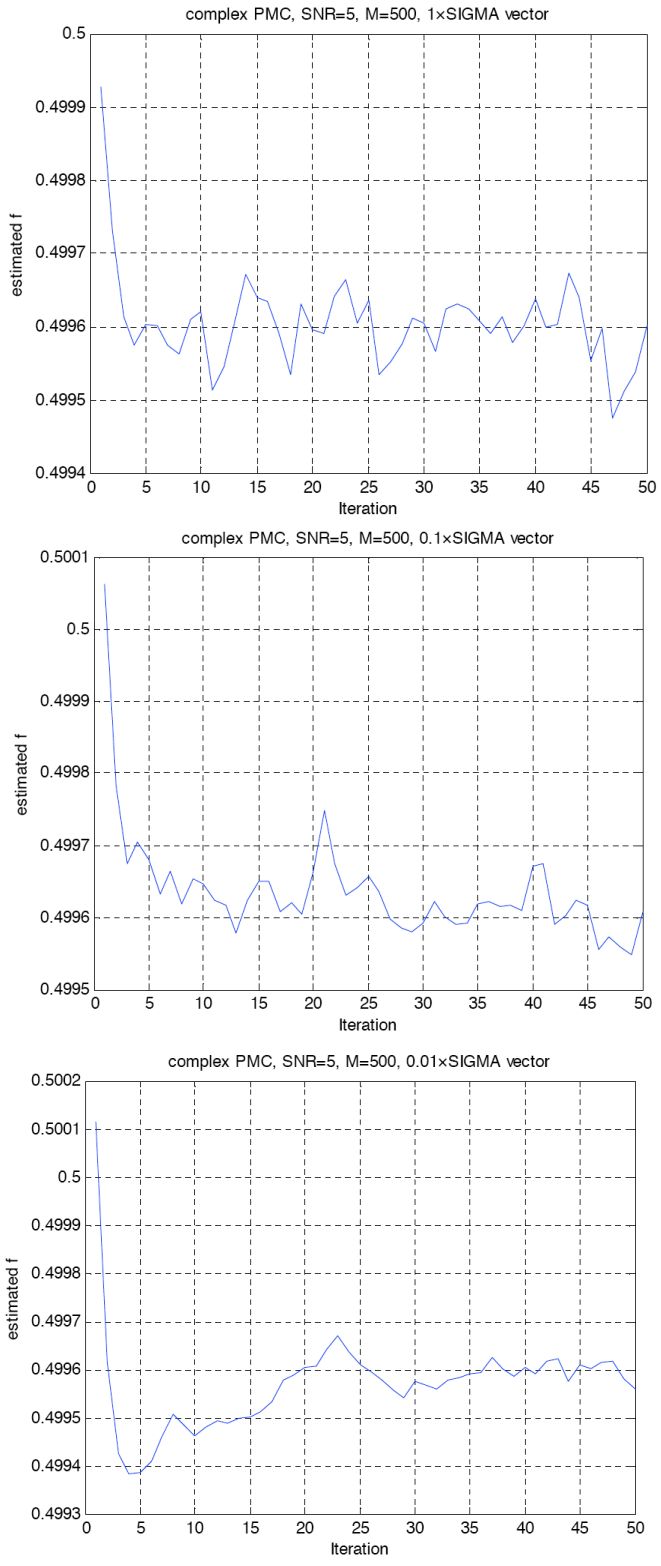


Figure 8.6: Estimated f with different predetermined σ_0 vectors.

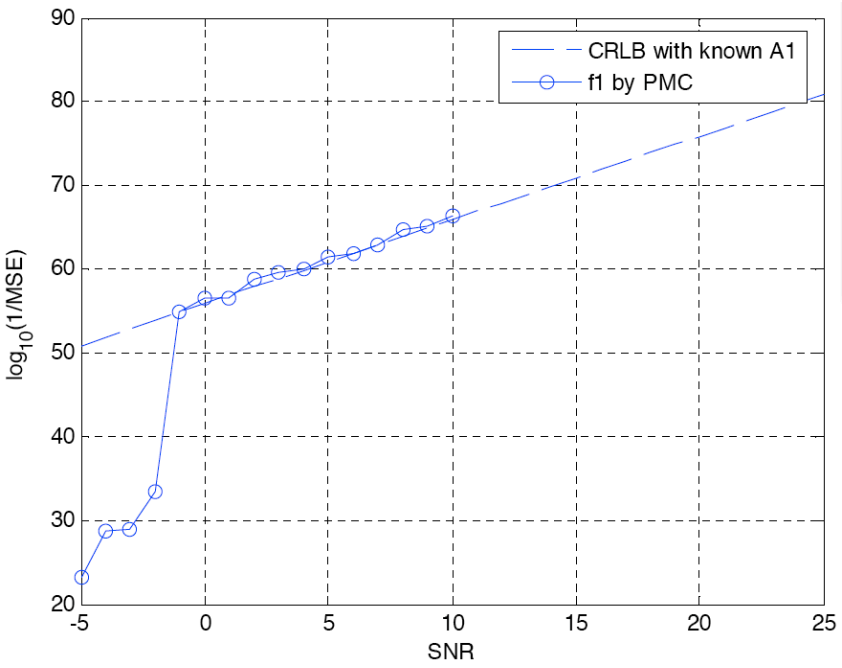


Figure 8.7: $10\log_{10}(1/MSE)$ vs. SNR for 1 fixed frequency estimated using the PMC algorithm.

each iteration, re-scaling was employed to ensure that the minimum weight for a each available variance was 0.05.

The parameter σ_0 for the predetermined variance vector was proposed as 1, 0.1, and 0.01, respectively. A set of estimation vs. iteration is shown in Figure 8.6. All three plots came from the same set of observations, where $f_1 = 0.5$, $d_y = 20$, $SNR = 5$. As seen, with larger σ_0 the estimation results vibrate and with smaller σ_0 the results curve moves smoother, however, the estimated results are within comparable level. For simplicity $\sigma_0 = 0.1$ is used for one-frequency estimation in this report. For the next steps, for multi-frequencies problem, σ_0 for each component might need to be determined with a more adaptive procedure. This might be explored as part of the future work.

8.2.3.1 Validation of PMC sampler with an example of one fixed frequency

Observations were generated by a fixed frequency value $f_1 = 0.5$, with complex amplitude $\tilde{A}_1 = 1$.More specifically

$$y_t = \tilde{A}_1 e^{i(2\pi f_1 t)} + v_t, \quad t = 1, 2, ..., d_y,$$

where in each run, $d_y = 20$ observations were employed.

The performance of the algorithm to this problem was quantified based on the MSE of the parameters to be estimated, which was calculated as in 8.17. The performance of PMC sampler in term of $10\log_{10}(1/MSE)$ is plotted along SNR and compared with the CRLB [45], shown as in Figure 8.7. All the points on the plot were averaged over $R = 500$ runs for each SNR from -5 to 10. In each run, PMC was applied using 500 particles and 50 iterations. It can be concluded from this plot that PMC performed well for solving this problem and achieved the CRLB for the range of $SNR \geq -1$.

8.2.3.2 Validation of PMC sampler with an example of one frequency from a uniform distribution

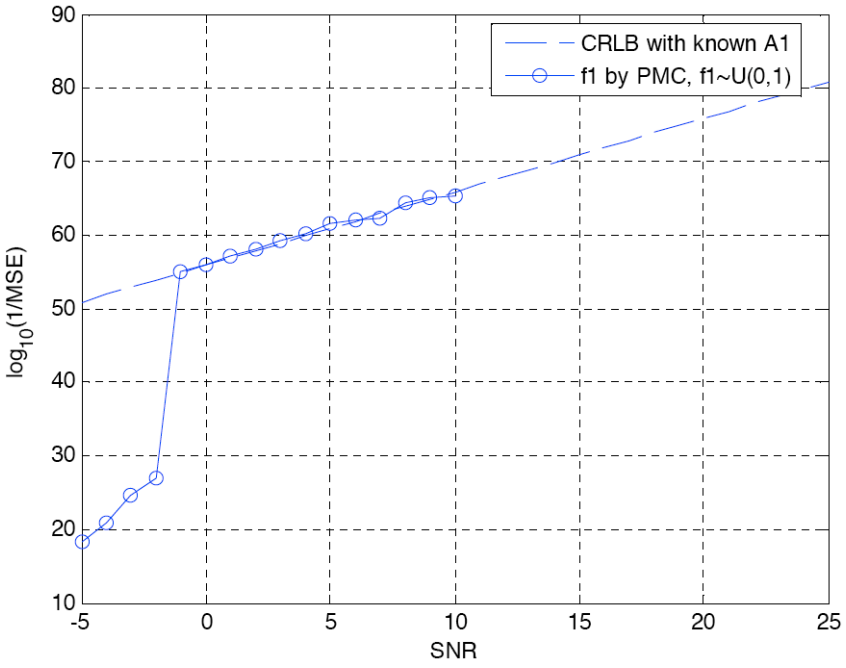


Figure 8.8: $10\log_{10}(1/MSE)$ vs. SNR for one frequency with uniform distribution estimated using the PMC algorithm.

The one-frequency problem with fixed frequency in the previous subsection is extended to frequency drawn from a given distribution in this part. The parameter f_1 is randomly selected from a uniform distribution as

$$f_1 \sim U[0,1]. \tag{8.8}$$

Then observations are obtained with this selected f_1 using

$$y_t = \tilde{A}_1 e^{i(2\pi f_1 t)} + v_t, \qquad t = 1, 2, ..., d_y$$

where the complex amplitude $\tilde{A}_1 = 1$, and observation number $d_y = 20$.

Figure 8.8 depicts the $10\log_{10}(1/MSE)$ of estimates by MPMC sampler vs SNR and compared with CRLB [45]. Each point on the plot was averaged over 500 runs, using 500 particles and 50 iterations for each run. In each single run, instead of using fixed f_1 to generate the observations, f_1 was drawn from $f_1 \sim U[0,1]$. It is shown that PMC performs well and achieves the CRLB for the range of $SNR \geq -1$.

Note that for a single frequency estimation problem, the CRLB is independent from the value of the frequency f_1. As shown in Figures 8.7 and 8.8, there is no difference between posterior CRLB (PCRLB) and CRLB for a single frequency with the same known amplitude and phase.

8.2.4 Comparison of PMC and MPMC sampler with an example of one frequency

Observations were generated by fixed frequency value $f_1 = 0.5$, with a complex amplitude $\tilde{A}_1 = 1$. More specifically

$$y_t = \tilde{A}_1 e^{i2\pi f_1 t} + v_t, \qquad t = 1, 2, ..., d_y,$$

where frequency f_1 was the nonlinear parameter, and complex amplitude $\tilde{A}_1$ was the linear parameter; in each run, $d_y = 20$ observations were employed.

PMC and MPMC were applied to estimate these two parameters f_1 and $\tilde{A}_1$, respectively. The prior for $\tilde{A}_1$ used in MPMC was

$$\tilde{A}_1 \sim \mathcal{N}(0, \sqrt{5}),$$

and $\tilde{A}_1$ was complex.

Equation 8.17 was employed to evaluate the performance of PMC and MPMC algorithms for this problem. Figure 8.9 shows the $10log_{10}(1/MSE)$ of estimates vs SNR for PMC and MPMC algorithms respectively. Each point on the plot was averaged over 500 runs, and for each run 500 particles and 50 iterations were used for estimation. As seen in the figures, with 50 iterations both PMC and MPMC perform well and achieve the CRLB [45] in the range from SNR of -1 to 10.

As shown in Figures 8.10, both the PMC and the MPMC converge to CRLB as the number of iterations increases. However, the PMC needs at least 20 iterations to get close to CRLB, while the MPMC only takes 5 iterations to achieve the bound. The main advantage of the MPMC over the PMC is the fast convergence to the estimates.

The MPMC also requires less computations in generation of samples which makes it more computational efficient when compared to the traditional PMC, as shown in Figures 8.11. For both the PMC and the MPMC, their performances converge to CRLB as the number of particles/samples increases. To estimate two complex unknowns, the PMC takes 300 particles to achieve the bound. While the MPMC marginalize the linear parameter $\tilde{A}_1$, it only needs 200 particles to hit on the bound. The MPMC requires less iterations and less samples/particles to converge to the CRLB. Therefore the MPMC has less computational complexity.

8.3 Example III: Estimation of frequencies of multiple complex sinusoids with unknown amplitude and phase

8.3.1 The problem

For this experiment the problem was modeled as

$$y_t = \sum_{k=1}^{K} \tilde{A}_k e^{i(2\pi f_k t)} + v_t, t = 1, 2, ..., d_y, \tag{8.9}$$

where $\tilde{A}_k \in C$ are the complex amplitudes of the k-th component whose frequency is $f_k \in [0, 1]$, and v_t is white Gaussian noise. The parameters to be estimated are $\mathbf{x} = [\tilde{A}_1, f_1, ..., \tilde{A}_K, f_K]$ with dimension $3K$. Assume that we know the total number of sinusoids K. The problem can be easily decomposed as K subproblems and handled by K PMC samplers, where the k-th sampler will target the unknowns $\mathbf{x}_k = [\tilde{A}_k, f_k]$ with dimension 3.

8.3.2 Comparison of PMC and MPMC sampler with an example of two frequencies

The problem of frequency estimation is considered in this section to compare the performance of the MPMC and other traditional approaches.

Observations were generated by two fixed frequency values $f_1 = 0.5$ and $f_2 = 0.52$, with complex amplitudes $\tilde{A}_1 = 1$ and $\tilde{A}_2 = 1e^{i\pi/4}$. More specifically

$$y_t = \tilde{A}_1 e^{i(2\pi f_1 t)} + \tilde{A}_2 e^{i(2\pi f_2 t)} + v_t, \quad t = 1, 2, ..., d_y,$$

where in each run, $d_y = 25$ observations were employed. Note that f_1 and f_2 were the nonlinear parameters, and $\tilde{A}_1$, and $\tilde{A}_2$ were the linear parameters.

The MPMC was applied to estimate the unknown parameters, where $\tilde{A}_1$, and $\tilde{A}_2$ were marginalized while f_1 and f_2 are estimated using PMC. The prior for $\tilde{A}_1$ and $\tilde{A}_2$ used are

$$A_1 e^{i\phi_1}, A_2 e^{i\phi_2} \sim \mathcal{N}(0, \sqrt{5}),$$

and A_1,ϕ_1, A_2 and ϕ_2 were complex.

The performance of the MPMC to this problem in terms of $10 \log_{10}(1/MSE)$ is shown in Figures 8.12. Each point was averaged over 500 runs of simulations. In each run, $M = 1,000$ samples and $J = 50$ iterations were used. As seen in the figure, the MPMC performed well and achieved the CRLB [45] in the range from SNR of 0 to 10. This result is comparable to a maximum likelihood estimator (MLE) and mean likelihood frequency estimator (MLLFE) stated in [46], shown in Figure 8.13.

The MPMC, the MLE and the MLLFE all achieved the CRLB. However, the computational complexity varies a lot. As mentioned in [46], using MLE for this problem needs a grid search for the nonlinear parameters which is difficult to implement and computational expensive. The MLLFE improves the efficiency by using a modified mean likelihood estimator. Though the MLLFE uses the concepts of importance sampling to obtain the mean likelihood estimate without the computation of multidimensional integrals. It still requires computation of two integrals and implementation of Golden search of one inverse of integral for each realization of the frequency. It requires generation of 2000 frequency realizations to achieve the CRLB for two sinusoids. The MPMC is computationally less expensive than the other two methods but performs as well.

8.3.3 Comparison of PMC, MPMC, multiple PMC, and multiple MPMC sampler with an example of three frequencies

The problem of frequency estimation of complex sinusoids in noise is considered in this section to show the performance of the distributed PMC algorithms. The model of the problem is

$$y_t = \sum_{k=1}^{K} \tilde{A}_k e^{i(2\pi f_k t)} + v_t, \quad t = 1, 2, ..., d_y, \tag{8.10}$$

where $0 < f_1 < f_2 < ... < f_K < 1$; $\tilde{A}_k$ are the complex amplitude of the k-th frequency component; and v_t is white complex Gaussian noise. The parameters to be estimated are $\mathbf{x} = [\tilde{A}_1, f_1, ..., \tilde{A}_K, f_K]$, and therefore the space of unknowns has dimension $3K$. Note that f_k denoted the k-th frequency and constituted a nonlinear parameter, and $\tilde{A}_k$ were the k-th complex amplitude and were conditionally linear parameters.

The complex noise was drawn from the distribution

$$v_t \sim \mathcal{CN}(0, \sigma_v^2),$$

or equivalently

$$real(v_t) \sim \mathcal{N}(0, \tfrac{\sigma_v^2}{2}), \text{ and } imag(v_t) \sim \mathcal{N}(0, \tfrac{\sigma_v^2}{2}),$$

and the value of the variance was defined by using the signal-to-noise ratio (SNR)

$$SNR = 1 - \log_{10} \frac{A_k^2}{{\sigma_v}^2},$$

where SNR was measured in dB, A_k was the amplitude of the signal, and σ_v^2 was the noise power.

Assume that we know the total number of sinusoids K. The total number of iterations used was J, and the total number of samples per iteration was M. At initial iteration $j = 1$, the IF $q_j^{(m)}(\mathbf{x})$ was chosen as follows for complex amplitudes $\tilde{\mathbf{A}}$ and frequencies $\mathbf{f}$, where $\mathbf{x}$ was consisted by $\tilde{\mathbf{A}}$ and $\mathbf{f}$. We used complex normal distributions for amplitudes $\tilde{\mathbf{A}}_k$, $k = 1,\ 2,\ ...,\ K$, with initial mean at 0 and variance randomly chosen from a set of multiple transition kernels [9]:

$$\mathbf{v}_{\tilde{A}} = [v_{A,1},\ v_{A,2},\ ...,\ v_{A,5}]^\top = {\sigma_{\tilde{A}}}^2 \times [1^2, 0.1^2, 0.01^2, 0.001^2, 0.0001^2]^\top$$

with ${\sigma_{\tilde{A}}}^2 = 0.5^2$. At initial step, each variance was selected with probability of $\alpha_{A_k,d} = \frac{1}{5}$, with $k = 1,\ 2,\ ...,\ K$ and $d = 1,\ 2,\ ,...,\ 5$. We used truncated normal distributions [48] for frequencies $\mathbf{f}_k$, $k = 1,\ 2,\ ...,\ K$, with initial mean at estimated obtained by the Yule-Walker method [67] or by periodogram and variance randomly chosen from

$$\mathbf{v}_f = [v_{f,1},\ v_{f,2},\ ...,\ v_{f,5}]^\top = {\sigma_f}^2 \times [1^2, 0.1^2, 0.01^2, 0.001^2, 0.0001^2]^\top$$

with ${\sigma_f}^2 = 0.1^2$. At initial step, each variance was selected with probability of $\alpha_{f_k,d} = \frac{1}{5}$, with $k = 1,\ 2,\ ...,\ K$ and $d = 1,\ 2,\ ,...,\ 5$. The IF for each frequency $\mathbf{f}_k$ was truncated as not smaller than the samples value of $\mathbf{f}_l$ with $l < k$ and 0, and not greater than the samples value of $\mathbf{f}_l$ with $l > k$ and 1.

Then the algorithm iterated as follows:

1. Draw samples $\mathbf{x}_j^{(m)}$ from $q_j^{(m)}(\mathbf{x})$. Note that we took random order to draw samples of frequencies.

2. Compute weights of the samples

$$\tilde{w}_j^{(m)} \propto \frac{p(\mathbf{x}_j^{(m)}|\mathbf{y})}{\frac{q_j^{(m)}(\mathbf{x}_j^{(m)})}{\prod_{k=1}^{K} c_{k,j}^{(m)}}}; \tag{8.11}$$

where $c_{k,j}^{(m)} = \Phi(\frac{b-\mu_{f_{k,j}}^{(m)}}{\sigma_{f_{k,j}}^{(m)}}) - \Phi(\frac{b-\mu_{f_{k,j}}^{(m)}}{\sigma_{f_{k,j}}^{(m)}})$ when the normal distribution was truncated by $[a,\ b]$. Note $\Phi(\ )$ was the cumulative distribution function of normal distribution.

3. Normalize the weights:

$$w_j^{(m)} = \frac{\tilde{w}_j^{(m)}}{\sum_{k=1}^{M} \tilde{w}_j^{(k)}}. \tag{8.12}$$

4. Update $\alpha_{A_k,d}$ and $\alpha_{f_k,d}$, with $k = 1,\ 2,\ ...,\ K$ and $d = 1,\ 2,\ ,..., 5$:

$$\alpha_{A_k,d} = \sum_{m=1}^{M} w_j^{(m)} \mathbf{I}_d \left(v_{A_k,j}^{(m)} = v_{A,d} \right) \tag{8.13}$$

$$\alpha_{f_k,d} = \sum_{m=1}^{M} w_j^{(m)} \mathbf{I}_d \left(v_{f_k,j}^{(m)} = v_{f,d} \right). \tag{8.14}$$

In order to keep every variance valid after each iteration, re-scaling was employed to $\alpha_{A_k,d}$ and $\alpha_{f_k,d}$ to ensure that the minimum weight for each available variance is 0.05.

5. Obtain the estimates using MAP

$$\hat{\mathbf{x}} = \max_x\ p(\mathbf{x}|\mathbf{y}) \propto \max_x\ p(\mathbf{y}|\mathbf{x})p(\mathbf{x}), \tag{8.15}$$

or using MMSE

$$\hat{\mathbf{x}} = \frac{\sum_{t=1}^{j} \sum_{m=1}^{M} \mathbf{x}_t^{(m)} w_t^{(m)}}{\sum_{t=1}^{j} \sum_{m=1}^{M} w_t^{(m)}}. \tag{8.16}$$

6. Resample the samples according to their weights.

7. At iteration $j > 1$, the IF $q_j^{(m)}(\mathbf{x})$ for complex amplitudes $\tilde{\mathbf{A}}$ and frequencies $\mathbf{f}$ were updated as:

   (a) We used complex normal distributions for amplitudes $\tilde{\mathbf{A}}_k$, with mean at $\mathbf{A}_{k,j-1}^{(m)}$ and variance randomly chosen from $\mathbf{v}_{\tilde{A}}$ randomly with probability of $\alpha_{A_k,d}$.

   (b) We used truncated normal distributions for frequencies $\mathbf{f}_k$, with mean at $\mathbf{f}_{k,j-1}^{(m)}$ and variance randomly chosen from $\mathbf{v}_f$ with probability of $\alpha_{f_k,d}$. The IF for each frequency $\mathbf{f}_k$ was truncated as not smaller than the samples value of $\mathbf{f}_l$ with $l < k$ and 0, and not greater than the samples value of $\mathbf{f}_l$ with $l > k$ and 1.

We had $d_y = 25$ observations with $K = 3$ sinusoids generated according to the model above. Observations were generated using three frequencies $f_1 = 0.2$, $f_2 = 0.5$, and $f_3 = 0.52$, with amplitudes $\tilde{A}_1 = 1$, $\tilde{A}_2 = 1$, and $\tilde{A}_3 = 1e^{i\pi/3}$. MPMC and MultiMPMC were applied to estimate parameters in this 9-dimensional problem. The prior for $\tilde{A}_k$ used for the MPMC and MultiMPMC was

$$\tilde{A}_k \sim \mathcal{CN}(0, 5).$$

The performances of the methods for estimation of the parameters of the third sinusoid are shown as Figure 8.14 and 8.15. The performances of the algorithms to this problem were quantified based on the MSE of the parameters to be estimated, given by 8.17. All the points on the plot were averaged over $R = 500$ runs with 1 to 20 iterations and with $SNR = 5$. In each run, $M = 600$ samples were generated from an initial importance function for PMC and MPMC, and $M_k = 200$ for each estimator, which sums up to $M = 600$ total samples, when implementing the MultiPMC and MultiMPMC algorithms. It can be concluded from the plots that MultiPMC and MultiMPMC performed more accurately and converged much faster.

The performances of the proposed methods in terms of the MSE of the estimated frequencies for various values of SNR are shown in Figure 8.17. All the points on the plot are averaged over $R = 500$ runs with sample size of $M = 600$ and iteration number of $J = 20$.

The methods were also compared with the MUltiple SIgnal Classification (MUSIC) algorithm, which estimated the pseudospectrum of the observations using Schmidt's eigenspace analysis method [65]. The conventional MUSIC algorithm performs similarly with the proposed methods at low SNRs, but does not improve with SNR as do the PMC methods. The proposed methods perform well, and their MSEs converge to the CRBL as the SNR increases.

Additionally, Figure 8.16 displays the cumulative distribution functions (CDFs) of the marginalized posteriors of the nonlinear parameters approximated by PMC, MPMC, MultiPMC and MultiMPMC algorithms. The CDFs show that the marginalized posterior of the frequencies has jumps around the true values of the frequencies.

The efficiencies of resampling of MultiPMC, MPMC, and MultiMPMC were also studied. Resampling rate along iterations can be represented using effective sample size (ESS), as

$$ESS_j = \sum_{m=1}^{M} [(w_j^{(m)})^2]^{-1},$$

where j denoted the iteration number, j = 1, 2, ...; and m represented the index of the particle, m = 1, 2, ..., M. ESS is the equivalent number of independent samples, which would provide the same information as the M-size sample set [53, 55]. The ESSs of the proposed methods for SNR=5 are shown in Figure 8.18. All the points on the plot were averaged over $R = 500$ runs for iteration number of 1 to 30 with $SNR = 5$. In each run, an amount of $M = 600$ samples were generated at each iteration for PMC and MPMC, which provides each unknown frequency $M = 200$ samples; $M_k = 200$ samples were generated for each estimator at each iteration when implementing the MultiPMC and MultiMPMC algorithms. It can be concluded from the figure that ESS of the MultiPMC and the MultiMPMC increases faster than the other two; also the MPMC and the MultiMPMC have higher ESS as the iterations increase and therefore are more efficient when using resampling.

8.3.4 Robustness of MPMC and multiple MPMC sampler with an example of three frequencies

The problem of frequency estimation of complex sinusoids in noise is considered in this section to show the robustness of the distributed MPMC algorithms. The model of the problem is the same as in 8.10. We had $d_y = 25$ observations with $K = 3$ sinusoids generated according to the model. Observations were generated using three frequencies $f_1 = 0.2$, $f_2 = 0.5$, and $f_3 = 0.5 + \Delta_f$, with amplitudes $\tilde{A}_1 = 1$, $\tilde{A}_2 = 1$, and $\tilde{A}_3 = 1e^{i\pi/3}$. The frequencies f_2 and f_3 are close to each other with a difference of Δ_f. MPMC and MultiMPMC were applied to estimate parameters in this 9-dimensional problem with different values of Δ_f. The prior for $\tilde{A}_k$ used for the MPMC and MultiMPMC was

$$\tilde{A}_k \sim \mathcal{CN}(0, 5).$$

We define a estimate estimate as an outlier when a frequency component was more than $\Delta = 0.1$ apart from the real value, shown as in Figure 8.19.

The performances of the MPMC and multiple MPMC in term of outlier rate for estimation of the frequencies are shown as Figure 8.20. The outlier rate is based on $R = 500$ runs with 1 to 20 iterations and with $SNR = 5$. In each run, $M = 600$ samples were generated from an initial importance function for the MPMC sampler; and $M_k = 200$ for each estimator, which sums up to $M = 600$ total samples, when implementing the MultiMPMC algorithm. It can be concluded from the plot that MultiMPMC performed more robust as the Δ_f decreased.

The performances of the MPMC and multiple MPMC algorithms to this problem were quantified based on the MSE of the parameters to be estimated, given by 8.17. All the points on the plot were generated from the above $R = 500$ runs after dropping the outliers. It can be concluded from the plots that even among the *good* estimates (non outlier), the multiple MPMC algorithm performed more accurately and more robust to a small Δ_f.

8.3.5 Comparison of MPMC and Gibbs MPMC sampler with an example of ten frequencies

We apply the Gibbs MPMC to a ten frequency problem, where each of the conditionals is a truncated Gaussian centered at the selected particle from the previous iteration. The conditioning parameters were used for deciding the truncation points of the Gaussian. For example, if the frequency $f_{k,j}$ needed to be generated, we used as a generating function the truncated Gaussian centered at $f_{k,j-1}^{(m)}$ with cutoff points $f_{k-1,*}^{(m)}$, and $f_{k+1,*}^{(m)}$, where the $*$ stands for the most recent sample of f_{k-1} and f_{k+1}, respectively.[1] The importance of this choice is demonstrated in the experimental results shown below.

We tested the method by conducting simulations as follows. We simulated $d_y = 25$ observations with $K = 10$ sinusoids, whose frequencies were $f_1 = 0.2$, $f_2 = 0.3$, $f_3 = 0.32$, $f_4 = 0.5$, $f_5 = 0.52$, $f_6 = 0.7$, $f_7 = 0.75$, $f_8 = 0.8$, $f_9 = 0.82$, and $f_{10} = 0.9$, with amplitudes $a_k = 1$, for $k = 1,\ 2, ...,\ 10$, and phases $\phi_k = 0$, for $k = 1, ...,\ 4,\ 6..., 10$ and $\phi_5 = \pi/4$, respectively. The value of the noise power was defined by using the signal-to-noise ratio (SNR)

$$SNR = 10 \log_{10} \frac{a_k^2}{\sigma_v^2}$$

measured in dB.

For comparisons, we employed the MPMC method, which also uses the truncated Gaussians with the same centers but with truncating points obtained in the previous iteration only (so, they are not the most recent drawings). We also found the CRLBs for the frequencies of interest. When implementing the algorithms, we used the Yule-Walker method for getting the initial estimates [44].

The performance of the algorithms was quantified in terms of the mean square error (MSE) given by

$$MSE(f_k) = \frac{1}{R} \sum_{r=1}^{R} (\widehat{f}_k(r) - f_k)^2 \tag{8.17}$$

[1] Note that when $k = 1$, the lower cutoff point is 0, and when $k = K$ the upper cutoff point is 1.

where R represents the number of realizations, $\widehat{f_k}(r)$ denotes the estimate obtained in the r-th run, and f_k is the true value of the frequency.

Figure 8.22 shows the MSE of the two algorithms as a function of iterations (the maximum number was $J = 20$ iterations). The SNR was 5 dB, $M = 900$ particles, and $R = 500$ runs. At each run, the estimates were obtained from all the particles in the $J = 20$ iterations. In the figure, the performance of the novel scheme is denoted by G-MPMC. From the graphs, it is clear that G-MPMC outperforms the MPMC. The G-MPMC estimates of the unknown frequencies converge much more quickly to the true values.

In Figure 8.23, we see the MSE for different values of particle size (M was changed from 300 to 1800 particles), SNR $= 5$ dB, $J = 10$ iterations, and $R = 500$ runs. At each run, as before, the estimates were obtained from all the particles in the $J = 10$ iterations. The plots show that the G-MPMC can achieve the same accuracy with a smaller amount of particles than the MPMC algorithms, and therefore, it is less computationally expensive.

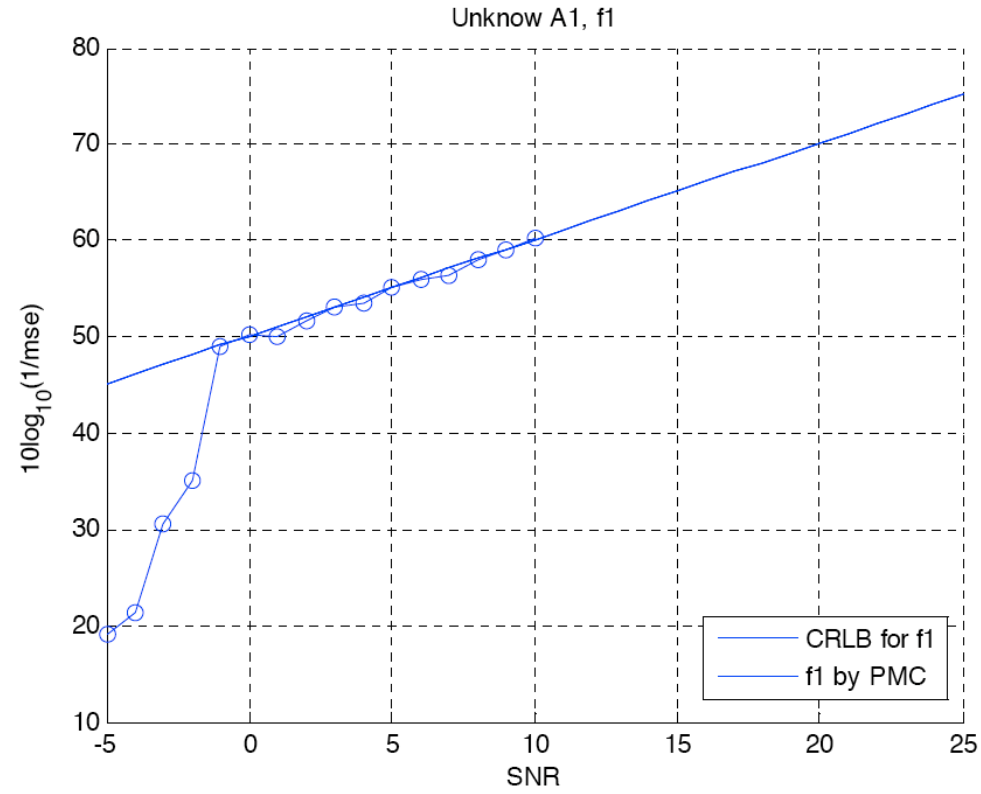
The MSE for various values of SNR is shown in Figure 8.24. All the points on the plot were averaged over $R = 500$ runs with a particle size of $M = 900$ and for $J = 10$ iterations. The proposed method again outperforms the MPMC considerably.

We point out that in all the simulations, we present the results of point estimates. The particles and their weights provide, however, much more information. They can be used to compute estimates of other integrals of functions or to simply produce other types of statistical inference.

	outliers	outlier rate
MUSIC	278	0.556
Yule-Walker	258	0.516
MPMC (M=900, J=10)	9	0.018
G-MPMC (M=900, J=10)	3	0.006
MPMC (M=900, J=20)	2	0.004
G-MPMC (M=900, J=20)	0	0

Table 8.1: Outliers of different methods.

Finally, we define a estimate estimate as an outlier when a frequency component was more than $\Delta = 0.1$ apart from the real value, shown as in Figure 8.19. Table 8.1 show the numbers of outliers of the MUSIC ((Multiple Signal Classification) algorithm, the Yule-Walker method, MPMC method, and the proposed G-MPMC method. All the data were averaged over $R = 500$ runs with $SNR = 5$. As seen, for high dimension problem, neither the Yule-Walker nor the MUSIC algorithm performs well as the G-MPMC and MPMC methods do.



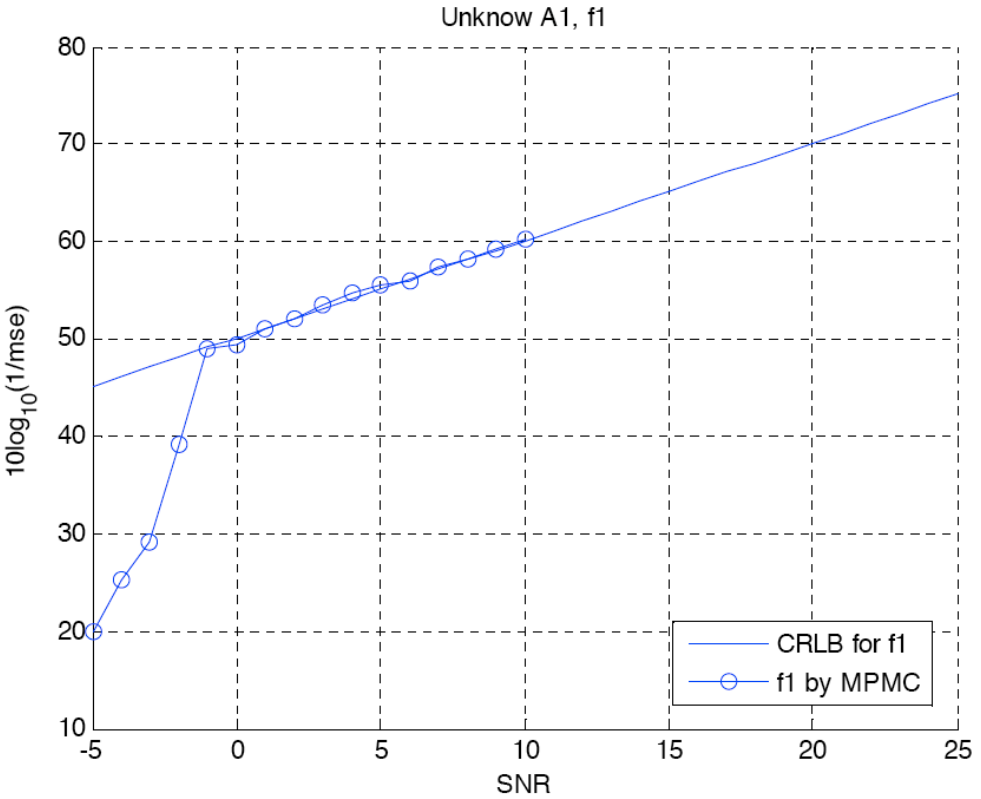
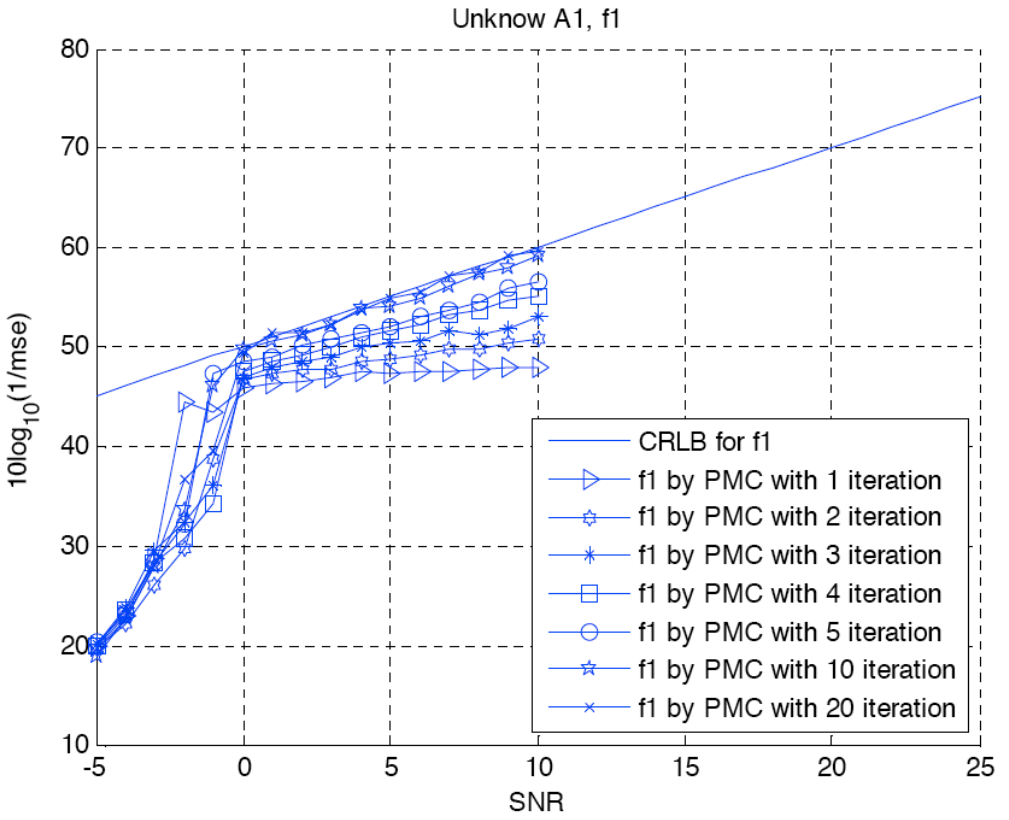


Figure 8.9: $10\log_{10}(1/MSE)$ vs. SNR for one frequency with unknown amplitude and phase estimated using the PMC and MPMC algorithms.



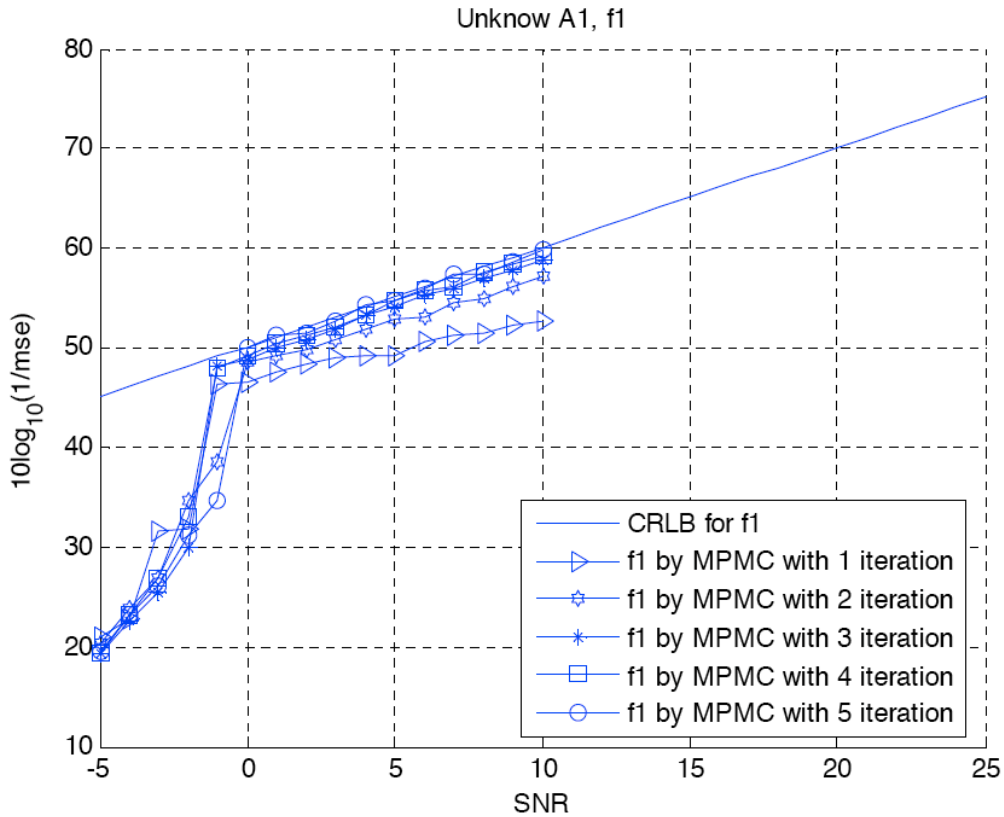


Figure 8.10: $10\log_{10}(1/MSE)$ vs. SNR for one frequency with unknown amplitude and phase estimated using the PMC and MPMC algorithms with different iterations.

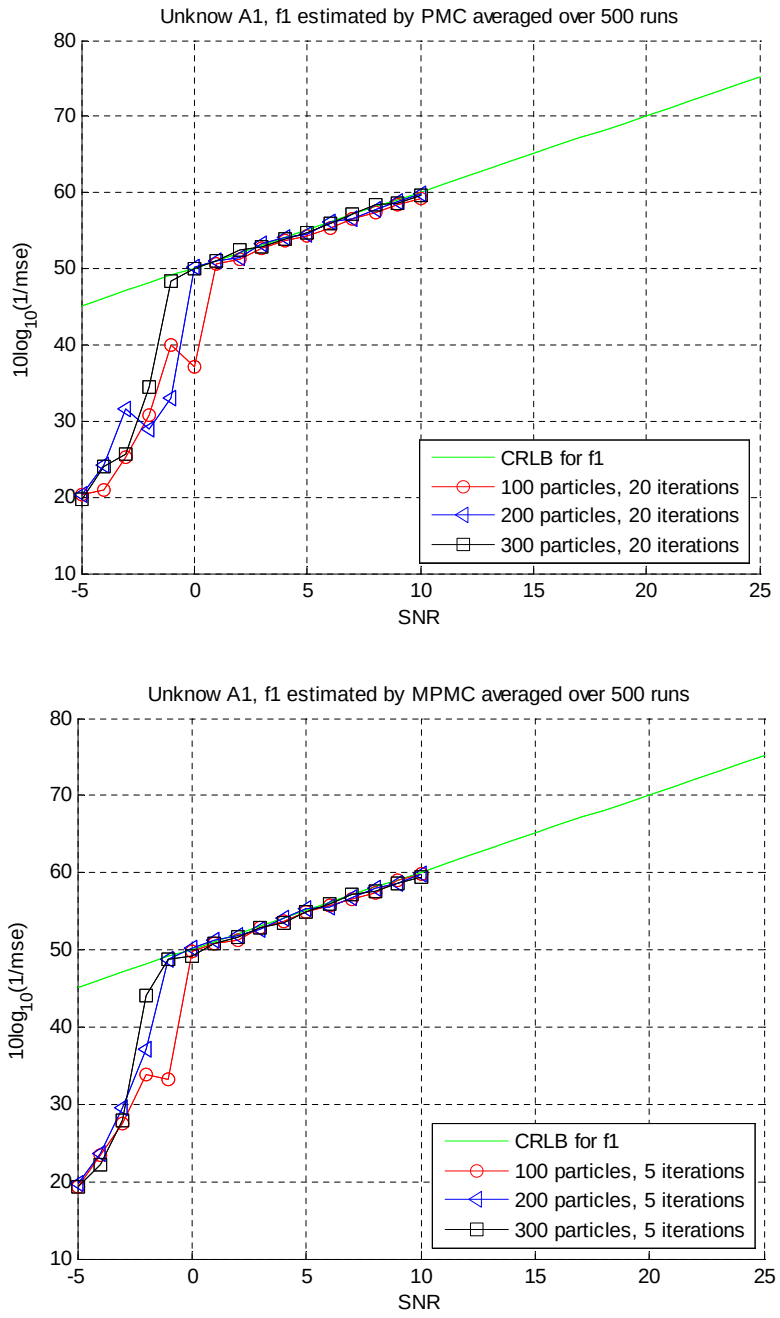


Figure 8.11: $10\log_{10}(1/MSE)$ vs. SNR for 1 frequency with unknown amplitude and phase estimated using the PMC and MPMC algorithms with different numbers of particles.

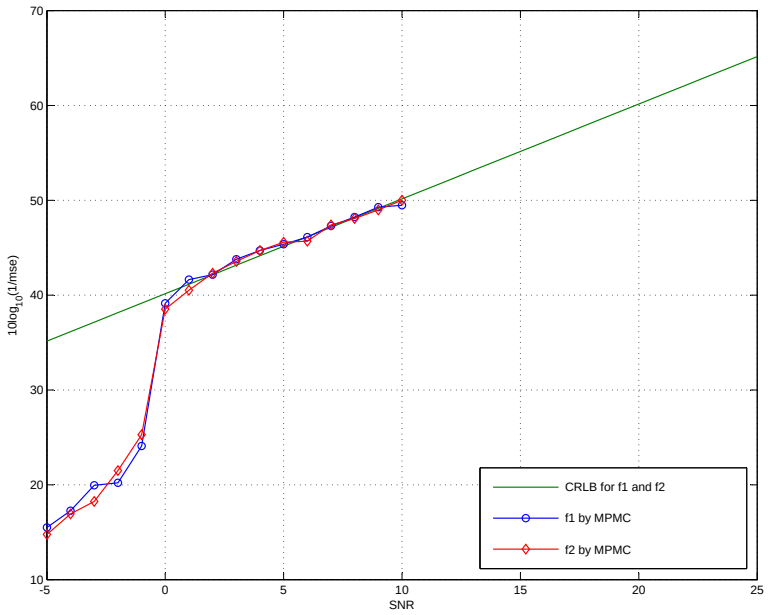


Figure 8.12: Performance of the MPMC for two complex sinusoids in AWGN.

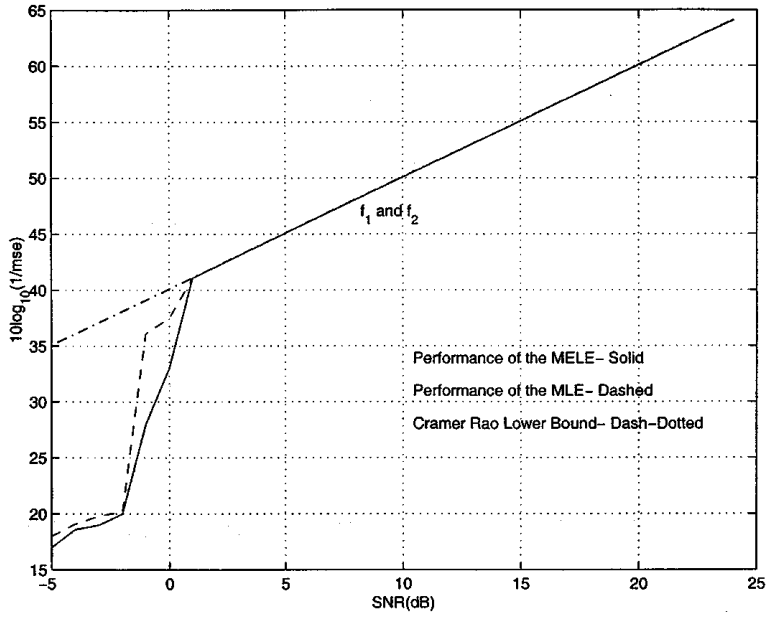


Figure 8.13: Performance of the MLLFE for two complex sinusoids in AWGN.

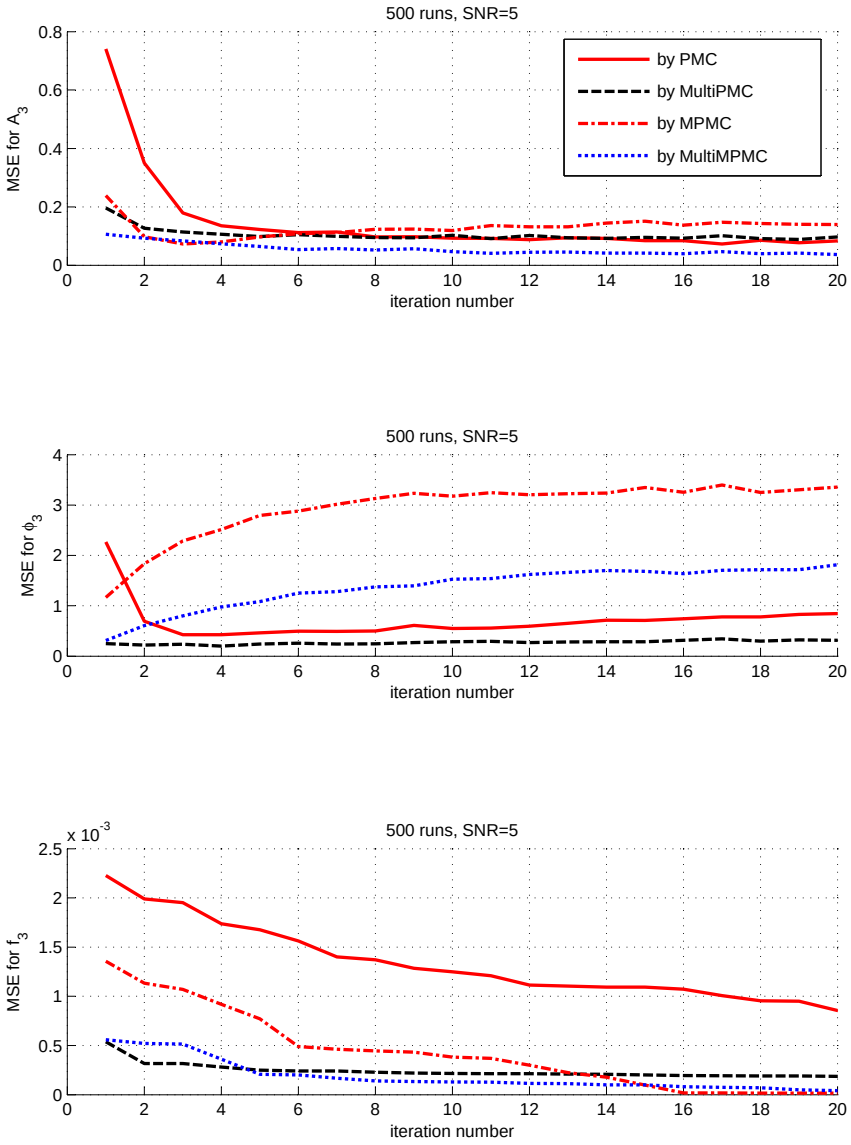


Figure 8.14: MSE of estimates vs iterations using the PMC, MultiPMC, MPMC, and MultiMPMC algorithms.

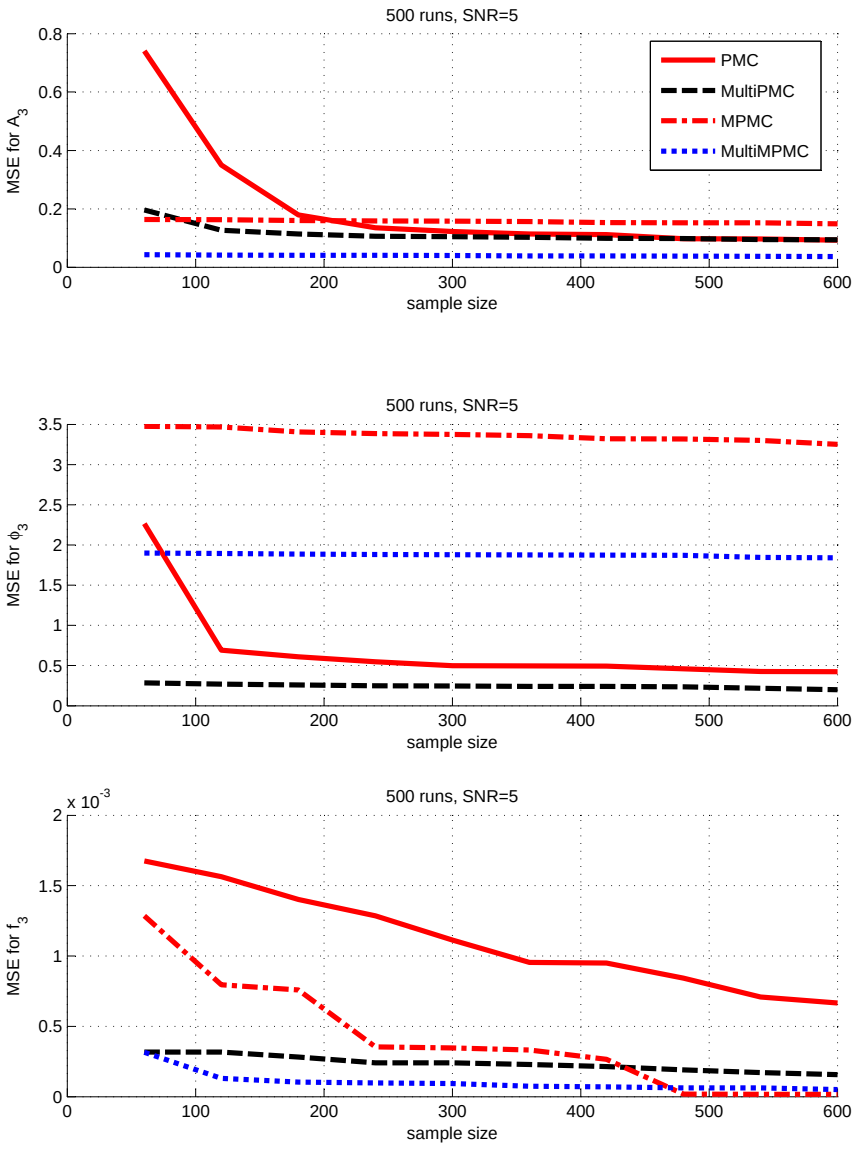


Figure 8.15: MSE of estimates vs number of samples using the PMC, MultiPMC, MPMC, and MultiMPMC algorithms.

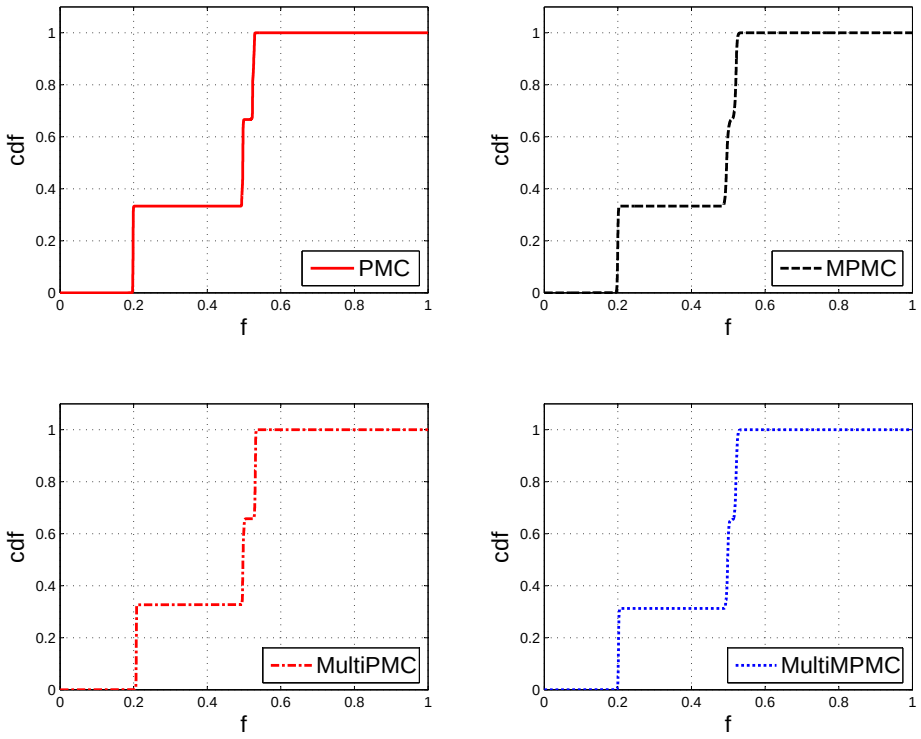


Figure 8.16: CDF of frequencies using PMC, MPMC, MultiPMC and MultiMPMC algorithms.

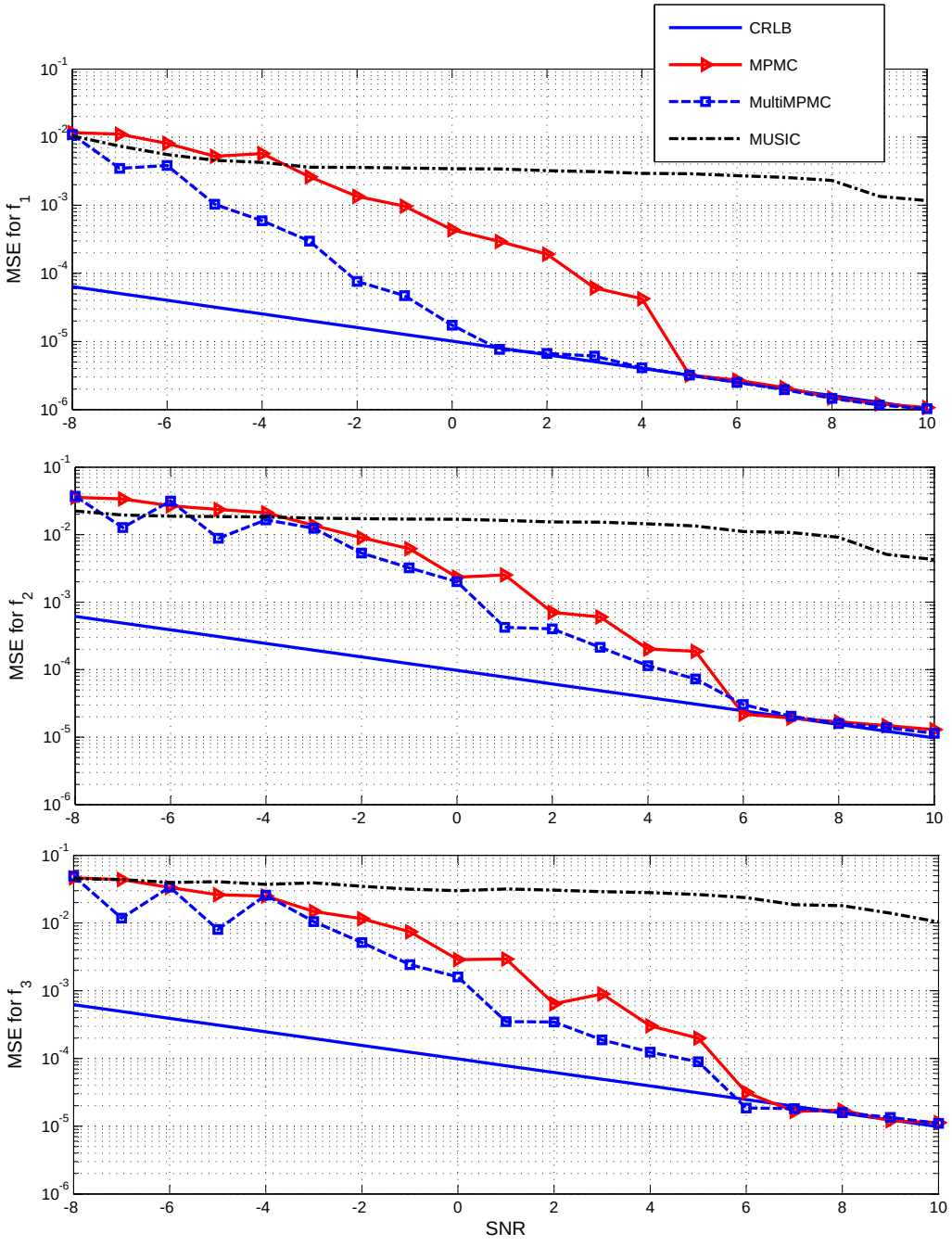


Figure 8.17: Estimates vs SNR using MPMC, MultiMPMC and MUSIC algorithms.

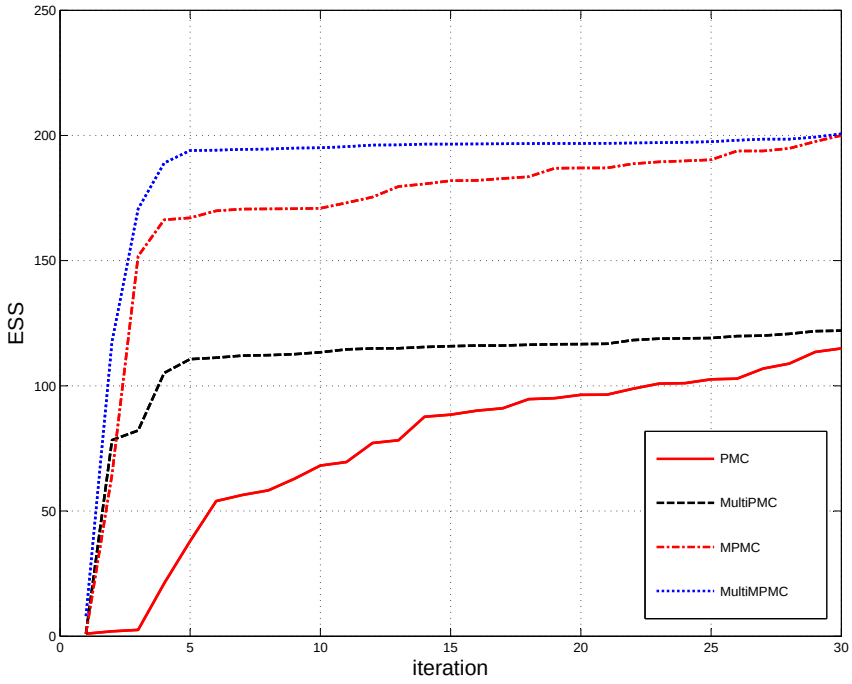


Figure 8.18: ESS vs iteration using PMC, MultiPMC, MPMC, and MultiMPMC algorithms.

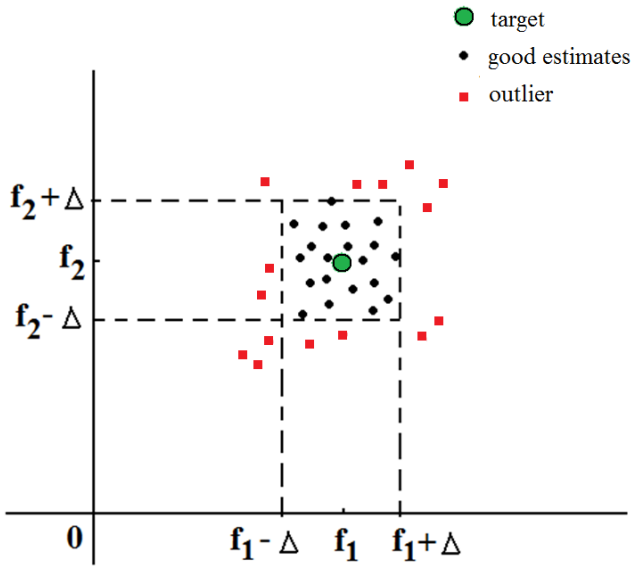


Figure 8.19: Definition of an outlier.

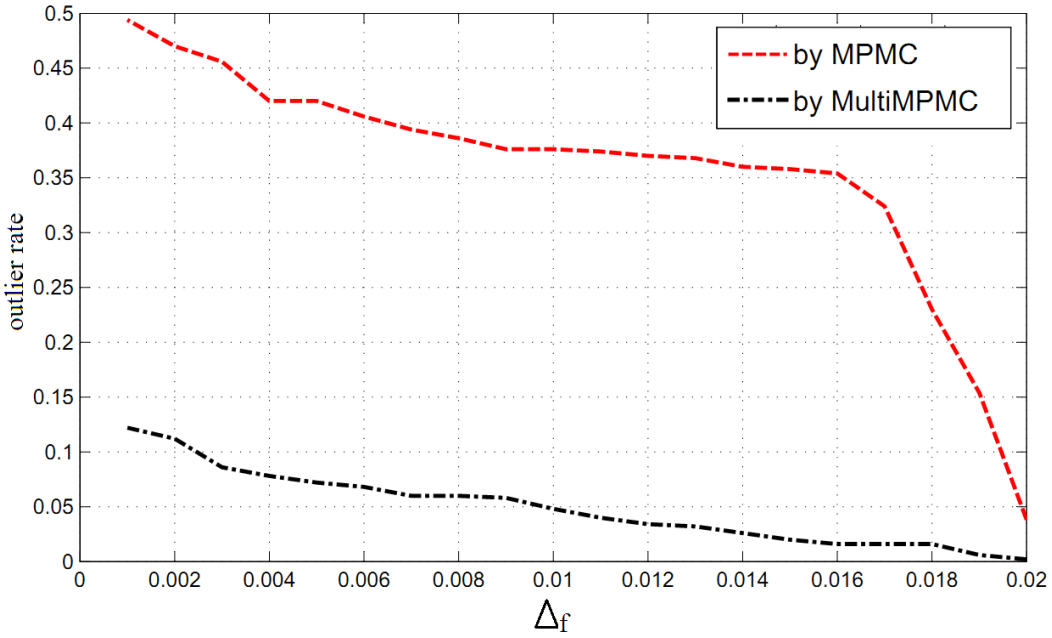


Figure 8.20: Outlier rate of MPMC and multiple MPMC algorithms.

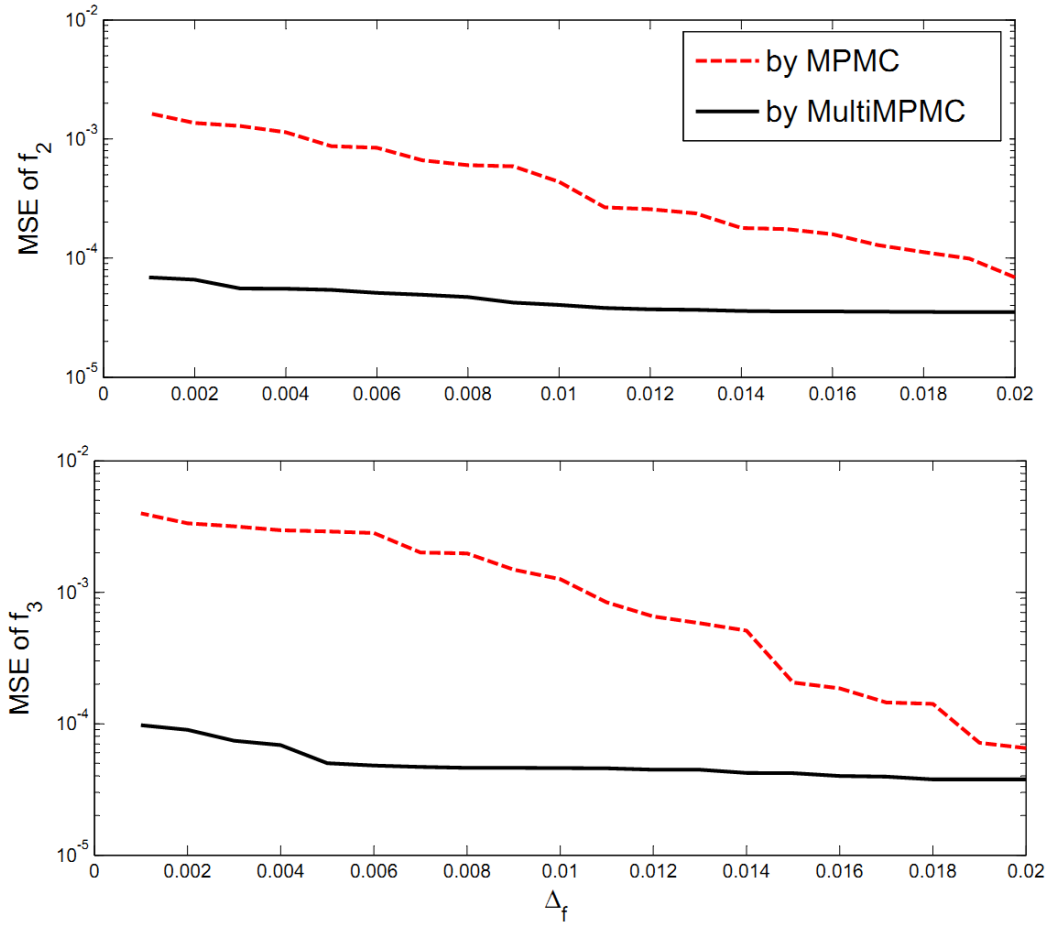


Figure 8.21: MSE of MPMC and multiple MPMC algorithms after dropping the outliers.

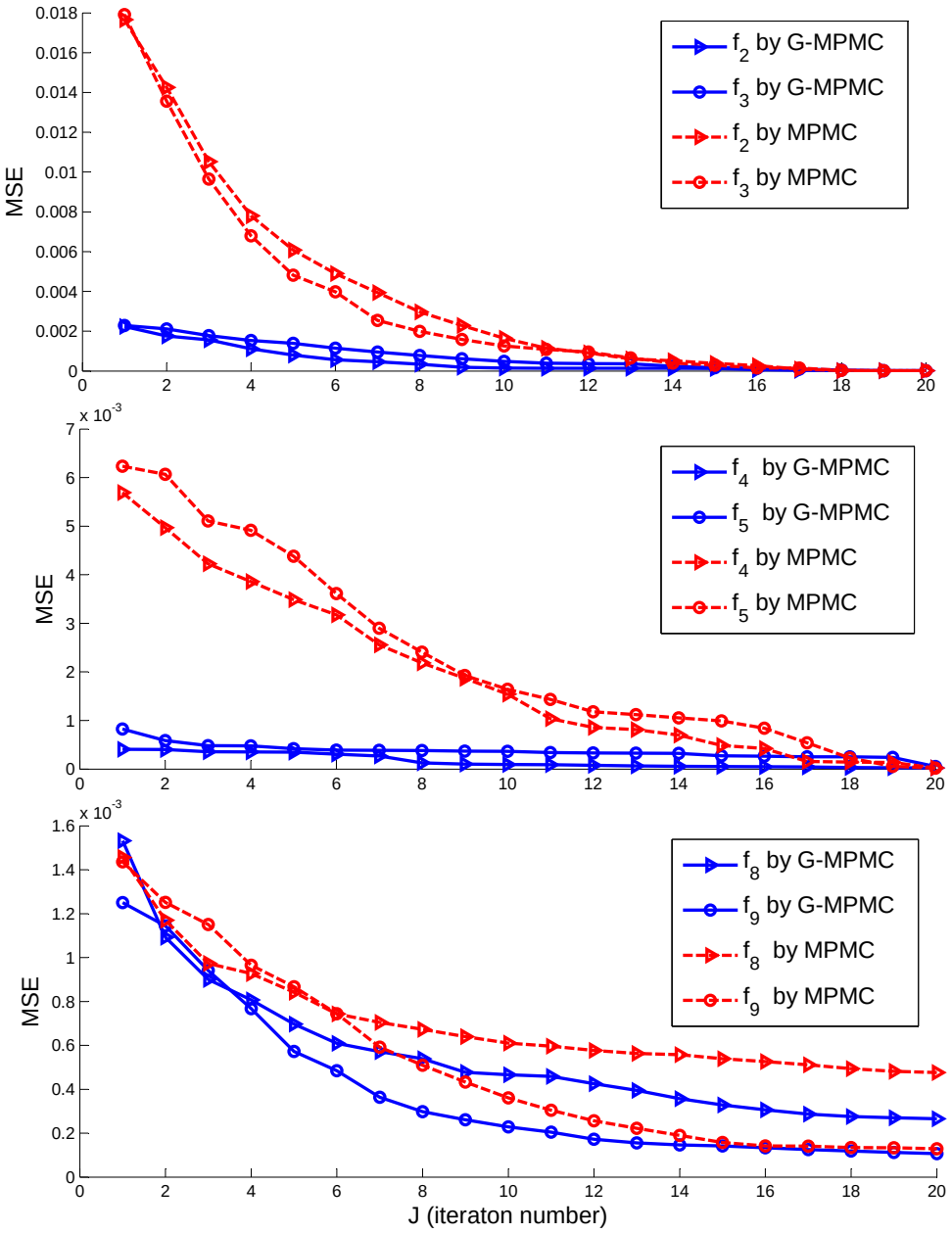


Figure 8.22: MSE as a function of iteration.

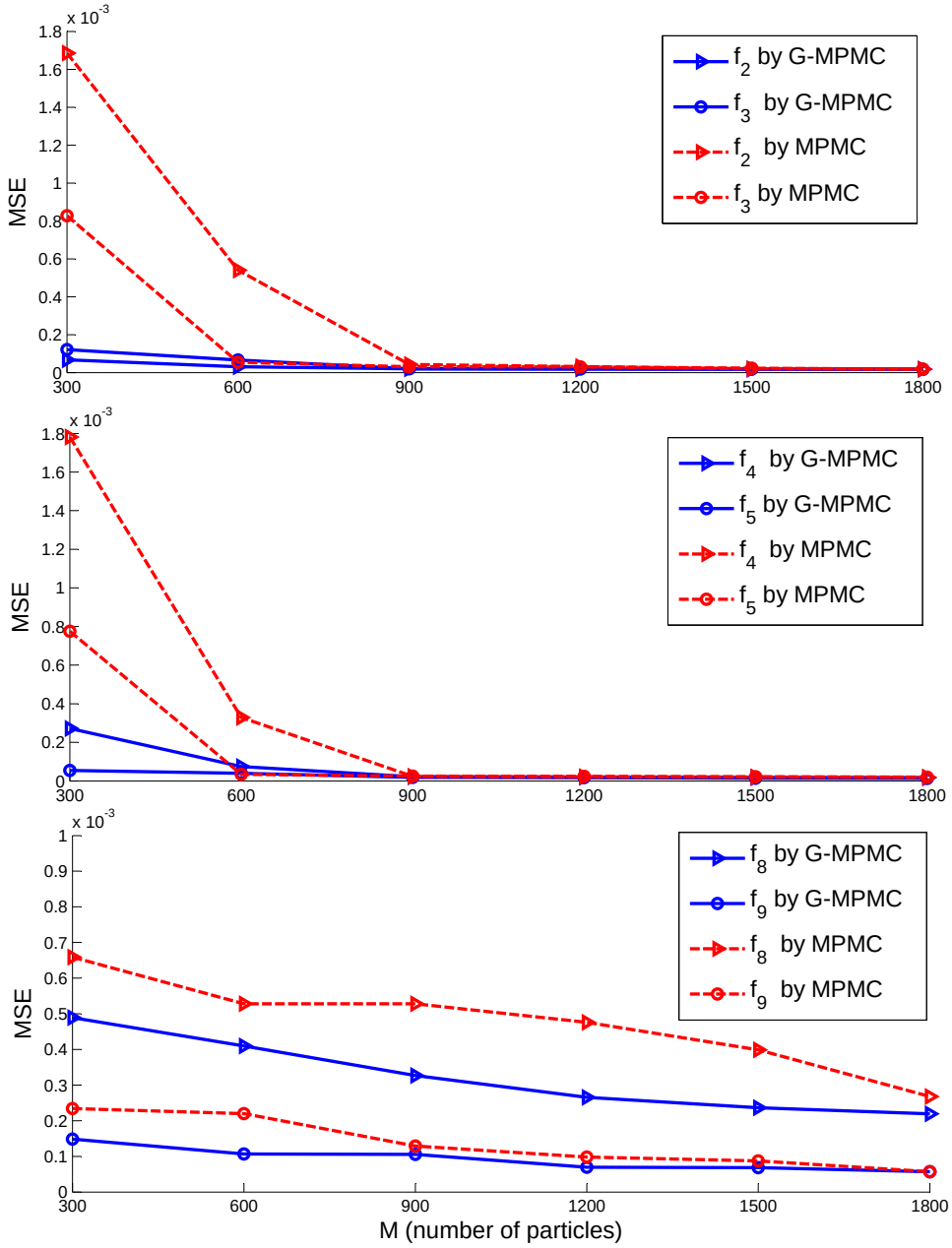


Figure 8.23: MSE as a function of particle size.

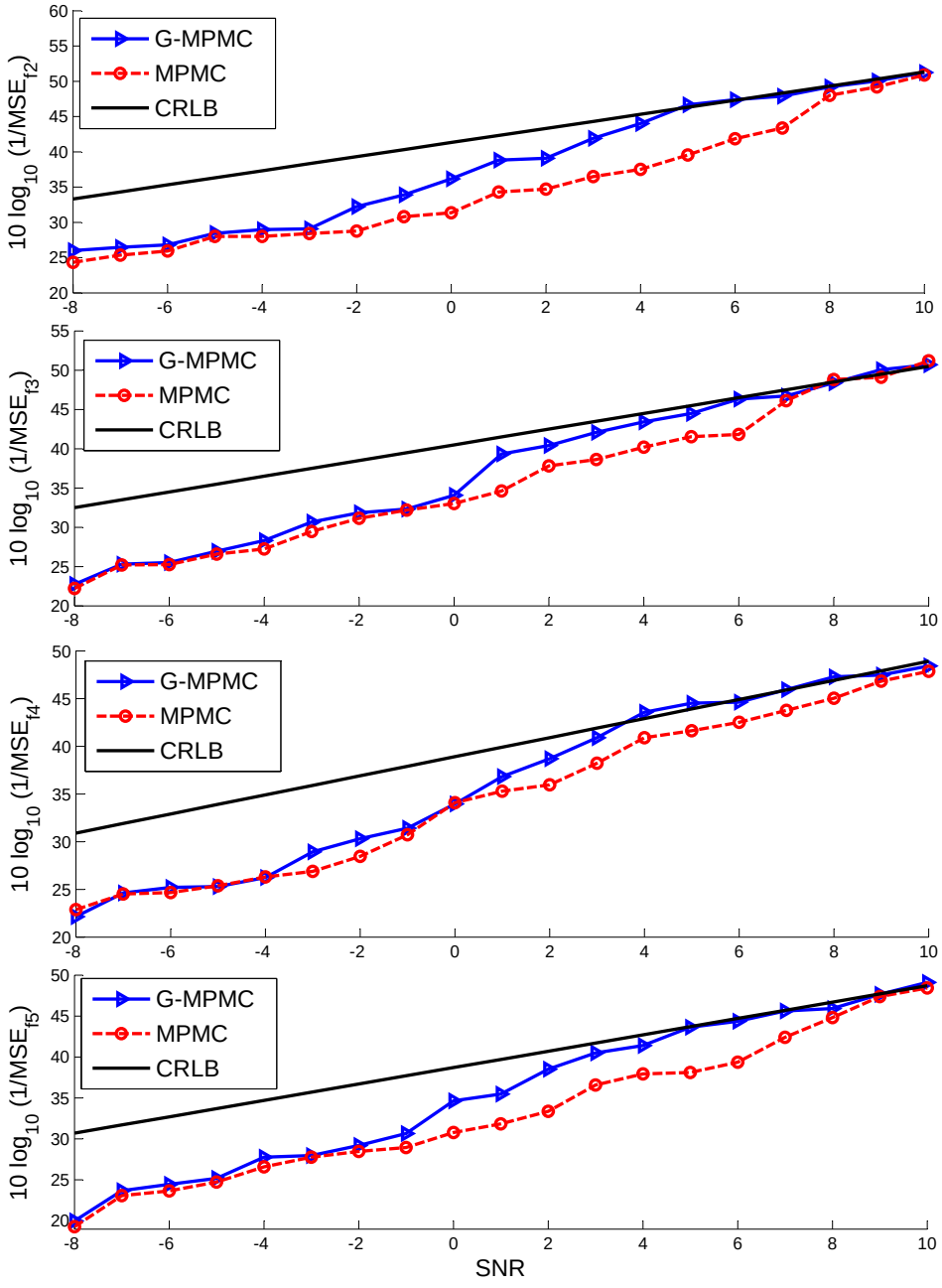


Figure 8.24: MSE as a function of SNR.

CHAPTER 9

Conclusions and future work

9.1 Conclusions

PMC is a methodology for approximating joint distributions of unknowns. The approximation is with random measures represented by samples and weights [10]. The method is iterative where at each iteration samples of the unknowns are generated from a known distribution. These particles are then assigned weights according to the importance sampling principle [63]. The particles and their weights from *all* the iterations are used for approximation.

The PMC has some resemblance to MCMC methods. However, the samples of PMC, unlike in MCMC methods, have different weights, and the estimates of integrals under the target distribution using the samples and the weights are approximately unbiased. The PMC has the advantage of not requiring asymptotic arguments.

The key principle for constructing the approximations with PMC is importance sampling, which is a technique for estimating properties of a particular distribution with samples generated from a different distribution. This principle is always employed in the well known particle filtering methods [18].

The estimation quality and convergence efficiency rely on many factors including the number of samples and the choice of importance function. The computational complexity of the PMC algorithm becomes increasingly challenging as the dimension of unknowns increases. In this work, we proposed several novel PMC algorithms for high-dimensional nonlinear problems. These new structures are proven to provide accurate performance and computational efficiency.

In particular, in some problems, we find that some of the unknown parameters are conditionally linear given the remaining parameters. The MPMC has been proposed to lower the computational cost by only generating samples of the nonlinear parameters and marginalizing

the remaining linear parameters [8]. The marginalization decreases the dimension of the unknowns to be estimate and therefore minimizes the corresponding computational cost. This approach has shown to be accurate and feasible.

We also propose distributed PMC algorithms, MultiPMC, where the state space of the system is partitioned into several subspaces of lower dimensions and handled by a set of marginalized PMC estimators. Partitioning of the original high-dimensional problem is a computationally attractive technique. The multiple structure breaks down the exponential increase of the computational burden into a linear increase. Indeed, the partitioning estimation algorithms have flexible structure for parallel processing, which will overcome the computational and storage constraints of a large class of practical applications. Simulation results show the accuracy and feasibility of the method as well as its improvement with respect to other conventional approaches. The MPMC and MultiPMC have also been combined to reach a more simplified and still accurate implementation of the PMC.

Another critical issue in applying PMC methods is the choice of generating functions of the particles. In this work, we also proposed that these functions are alternating conditionals, as in the case of Gibbs sampling. Thus, the overall proposal function is a product of conditionals, where the sampling from each conditional is easy. It is expected that with alternating conditionals one can efficiently generate particles in high dimensions. The method was tested on the problem of frequency estimation of multiple sinusoids. The obtained results show very good performance of the method.

9.2 Future work

Further study on the theory of the PMC family of algorithms is of great interest. Applications of PMC algorithms to real problems of high dimensional and complex systems will also be pursued.

9.2.1 Theory

9.2.1.1 Systematic partition of the space of unknowns

One future direction of research is to investigate the systematic partitioning of the problem of interest. How to group the unknowns and how many unknowns to be grouped will be a main topic of interest. For instance, in a typical weather forecast problem [29], the dimension of the state-vector in most oceanic and atmospheric models is extremely high, often including 10^6 and 10^8 parameters. For such high-dimensional systems, sampling is a challenging computational task. The use of a proper state partitioning data structure can reduce the dimension of the data to be processed. Two of the critical issues are how to extract the dependencies among all the parameters, and how to partition the high-dimensional space of the unknowns.

The formalism of probabilistic graphical models [71] may provide a unifying framework for capturing complex dependencies among random variables, and building large-scale multivariate statistical models. Graphical models will be of great help in state partition problems. Exponential family representations [71] will help to simplify and clarify the connections among the likelihoods, marginal probabilities and joint probabilities in a complicated and high-dimensional system.

The partition of the space of unknowns should be optimized to achieve the least computational cost and maximized efficiency. One may also study the parallel implementation of the distributed PMC algorithms once the problem space is partitioned. How to assign the estimators to the subspaces can also be a future research topic.

9.2.1.2 Systematic population size

Another future direction is to study the systematic population size for a fixed dimensional problem. Currently there is not any work done on the population size for a given problem. Most people use the population size based on experiments when implementing the PMC algorithms. However, it might become crucial to determine how many samples are necessary to

explore the whole space of unknowns as the dimension increases dramatically and computational cost goes up accordingly.

Another interesting topic is to study the systematic population size along iterations. As the samples and estimates converge to the target area of the space of unknowns, less samples will be needed to explore the local area yet maintaining the accuracy. Dynamic population size scheme will further decrease the computation burden while achieving the same accuracy.

9.2.1.3 Cost-based PMC

The general PMC requires a complete knowledge of the probabilistic distributions of the measurement noise. However, in some cases this is not available, and therefore the calculation of the weights when implementing the PMC algorithm is not possible. An alternative PMC scheme, called cost-based PMC, will be researched.

A cost function will be defined and used instead of the traditional Bayesian weight update at each iteration. A similar approach has been considered for sensor self-localization under beacon position uncertainty [70], where costs are described by spatial parametric regions. One may also refer to [57,58], where cost-reference importance sampling are elaborated in PFs for MAP symbol detection without explicit estimation of the channel parameters.

9.2.1.4 Nonadditive noise problems

In some real-world problems, the noise is not always additive to the signals. A more general model of observations is

$$\mathbf{y} = h(\mathbf{x}, \mathbf{v}), \tag{9.1}$$

which is the usual case if the noise interfere with the communication devices such as mobile telephones, paging devices, digital hearing aids and protection devices. These scenarios are also of interest to explore.

Note that, one specific case of Equation 9.1 is the multiplicative noise,

$$\mathbf{y} = \mathbf{x} \cdot \mathbf{v},$$

which is a model widely used in image processing [31, 51]. The noise can be converted to an additive one if taken the natural logarithm on both sides, when $\mathbf{x} > 0$ and $\mathbf{v} > 0$:

$$ln\mathbf{y} = ln\mathbf{x} + ln\mathbf{v}.$$

Then the algorithms for signals in additive noise would fit for the converted model. For any other case, other solutions will be devised. Note that, a different noise model does not change the implementation of the PMC algorithms; it only affects the way the one updates the IFs.

9.2.1.5 Convergence

The convergence of MC algorithms were discussed in [23] and [14]. For a recent study on the convergence of PMC, one can refer to [21]. If PMC algorithm converges, MPMC also converges. At this time, we do not have proof for convergence of the multiple PMC algorithms. Further study will find the conditions when the multiple PMC schemes converge.

9.2.2 Applications

We will also be interested in looking for applications to real problems using PMC and MPMC. In high dimensional problems when the likelihood function is analytically or computationally intractable, MC methods can provide ways of evaluating posterior distributions. Some MC methods are applied to specific real-world high-dimensional system and proved to be effective as presented in the following.

9.2.2.1 Geophysical problems

Many modern geophysical problems, such as weather forecasting [26], are characterized by extremely high-dimensional systems and pose difficult challenges for assimilation of system information and observations. Two MC-based Kalman filters were applied to such problems in [29] in order to update and forecast the atmospheric states in high-dimensional systems.

The atmospheric state are modeled using a probability density function, with extremely high-dimensions [29]. Many of the parameters are highly correlated with each other and the singularity of the covariance matrix makes it difficult to estimate with traditional methods. However, with the new proposed methods we expect to marginalize this part of the parameters to simplify the problem and meanwhile overcome the computational problem caused by the singular covariance matrix. The performance might be further improved by proper partition of the problem space.

9.2.2.2 Finance problems

MC methods are widely used in finance and mathematical finance for the last few decades. They became an important tool in valuing and analyzing complex instruments, portfolios and investments [32]. MC methods have been extensively used in option pricing for a few decades [41,50]. There have been a wide variety of MC approaches applied: quasi MC (QMC), stratification, importance sampling, etc. The importance of MC for option pricing, against other numerical approaches, is the ability to deal with high-dimensional integrals.

However, there still could be more improvement of the performance. For instance, a typical multi-factor tree, usually the joint product of a number of binomial trees, expanses exponentially as iteration goes up [41]. Some part of the tree becomes less correlated or even isolated from the rest and one may partition this part from the main tree while valuate the whole trend of the tree. Moreover, under certain conditions, some of the option parameter linearly depend on the rest. In these cases, one can apply the marginalized algorithm to achieve same accurate estimation but with much less computational cost.

9.2.2.3 Biological problems

Shadow Hybrid MC, as a new method for sampling the phase space of large biological molecules, is employed in [35] to achieve more efficient sampling for protein sequences. It improves sampling by allowing larger time steps and system sizes in the molecular estimation. However, [35] uses a general scheme with uniform population size for all sized proteins. One might want to use a PMC scheme with systematic population size. For instance, a small sized protein with less parameters needs less samples, while a large sized protein with more parameters needs more samples to extract the information. One might also want to partition the space of unknowns/parameters and deal with the protein sequence with multiple samplers and estimators.

Transcription factor binding site identification problems [40] have faced difficulties due to model mismatch and the nature of the biological sequence, which results in inference in a high dimensional, highly multimodal space, and consequently often display only local convergence. A MC-based method is applied to the problem [40] to improve the performance in this scenario. The MC element increases the robustness of the algorithm. However, one might expect samples of better "quality" when using the Gibbs PMC scheme and may achieve better performance regarding robustness and convergence.

To summarize, there have been many PMC applications in real problems. However, only the standard PMC schemes have been discussed and employed. One may expect the advantages of the proposed new members in PMC family. Especially for high-dimensional systems, one may achieve accuracy and computation efficiency while implementing the proposed algorithms.

BIOGRAPHY OF THE AUTHOR

Bingxin Shen was born in Jiangsu, China, on July 26, 1983. She graduated from the Handan Diyi High school, Hebei, China, in 2001. Following the completion of her high school education, she entered Department of Electronic Engineering at Beijing Institute of Technology. In 2005, She graduated as a member of honor 01'Experimental Class (consisted of top 43 students out of the Class of 2005) and received a degree in Information Engineering.

In September 2005, she was enrolled for graduate study in Electrical Engineering at The University of Maine and served as a Graduate Research Assistant in WiSe-Net Lab. Her research interests included error correction coding and wireless sensor networks. Bingxin got her Master of Science degree in Electrical Engineering from The University of Maine in August 2007.

In September 2007, she jointed the COSINE Lab in Electrical and Computer Engineering Department at Stony Brook University and started her Ph.D study in communication theory. Her current research interests include estimation and detection, particle filtering and population Monte Carlo. Bingxin is a Ph.D candidate in Electrical Engineering and expects her degree from Stony Brook University in August 2011.

REFERENCES

[1] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:543, 2003.

[2] M. S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[3] G. E. Barter, L. K. Purvis, N. P. Teclemariam, and T. H. West. Analysis of detection systems for outdoor chemical or biological attacks. In *IEEE Conference on Technologies for Homeland Security*, Boston, MA, USA, May 2009.

[4] M. J. Beal, A. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. In E. Alpaydin, editor, *Machine Learning*, pages 29–245. MIT Press, 2002.

[5] H. Bhaskar, L. Mihaylova, and S. Maskell. Population-based particle filtering. In *IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications*, Birmingham, UK, April 2008.

[6] C. Bi. A Monte Carlo EM algorithm for de novo motif discovery in biomolecular sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(3):370–386, 2009.

[7] M.C.A.M. Bink, M.P. Boer, C.J.F. Braak, J. Jansen, R.E. Voorrips, and W.E. van de Weg. Bayesian analysis of complex traits in pedigreed plant populations. *Euphytica*, 161(1):85–96, 2008.

[8] M. Bugallo, M. Hong, and P. M. Djurić. Marginalized Population Monte Carlo. In *International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, April 2009.

[9] O. Cappé, R. Douc, A. Guillin, J. M. Marin, and C. P. Robert. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459, 2008.

[10] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13:907–929, 2004.

[11] G. Casella and C. P. Robert. Rao-Blackwellization of sampling schemes. *Biometrika*, 84:81–94, 1996.

[12] G. Celeux, J. M. Marin, and C. P. Robert. Iterated importance sampling in missing data problems. *Computational Statistics & Data Analysis*, 50(12):3386–3404, 2006.

[13] K. L. Chung. *A Course in Probability Theory*. Academic Press, New York, second edition, 2000.

[14] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(2):736–746, 2002.

[15] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *10th IEEE International Symposium on High Performance Distributed Computing*, San Francisco, CA, USA, August 2001.

[16] M. De Iorio, R.C. Griffiths, R. Leblois, and F. Rousset. Stepwise mutation likelihood computation by sequential importance sampling in subdivided population models. *Theoretical Population Biology*, 68(1):41–53, 2005.

[17] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B Statistical Methodology*, 68(3):411–436, 2006.

[18] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, 2003.

[19] P. M. Djurić, T. Lu, and M. Bugallo. Multiple particle filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Honolulu, HI, USA, April 2007.

[20] P. M. Djurić, B. Shen, and M. F. Bugallo. Population Monte Carlo methodology a la Gibbs sampling. In *the Proceedings of EUSIPCO*, Barcelona, Spain, August 2011.

[21] R. Douc, A. Guillin, J. M. Marin, and C. P. Robert. Convergence of adaptive mixtures of importance sampling schemes. *Annals of statistics*, 35:420–448, 2007.

[22] R. Douc, A. Guillin, J. M. Marin, and C. P. Robert. Minimum variance importance sampling via Population Monte Carlo. *ESAIM: Probability and Statistics*, 11:427–447, 2007.

[23] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Information Science and Statistics. Springer, New York, 2001.

[24] A. Doucet, S. Sénécal, and T. Matsui. Space alternating data augmentation: Application to finite mixture of Gaussians and speaker recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, PA, USA, March 2005.

[25] J. Dreitlein. Stochastic simulation of classical lattice systems. *Physical Review*, 31:1658–1666, 1985.

[26] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:143–162, 1994.

[27] S. Fan. *Sequential Monte Carlo Methods for Physically based rendering*. PhD thesis, University of Wisconsin at Madison, 2006.

[28] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer, New York, 1995.

[29] R. Furrer and T. Bengtsson. Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants. *Journal of Multivariate Analysis*, 98:227–255, 2007.

[30] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, New York, 1995.

[31] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[32] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York, 2003.

[33] N. Gordon, J. Salmond, and A. F. M. Smith. A novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing*, 140:107–113, 1993.

[34] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. London and John & Sons, New York, 1964.

[35] S. S. Hampton and J. A. Izaguirre. Improved sampling for biological molecules using shadow hybrid Monte Carlo. In *International Conference on Computational Science*, Krakow, Poland, June 2004.

[36] J. E. Handschin and D. Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9:547–559, 1969.

[37] P. L. Hsu and H. Robbins. Complete convergence and the law of large numbers. *Proceedings of National Academic Science*, 33:25–31, 1947.

[38] Y. Iba. Population Monte Carlo algorithms. *Transactions of the Japanese Society for Artificial Intelligence*, 16(2):279–286, 2001.

[39] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[40] E. S. Jackson and W. J. Fitzgerald. A sequential Monte Carlo EM approach to the transcription factor binding site identification problem. *Bioinformatics*, 23(11):1313–1320, 2007.

[41] A. Jasra and P. D. Moral. Sequential Monte Carlo Methods for Option Pricing. *Stochastic Analysis and Applications*, 29(2):292–316, 2011.

[42] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of ASME-Journal of basic Engineering*, pages 35–45, 1960.

[43] R. Karlsson, T. Schon, and F. Gustafsson. Complexity analysis of the marginalized particle filter. *IEEE Transactions on Signal Processing*, 53:4408–4411, 2005.

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

[45] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[46] S. M. Kay and S. Saha. Mean likelihood frequency estimation. *ESAIM: Probability and Statistics*, 11:427–447, 2007.

[47] Y. R. Kim, S. K. Sul, and M. H. Park. Speed sensorless vector control of induction motor using extended kalman filter. *IEEE Transactions on Industry Applications*, 30:1225 – 1233, 1994.

[48] J. H. Kotecha and P. M. Djurić. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *Proceedings of the Acoustics, Speech, and Signal Processing Conference*, Phoenix, AZ, USA, March 1999.

[49] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 1998.

[50] Y. Lai. Adaptive Monte Carlo methods for matrix equations with applications. *Journal of Computational and Applied Mathematics*, 231(2):705–714, 2009.

[51] Y. C. Lai, S. Fan, S. Chenney, and C. Dyer. Photorealistic image rendering with population Monte Carlo energy redistribution. In *The Annual Conference of the European Association for Computer Graphics*, Prague, Szech, September 2007.

[52] D. G. Lainiotis. *Partitioning: The multi-model framework for estimation and control, I: Estimation*. Springer Berlin / Heidelberg, New York, 1979.

[53] C. Lemoine and M. Drissi. Estimating the effective sample size to select independent measurements in a reverberation chamber. *IEEE Transactions on electromagnetics*, 50:227–236, 2008.

[54] Y. Lin, Y. Peng, and X. Wang. Maximum likelihood parameter estimation of multiple chirp signals by a new Markov chain Monte Carlo approach. In *Proceedings of the IEEE Radar Conference*, Philadelphia, PA, USA, April 2004.

[55] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, New York, 2008.

[56] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44:335–341, 1949.

[57] J. Mguez and P. M. Djuric. Blind equalization of frequency selective channels by sequential importance sampling. *IEEE Transactions on Signal Processing*, 52(10):2738–2748, 2004.

[58] J. Mguez, T. Ghirmai, M. F. Bugallo, and P. M. Djuric. A sequential monte carlo technique for blind synchronization and detection in frequency-flat rayleigh fading wireless channels. *Signal Processing*, 84(11):2081–2096, 2004.

[59] M. S. Oh and J. O. Berger. Adaptive importance sampling in Monte Carlo integration. *Journal of Statistical Computation and Simulation*, 41:143–168, 1992.

[60] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Companies, New York, third edition, 1991.

[61] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[62] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2004.

[63] D. B. Rubin. A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creation a few imputations when fractions of missing information are modest: the SIR algorithm. *Journal of the American Statistical Association*, 82:543–546, 1987.

[64] L. L. Scharf. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison Wesley, Boston, MA, 1991.

[65] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas Propagation*, 34:276–280, 1986.

[66] B. Shen, M. F. Bugallo, and P. M. Djurić. Multiple marginalized population Monte Carlo. In *the Proceedings of EUSIPCO*, Aalborg, Denmark, 2010.

[67] P. Stoica, R. L. Moses, B. Friedlander, and T. Soderstrom. Maximum likelihood estimation of the parameters of multiple sinusoids from noisy measurements. *IEEE Transactions on Acoustics, Speech, and Signal Processing Conference*, 37(3), 1989.

[68] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*. John Wiley & Sons, New York, 1968.

[69] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H. J. Zhang. Image classification for content-based indexing. *IEEE Transactions on Image Processing*, 10:117–130, 2001.

[70] M. Vemula, M. Bugallo, and P.M. Djurić. Cost-based Monte Carlo sampling approaches for sensor self-localization under beacon position uncertainty. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Honolulu, HI, USA, April 2007.

[71] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

[72] G. Welch and G. Bishop. *An Introduction to the Kalman Filter*. University of North Carolina at Chapel Hill Press, 1995.

[73] S. Wilson and G. Stefanou. Bayesian approaches to content-based image retrieval. In *Proceedings of the International Workshop/Conference on Bayesian Statistics and Its Applications*, Varanasi, India, January 2005.

[74] J. Zhang and Q. Chen. Monte Carlo simulation in demography. In *3rd International Conference on Intelligent System and Knowledge Engineering*, Xiamen, China, November 2008.