LUND UNIVERSITY

Sequential Monte Carlo Methods with Applications to Positioning and Tracking in Wireless Networks

Bizjajeva, Svetlana

2008

Sequential Monte Carlo Methods with Applications to Positioning and Tracking in Wireless Networks

Svetlana Bizjajeva

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden

http://www.maths.lth.se/

Doctoral Theses in Mathematical Sciences 2008:8
ISSN 1404-0034

ISBN 978-91-628-7573-2
LUTFMS-1034-2008

Printed in Sweden by KFS AB, Lund 2008

Contents

Acknowledgements

I would like to thank all people who supported me during 5 years of my Ph.D. study.

My first thanks go to person who is worth to mention above all, my supervisor Professor Tobias Rydén, for the inspiration and professional guidance, for always having time to discuss various problems I met during the study and for extensive proof reading of all my writings.

I am also grateful to my co-supervisor Bengt Lindoff from Ericsson and to my co-author Professor Ove Edfors from the Department of Electrical and Information Technology, for opening the gates to the world of Telecommunication in front of me and for making me to feel more comfortable in this field.

I would like to thank all my past and present fellow-workers at the Department of Mathematical Statistics for the warm and friendly atmosphere. Linda, Sara, Azra, Anastassia, Fredrik, Erik, Klas, Anders, Sofia, Sebastian, Thomas, Joerg, Johan Lindström and Johan Sandberg, Susann, Mats - I thank your for the time you have spent with me during coffee-breaks and by-the-way chats. Especially thank Nadia for warm conversations during our lunches together and for extremely tasty crash-course in Italian Culinary. Finally, I would like to express my gratitude to Jimmy Olsson for always being friendly and supportive. I have learned a lot from our collaboration and from the inspiring discussions occurred during this time.

I am grateful to Mona Forsler, Aurelia Vogel and James Hakim for making things in the department's everyday life run smoothly and for their help with various non-statistical problems. As promised, I thank James once more - for lovely pictures of my family at the department Ph.D. parties.

I would like to express my warmest thanks to my family. My older brother Alexander, You know that I was always taking You as a model. My mamma, thank You for all care and support during all my life, especially last years through hundreds kilometers between Kohtla - Järve and Malmö. I also would like to thank my friends from all over the world - Alena Koubkova, Dilini Kulawansa, Vanda Nissen, Andrey Ghulchak, Anastassia Zaitseva, Kristina Pärnoja, Boris Kudimov, Aleksej Kotlov, Vera Mahrova - for the friendship and for the fun we have together.

Special thanks go to Boris Mahrov for sending me the link with announcement about Ph.D. position in Lund University in January 2003. Yes, that's how it started.

Last but not least I would like to thank Juri for his endless love, support and patience, for always believing in me and for keeping me alive during the periods of hard work. And, of course, thank my little Maria for the joy and happiness she brings into my life.

Lund, August 2008 *Svetlana Bizjajeva*

List of papers

This thesis is based on the following papers referred to in the text with the capital letters A–E.

A. Svetlana Bizjajeva and Tobias Rydén: Maximum Likelihood and Particle Filter-based Mobile Positioning from Signal Strength Measurements. In submission to *European Transactions on Telecommunications.*

B. Svetlana Bizjajeva, Tobias Rydén and Ove Edfors: Mobile Positioning in MIMO System Using Particle Filtering. Preliminary version is published in *Proc. VTC Fall 2007*, pp. 792–798.

C. Svetlana Bizjajeva and Tobias Rydén: Sequential Monte Carlo Methods and Decomposable State-Space Models. Submitted to *Communications in Statistics - Simulation and Computation.*

D. Svetlana Bizjajeva and Tobias Rydén: Sequential Monte Carlo Methods: strategies for changing the instrumental sample size.

E. Svetlana Bizjajeva and Jimmy Olsson: Antithetic Sampling for Sequential Monte Carlo Methods with Application to State Space Models. *Preprints in Mathematical Sciences 2008:14*, Lund University. In submission to *Journal of Applied Probability*.

Introduction

This thesis is concerned with the filtering problem in non–linear non–Gaussian state–space models together with the application of filtering techniques for the positioning in wireless networks and is based on 5 papers.

The state–space model is a stochastic process on two levels. On the first level there is a discrete time Markov chain, which is *not observed* directly, but only through the second level process. This second *observation* process is related to the *hidden* chain in such a way that its distribution at any time point is determined by the corresponding value of this chain.

The optimal filtering problem (or, more general, the problem of smoothing) refers to the inference about the hidden process based on the values of the observed process. In case of linear Gaussian state–space model the solution is provided by well known *Kalman filtering* technique. For non-linear Gaussian case methods like *extended Kalman filter* or *Gaussian sum filter* exist.

In case of non–linear non–Gaussian state–space model the analytical solution to the filtering problem is not feasible and approximation methods have to be employed. The *sequential Monte Carlo* methods provide an approximation of the distribution of interest and since early nineties have been widely applied in the field of non–linear filtering. These techniques (also known as *particle filters*) found a lot of applications in different areas like signal processing, automatic control, localization and tracking. Two papers of the presented thesis are devoted to the application of particle filtering to positioning of mobile devices in wireless networks.

The thesis starts with the introduction to the topic, where we define the optimal filtering problem and present basic concepts of sequential Monte Carlo methods. We give a brief overview of the wireless location and show that this problem fits the state–space model framework and can be solved using SMC methods. Second part consists of 5 papers, from applied papers A and B on wireless location to paper E, where some theoretical developments of the existing particle filtering techniques are presented.

1 State–Space Models

1.1 Definition and Examples

A *state–space model* is defined as a stochastic process on two levels. On the first level we have a discrete time Markov chain $X \triangleq \{X_n\}_{n=0}^{\infty}$ taking values in some measurable space $(\mathsf{X}, \mathcal{X})$. We call this chain the *state* or the *state process* and refer to X as the *state space*. There are no specific requirements to the structure of the state space, but in a lot of applications one takes X being subset of $\mathbb{R}^k$. Models with finite X constitute a special class which is often called in the literature *hidden Markov models*, see, for example, MacDonald and Zucchini (1997).

Term *hidden* with respect to the state process arises from the fact that X is not observed directly, but only through another stochastic process $Y \triangleq \{Y_n\}_{n=0}^{\infty}$, taking values in some measurable space $(\mathsf{Y}, \mathcal{Y})$. We call Y the *observation* or *measurement process*, and describe its connection to the state process in the following way. Given the states, all observations are conditionally independent in such a way that for any $n \geq 0$ the conditional distribution of Y_n depends on the value of X_n only.

Denote by $g_n(\cdot\,, x_n)$ the conditional density of the observations given the states, and by Q and ν the probability transition kernel and the initial distribution of the states, respectively. Assume that the kernel Q admits the density q with respect to some measure η, i.e. $Q(x, dx') = \int q(x, x')\eta(dx')$. The state–space model is then described by the set of equations, which consists of the *initial equation*

$$X_0 \sim \nu, \tag{1.1}$$

the *evolution equation*

$$X_{n+1}|X_n = x_n \sim Q(x_n, \cdot), \tag{1.2}$$

and the *observation equation*

$$Y_n|X_n = x_n \sim g_n(\cdot\,, x_n). \tag{1.3}$$

Alternatively, on can specify the state–space model in a following way,

$$\begin{cases} X_{n+1} = f(X_n, V_{n+1}), \\ Y_n = h(X_n, W_n), \end{cases} \tag{1.4}$$

where f and h are arbitrary vector-functions. The *state noise* $\{V_n\}_{n=1}^{\infty}$ and the *observation noise* $\{W_n\}_{n=0}^{\infty}$ are sequences of independent random variables with known distributions, which determine the kernel Q and density g in formulation (1.2)–(1.3), respectively.

The description above is fairly general and includes variety of models applied in the different scientific disciplines, such as telecommunications or finance. Let us consider a few examples.

Example 1.1. *Linear Gaussian State–Space Model*

This state–space model is usually considered in the relation to standard time–series analysis, see Brockwell and Davis (2002), Chapter 8 and is widely employed in engineering.
The model is given by

$$\begin{aligned} X_0 &\sim N(\mu_0, \Sigma_0), \\ X_{n+1} &= AX_n + RV_{n+1}, \\ Y_n &= BX_n + SW_n, \end{aligned} \tag{1.5}$$

where the state noise $\{V_n\}_{n=1}^{\infty}$ and the observation noise $\{W_n\}_{n=0}^{\infty}$ are sequences of independent standard (multivariate) Gaussian random variables. In terms of (1.2) and (1.3), the kernel $Q(x_n, \cdot)$ corresponds to the multivariate Gaussian distribution $N(Ax_n, RR')$ and the density $g_n(\cdot\,, x_n)$ corresponds to the density of the multivariate Gaussian distribution with mean vector Bx_n and covariance matrix SS'.

Example 1.2. *Bearings-only Tracking*

Consider the problem of tracking an object traveling in two-dimensional space. Both the position and velocity of the target are unknown, and have to be estimated using the noisy measurements of the angle between the target and the observer with known position. In this case the state vector is four–dimensional and represents horisonal and vertical positions and velocities of the target, $X_n = (X_{1,n}, \dot{X}_{1,n}, X_{2,n}, \dot{X}_{2,n})^{\mathrm{T}}$. Suppose measurements are done with time interval Δt and base the evolution equation on well-known physical relationships between position, velocity and acceleration,

$$X_{n+1} = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix} X_n + \begin{pmatrix} (\Delta t)^2/2 & 0 \\ \Delta t & 0 \\ 0 & (\Delta t)^2/2 \\ 0 & \Delta t \end{pmatrix} V_{n+1}, \tag{1.6}$$

where $\{V_n\}_{n=1}^{\infty}$ is a sequence of independent two-dimensional Gaussian random variables with zero mean vector and covariance matrix $\sigma_v^2 \mathbb{I}_2$. The measurement equation is given by

$$Y_n = \tan^{(-1)} \frac{X_{2,n} - O_{2,n}}{X_{1,n} - O_{1,n}} + W_n, \tag{1.7}$$

where $(O_{1,n}, O_{2,n})^{\mathrm{T}}$ denotes the known position of an observer and $\{W_n\}_{n=0}^{\infty}$ is a sequence of independent Gaussian random variables with zero mean and variance σ_w^2.

1.2 State Inference in State–Space Models

In this section we consider the problem of inference about the hidden process, i.e. drawing conclusions about the properties of the hidden process that usually provide solutions to problems arising in practice. For example, in bearings only tracking one would like to estimate the state vector at certain time points given the observations (measurements) up to this time. A naive way to address this problem is maximum likelihood estimation (see Casella and Berger (1990)). Suppose we are given N observations at some time point n, $y_n^{1:N} \triangleq (y_n^1, \ldots, y_n^N)^{\mathrm{T}}$ and write the full conditional log-likelihood function of the observations at this time as

$$l(y_n^{1:N}|x_n) = \sum_{i=1}^{N} \log g_n(y_n^i, x_n).$$

Then the state estimates are obtained by maximising the function above with respect to x_n. However, this approach is far from efficient and is not applicable in general settings. First, it does not take into account the underlying dynamics of states. Second, the conditional densities of observations might involve complicated multimodal functions, making the maximisation unfeasible. Moreover, there is usually only one observation avaliable at the fixed time point and the variance of ML estimates becomes quite large in this case.

Suppose we are given the observation sequence in time, $y_{0:n} \triangleq (y_0, y_1, \ldots, y_n)^{\mathrm{T}}$, and the goal is to make an inference about the distribution of the hidden states based on these observations. Denote by $\varphi_{\nu,k:l|n}$ the density of the conditional distribution of states $X_{k:l}$ given the observations $y_{0:n}$. Different choices of k and l results in several cases of interest:

- *joint smoothing* $\varphi_{\nu,0:n|n}$ for $k = 0$, $l = n$,

- *p - step prediction* $\varphi_{\nu,n+p|n}$ for $k = l = n + p$,
- *filtering* $\varphi_{\nu,n|n}$ for $k = l = n$.

Smoothing, prediction and filtering are widely explored in the literature since 1960s. However, early works on these topics, starting from Kalman and Bucy (1961) focused on the linear Gaussian state–space models. In contrast, work on smoothing in hidden Markov models was done by Baum *et al.* (1970). Until late 1990s, these two cases – linear Gaussian and finite state space models – dominated in the research except of some contributions in non-linear filtering, for example, by Handschin (1970).

1.3 Optimal Filtering Problem

The historical references to the *optimal filtering problem* date back to 1970, for example, a book by Anderson and Moore (1979). This topic receives a lot of attention within the engineering community, especially in signal processing. Within this framework the term *filtering* is often used in the following sense. Suppose there exists a *system* of which noisy *measurements* are avaliable, then the filtering means the recovery of the system from the measurements. A classical example of filtering is signal transmission, when the signal being the sequence of bits is transmitted to the receiver. The received signal is obviously corrupted by noise, and has to be filtered to recover the transmitted sequence as well as possible.

In the statistical community, the optimal filtering problem consists in computation of the filtering distribution at time n based on measurements up to this time. This can be done by iterative evaluation of one-step prediction

$$\varphi_{\nu,n+1|n}(x_{n+1}) \triangleq \mathbb{p}(x_{n+1}|y_{0:n})$$

and filtering

$$\varphi_{\nu,n|n}(x_n) \triangleq \mathbb{p}(x_n|y_{0:n})$$

densities[1]. We start the iterations from the initial distribution,

$$\varphi_{\nu,0|-1}(x_0) = \nu(x_0),$$

[1] Here $\mathbb{p}$ is a generic symbol for density.

and for the filtering density application of Bayes' theorem yields at $n = 0$

$$\begin{aligned}\varphi_{\nu,0|0}(x_0) = \mathbb{P}(x_0|y_0) &= \frac{\mathbb{P}(y_0|x_0)\mathbb{P}(x_0|y_{-1})}{\mathbb{P}(y_0|y_{-1})} \\ &= \frac{\mathbb{P}(y_0|x_0)\mathbb{P}(x_0|y_{-1})}{\int_{\mathsf{X}} \mathbb{P}(y_0|x)\mathbb{P}(x|y_{-1})dx} = \frac{g_0(x_0)\,\nu(x_0)}{\int_{\mathsf{X}} g_0(x)\,\nu(x)dx}.\end{aligned} \tag{1.8}$$

Similarly, for $n \geq 0$,

$$\begin{aligned}\varphi_{\nu,n+1|n}(x_{n+1}) = \mathbb{P}(x_{n+1}|y_{0:n}) &= \int_{\mathsf{X}} \mathbb{P}(x_{n+1}|x_n)\mathbb{P}(x_n|y_{0:n})dx_n \\ &= \int_{\mathsf{X}} q(x_n, x_{n+1})\,\varphi_{\nu,n|n}(x_n)\,\eta(dx_n), \\ \varphi_{\nu,n+1|n+1}(x_{n+1}) &= \frac{g_{n+1}(x_{n+1})\,\varphi_{\nu,n+1|n}(x_n)}{\int_{\mathsf{X}} g_{n+1}(x'_{n+1})\,\varphi_{\nu,n+1|n}(x'_{n+1})\,\eta(dx'_{n+1})}.\end{aligned} \tag{1.9}$$

Note, that here and in following we use shortened notation $g_n(x_n) \triangleq g_n(y_n, x_n)$ for $n \geq 0$.

The intuitive explanation behind this recursion is quite straightforward. One-step prediction is obtained by passing the value filtered at the previous step through the dynamic of the state model. Then, for the filtering, one uses the conditional density of the observation at the given time point as a kind of weight for the predicted values.

The recursive solution to the filtering problem seems to be attractively simple, but the implementation of (1.9) is complicated by the presence of complex multidimensional integrals. The analytical evaluation of the involved integrals is possible in only a few special cases. The most important case which has been successively applied in various fields are the models with linear Gaussian structure. Filtering and one-step prediction algorithm for models of this kind is widely known as a *Kalman filter*, derived by Kalman (1960). Due to the properties of the Gaussian distribution, both the filter and the one-step predictor are Gaussian and the filtering recursion is reduces to the sequential update of corresponding means and variances. Another special case are hidden Markov models, for which the integrals in (1.9) are replaced by sums. For filtering algorithms in this context we refer to works by Baum *et al.* (1970) and Rabiner (1989).

In case of non–linear (non Gaussian) state–space model analytical solution to the filtering problem is not avaliable. Since 1960s many works have been

devoted to finding approximate solution. Proposed methods include the *extended Kalman filter* (Anderson and Moore (1979)), the *Gaussian sum filter* (Sorenson and Alspach (1971)) and grid–based methods (Bucy and Senne (1971)). In late 1990s, the great increase of computational power entailed a rapid growth of numerical integration methods for optimal filtering, called *sequential Monte Carlo* methods.

2 Sequential Monte Carlo Methods

During the last decades, sequential Monte Carlo methods – alternatively termed *particle filters* – have received much attention as a powerful tool for finding an approximate solution to the optimal filtering problem. SMC methods constitute a class of simulation-based algorithms which approximate recursively a sequence of target measures by a sequence of empirical distributions associated with properly weighted samples of *particles*. Originated from work of Gordon *et al.* (1993), sequential Monte Carlo techniques have been widely applied within wide range of scientific disciplines. For early developments of the SMC methods from a practical point of view we refer to Doucet *et al.* (2001), and for more extensive treatment of these algorithms both in theory and practice see Del Moral (2004).

2.1 Sequential Importance Sampling and Resampling

Denote by μ the *target distribution*, which is a probability measure of interest on a measurable space $(\mathsf{X}, \mathcal{X})$. Usually this measure is known up to a normalising constant and our aim is to approximate this measure and integrals of the form

$$\mu(f) = \int_{\mathsf{X}} f(x)\mu(dx), \tag{2.1}$$

where f is a real-valued measurable function. For example, with $f(x) = x$ we recover a problem of estimating the mean, and with $f(x) = (x - \mathbb{E}(x))^2$ one can estimate the variance. If it is possible to sample from the target distribution, the Monte Carlo estimates for integrals of this type are obtained by first drawing the sample $\{\xi^i\}_{i=1}^N$ from μ and then evaluating sample means

$$\hat{\mu}_{MC}(f) = \frac{1}{N}\sum_{i=1}^{N} f(\xi^i). \tag{2.2}$$

The *importance sampling* (IS) technique comes to the assistance when sampling from the target distribution is not feasible. Instead one chooses another measure λ, which is easy to simulate from and such that μ is absolutely continuous with respect to λ. This measure is referred to as the *importance sampling distribution*.

Given the i.i.d. sample $\left\{\xi^i\right\}_{i=1}^N$ of particles from the importance distribution, the empirical importance sampling estimate of the target distribution is given by

$$\hat{\mu}_{\mathrm{IS},N}(dx) = \sum_{i=1}^{N} \frac{\omega^i}{\sum_{j=1}^N \omega^j} \delta_{\xi^i}(dx), \tag{2.3}$$

where δ_{ξ^i} is the delta-Dirac mass located at the particle ξ^i, and ω^i denotes the *importance sampling weight* associated with this particle. The importance weights are proportional to the Radon-Nikodym derivative of the target measure with respect to the instrumental measure, $\omega^i \propto \frac{d\mu}{d\lambda}(\xi^i)$. Recall that μ is absolutely continuous with respect to λ, i.e. for any μ-integrable function f,

$$\mu(f) = \int \frac{d\mu}{d\lambda}(x) f(x) \lambda(dx). \tag{2.4}$$

Then the importance sample estimates for integrals of type (2.1) are given by the corresponding weighted sample means,

$$\hat{\mu}_{\mathrm{IS},N}(f) = \sum_{i=1}^{N} \frac{\omega^i}{\sum_{j=1}^N \omega^j} f(\xi^i). \tag{2.5}$$

Due to the normalisation, both estimators (2.3) and (2.5) are free from any scaling factor in the Radon-Nikodym derivative and can be used when either target or importance distribution are known up to a constant only.

The IS algorithm can be modified to allow sequential implementation when targeting the *sequence* of measures $\{\mu_n\}_{n=0}^{\infty}$, which satisfies the recursion

$$\mu_{n+1}(dx_{0:n+1}) = \mu_n(dx_{0:n}) \mu_{n+1}(dx_{n+1}|x_{0:n}). \tag{2.6}$$

This is done by selecting the importance distribution of the similar form,

$$\lambda_{n+1}(dx_{0:n+1}) = \lambda_n(dx_{0:n}) \lambda_{n+1}(dx_{n+1}|x_{0:n}), \tag{2.7}$$

so that new particles $\left\{\xi^i_{0:n+1}\right\}_{i=1}^N$ are obtained by drawing the last component ξ^i_{n+1} from the conditional distribution $\lambda_{n+1}(dx_{n+1}|\xi^i_{0:n})$.

It is easy to see that the importance weights are then sequentially updated according to

$$\begin{aligned}\omega^i_{n+1} &= \frac{d\mu_{n+1}}{d\lambda_{n+1}}(\xi^i_{0:n+1}) = \frac{d\mu_n}{d\lambda_n}(\xi^i_{0:n})\frac{d\mu_{n+1}(\xi^i_{0:n},\cdot)}{d\lambda_{n+1}(\xi^i_{0:n},\cdot)}(\xi^i_{n+1}) \\ &= \omega^i_n \frac{d\mu_{n+1}(\xi^i_{0:n},\cdot)}{d\lambda_{n+1}(\xi^i_{0:n},\cdot)}(\xi^i_{n+1}).\end{aligned} \tag{2.8}$$

The estimates of the target distribution and corresponding integrals are again the empirical distribution of the weighted sample,

$$\hat{\mu}_{n,\mathrm{SIS},N}(dx_{0:n}) = \sum_{i=1}^N \frac{\omega^i_n}{\sum_{j=1}^N \omega^j_n}\delta_{\xi^i_{0:n}}(dx_{0:n}), \tag{2.9}$$

and the weighted sample means

$$\hat{\mu}_{n,\mathrm{SIS},N}(f) = \sum_{i=1}^N \frac{\omega^i_n}{\sum_{j=1}^N \omega^j_n} f(\xi^i_{0:n}), \tag{2.10}$$

as in (2.3) and (2.5).

The sequential importance sampling (SIS) algorithm is attractively easy to implement, and is applicable for a wide class of distributions irrespectively of their shape. However, it suffers from a serious drawback known as *weight degeneracy*. For longer time records, the distribution of the importance weights becomes more and more skewed. This means that after several time iterations only a few particles have non-zero weights and the representation of the distribution of interest is not any longer adequate. To avoid degeneracy, one should use the *resampling procedure*, proposed by Gordon *et al.* (1993). The idea of resampling procedure is quite simple and consists of eliminating particles having relatively small importance weights and duplicating particles with relatively large importance weights. Denote by M^i_n the number of offspring of particle $\xi^i_{0:n}$. This number is selected in a way that $\sum_{i=1}^N M^i_n = N$ and the weighted empirical distribution and the empirical distribution of selected particles produce the same approximations to

the target measure and to the integrals with respect to the target measure, i.e.

$$\begin{aligned}\hat{\mu}_{n,\mathrm{SISR},N}(dx_{0:n}) &= \frac{1}{N}\sum_{i=1}^{N} M_n^i \delta_{\xi_{0:n}^i}(dx_{0:n}) \approx \sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{j=1}^{N}\omega_n^j}\delta_{\xi_{0:n}^i}(dx_{0:n}),\\ \hat{\mu}_{n,\mathrm{SISR},N}(f) &= \frac{1}{N}\sum_{i=1}^{N} M_n^i f(\xi_{0:n}^i) \approx \sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{j=1}^{N}\omega_n^j} f(\xi_{0:n}^i).\end{aligned} \tag{2.11}$$

The most popular way to select the particles is introduced in the original paper by Gordon *et al.* (1993), and consists in resampling with replacement from a multinomial distribution with probabilities of selection given by the normalised importance weights. Note that it is not necessary to include the resampling step at each time iteration, but only at time points when the weights start to degenerate. The simplest criterion for degeneracy is the *coefficient of variation* used by Kong *et al.* (1994),

$$\mathrm{CV}_{N,n} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(N\frac{\omega_n^i}{\Omega_n} - 1\right)},$$

where we denote by Ω_n the normalising sum, $\Omega_n \triangleq \sum_{j=1}^{N}\omega_n^j$. The related criterion, called the *effective sample size*

$$N_{\mathrm{e},n} = \left[\sum_{i=1}^{N}\left(\frac{\omega_n^i}{\Omega_n}\right)^2\right]^{-1}$$

have been widely applied in practice due to its simple interpretation.

2.2 Application of SISR to the Optimal Filtering

In this section we describe the sequential Monte Carlo solution to the optimal filtering problem. The importance sampling procedures, being introduced at quite general level, are applied to the larger problem of estimating joint smoothing distributions with the densities $\varphi_{\mathrm{v},0:n|n} \triangleq \mathbb{p}(x_{0:n}|y_{0:n})$. After that, the estimates of the filtering distribution are obtained by restricting the approximate joint smoothing distribution to its last component.

Using Bayes' theorem one can derive the following form of the joint smoothing distribution. For the particular sequence of observations,

$$\varphi_{\nu,0:n|n}(dx_{0:n}) = L_n^{-1}\nu(dx_0)g_0(x_0)\prod_{k=1}^{n} Q(x_{k-1}, dx_k)g_k(x_k),$$

where L_n denotes the full likelihood of the observations,

$$L_n(y_{0:n}) = \int \dots \int \nu(dx_0)g_0(x_0)\prod_{k=1}^{n} Q(x_{k-1}, dx_k)g_k(x_k).$$

From this point we drop the initial distribution of the chain ν from the notation since the dependence of the joint smoothing distribution with respect to the initial distribution is out of importance here. We write the recursive update for the joint smoothing distribution in a following way. Starting from the initial smoothing distribution,

$$\varphi_0(dx_0) = \frac{\nu(dx_0)g_0(x_0)}{\int \nu(dx)g_0(x)}, \tag{2.12}$$

for $n = 0, 1, 2, \dots$ we obtain

$$\varphi_{0:n+1|n+1}(dx_{0:n+1}) = \varphi_{0:n|n}(dx_{0:n})T_n^{\mathrm{u}}(x_n, dx_{n+1}), \tag{2.13}$$

where T_n^{u} is the unnormalised transition kernel defined by

$$T_n^{\mathrm{u}}(x, dx') = \left(\frac{L_{n+1}}{L_n}\right)^{-1} Q(x, dx')g_{n+1}(x'). \tag{2.14}$$

The likelihood ratio in (2.14) is not computable in a closed form, and the problem of computing the joint smoothing distribution perfectly fits into the sequential Monte Carlo framework with $\mu_{n+1}(dx_{0:n+1}) = \varphi_{0:n+1|n+1}(dx_{0:n+1})$ and $\mu_{n+1}(dx_{n+1}|x_{0:n}) = T_n^{\mathrm{u}}(x_n, dx_{n+1})$.

Consequently, the importance sampling distribution should satisfy a recursion of form (2.7). Denote by $\{\rho_{0:n}\}_{n=0}^{\infty}$ the family of probability measures associated with the inhomogeneous Markov chain with initial distribution ρ_0 and transition kernels $\{R_n\}_{n=0}^{\infty}$,

$$\rho_{0:n}(dx_{0:n}) = \rho_0(dx_0)\prod_{k=0}^{n-1} R_k(x_k, dx_{k+1}). \tag{2.15}$$

Assume that the initial distribution is absolutely continuous with respect to the initial importance sampling distribution and, moreover, the unnormalised kernels are absolutely continuous with respect to the importance sampling kernels. Inserting $\mu_{n+1}(dx_{n+1}|x_{0:n}) = T_n^{\mathrm{u}}(x_n, dx_{n+1})$ and $\lambda_{n+1}(dx_{n+1}|x_{0:n}) = R_n(x_n, dx_{n+1})$ into (2.8) we obtain following update for the importance weights,

$$\omega_{n+1}^i = \omega_n^i \frac{dT_n^{\mathrm{u}}(\xi_n^i, \cdot)}{dR_n(\xi_n^i, \cdot)}(\xi_{0:n+1}^i) = \omega_n^i g_{n+1}(\xi_{k+1}^i) \frac{dQ(\xi_n^i, \cdot)}{dR_n(\xi_n^i, \cdot)}(\xi_{n+1}^i) \tag{2.16}$$

with the initial weights given by $\omega_0^i = \frac{d\varphi_0}{d\rho_0}(\xi_0^i)$. At each time step, approximations of the joint smoothing distributions and corresponding integrals are calculated according to (2.11). Approximations of the filtering distribution and integrals w.r.t. the filtering distribution are given by the marginal estimates,

$$\begin{aligned} \hat{\varphi}_{n|n,\mathrm{SISR},N}(dx_n) &= \sum_{i=1}^N \frac{\omega_n^i}{\Omega_n} \delta_{\xi_n^i}(dx_n), \\ \hat{\varphi}_{n|n,\mathrm{SISR},N}(f) &= \sum_{i=1}^N \frac{\omega_n^i}{\Omega_n} f(\xi_n^i). \end{aligned} \tag{2.17}$$

Note, that for the sequential approximation of the filtering distribution one does not have to store all particle trajectories obtained up to the current time point. This enables on-line implementation of the algorithm, updating current estimates at the moment when a new observation becomes avaliable.

The general algorithm for approximating joint smoothing distribution via SISR procedure is presented in a following scheme.

Algorithm 1: standard SISR algorithm

- `Initialisation: Draw an i.i.d. sample` $\xi_0^1, \ldots, \xi_0^N$ `from` ρ_0 `and set`

$$\omega_0^i = g_0(\xi_0^i) \frac{d\varphi_0}{d\rho_0}(\xi_0^i), \quad i = 1, \ldots, N.$$

`For` $n = 0, 1, \ldots$

- `Sampling: Draw` $\bar{\xi}_{n+1}^1, \ldots, \bar{\xi}_{n+1}^N$ `conditionally independently given` $\{\xi_{0:n}^i\}_{i=1}^N$ `from the importance distribution`

$$\bar{\xi}_{n+1}^i \sim R_n(\xi_n^i, \cdot), \quad i = 1, \ldots, N.$$

Compute the updated importance weights

$$\omega_{n+1}^i = \omega_n^i g_{n+1}(\bar{\xi}_{n+1}^i) \frac{dQ(\xi_n^i, \cdot)}{dR_n(\xi_n^i, \cdot)}(\bar{\xi}_{n+1}^i) \quad i = 1, \ldots, N.$$

- Resampling (optional): Draw, conditionally independently given $\{\xi_{0:n}^i, \bar{\xi}_{n+1}^j\}_{i,j=1}^N$ the indices $I_{n+1}^1, \ldots, I_{n+1}^N$ from the multinomial distribution with probabilities

  $$\frac{\omega_{n+1}^1}{\Omega_{n+1}}, \ldots, \frac{\omega_{n+1}^N}{\Omega_{n+1}}.$$

  Reset all importance weights to a constant value. If the resampling step is not applied, set $I_{n+1}^i = i,\ i = 1, \ldots, N$.

- Trajectory update: Set $\xi_{0:n+1}^i = (\xi_{0:n}^{I_{n+1}^i}, \bar{\xi}_{n+1}^{I_{n+1}^i})$.

The importance sampling and resampling steps are common for all particle filters and can be accomplished in various ways, see the collection of Doucet *et al.* (2001) for different variations of the basic scheme and many applications of state-space models. Often particle filtering are referred to as *genetic type* algorithms, with the importance sampling step as *mutation* step, followed with the *selection* procedure.

Two key issues of the particle filtering algorithms are the choice of the proposal kernel for the mutation of the particle swarm and choice of the resampling scheme in the selection step. In the following sections we will briefly describe various alternatives to address the first issue and conclude the chapter with the short overview of the basic convergence results for the particle filtering estimates.

Regarding the resampling schemes, most straightforward is the multinomial resampling, and drawing N random indices $I^1, \ldots, I^N$ conditionally independently given the $\mathcal{G}$[2] from the set $\{1, \ldots, M_N\}$ such that $\mathbb{P}(I^j = i|\mathcal{G}) = \tilde{\omega}^i$ is usually conducted by the inverse transform method.

There are several variations of the standard resampling scheme reducing its computational complexity. These include residual resampling (Liu and Chen

[2]We denote by $\mathcal{G}$ a σ-field such that all current particles and normalised importance weights are $\mathcal{G}$-measurable.

(1998)), stratified resampling (Kitagawa (1996)) and systematic resampling (Carpenter *et al.* (1999)). For overview of these schemes we refer to Cappé *et al.* (2005), pp. 242-250 and references therein.

Note, that the number of particles sampled in the mutation step might be larger, than the size of the final sample. In other words, one can draw an instrumental sample of size αN from the instrumental distribution and resample only N particles in the selection step. The use of multiple offspring has been suggested by Rubin (1987) and has been subsequently taken into account in studying the asymptotic properties of importance sampling/resampling estimates, see, for example Cappé *et al.* (2005) or Douc and Moulines (2005).

The simplest way to introduce multiple offspring is to mutate several times independently from each of the initial particles. In paper D we apply this scheme to the standard bootstrap filter and suggest to modify the sampling step introducing the correlation between particles mutated from the same ancestor. We expect that the correlated particle cloud will explore state-space in a more systematic way, leading to the improvement of the standard scheme with independent mutations in terms of mean squared error. In addition, we investigate the scheme with number of offspring different for different initial particles and let this number to be determined by the observations. Last is carried out via *two-stage sampling* procedure, which will be considered in more details in the end of next section.

In paper E we investigate correlated mutation scheme on a theoretical level and prove that if the importance weights are close to uniform and the correlation structure of each block is negative, the asymptotic variance of the obtained estimates is decreased.

2.3 Choice of the Instrumental Kernel

2.3.1 Prior Kernel

The most obvious and simple choice of the importance kernel is setting $R_n = Q,\ \forall n \geq 0$, i.e. to propagate the particles according to the distribution of the hidden chain. In this case the particle filter is referred to as the *bootstrap particle filter*. This filter is easy to implement, and very adaptable in a sense that when changing the problem one needs only to change the expressions for the importance distribution and the importance weights in the code. Note that updating coefficient for the importance weights are reduced to the conditional density of

the observations given the current particle,

$$\omega_{n+1}^i = \omega_n^i g_{n+1}(\xi_{n+1}^i),$$

and does not depend on the previous position of the particles.

Despite of its simplicity the bootstrap filter suffers from a serious drawback. Namely, this filter moves the particle cloud "blindly" in the state space without taking into account the observations. Such way of propagation leads to poor performance if there is a large mismatch between the prior and the posterior distributions, for example, if the sequence of observations contains the outliers. In this case under the prior kernel all particles might be propagated to the region where the conditional density of the observation is low, and even the particles with large normalised weights might be not important for the distribution of interest.

2.3.2 Optimal Kernel

The exact opposite way to sampling from the prior kernel is to propagate the particles according to the normalised version of the optimal kernel (2.14),

$$T_n(x, dx') = \frac{Q(x, dx')g_{n+1}(x')}{\int_{\mathsf{X}} Q(x, dx')g_{n+1}(x')}. \tag{2.18}$$

The optimal kernel incorporates the information both on the dynamics of states and on the current observation. The increment of the importance weight in one time step is now equal to the normalising coefficient,

$$\omega_{n+1}^i = \omega_n^i \int Q(\xi_n^i, d\xi_{n+1}^i)g_{n+1}(\xi_{n+1}^i),$$

and depend on the previous position of the particle only. While sampling from the optimal kernel, particles tend to cluster in the regions with high conditional density of the observations, providing more robust estimates.

However, sampling directly form the optimal kernel might not be feasible and the expression for the importance weights might not be computable in the closed form. Different approaches for sampling from the optimal kernel include local approximations, introduced as the *auxiliary particle filter* by Pitt and Shephard (1999). This algorithm is applied when the optimal distribution is unimodal with the mode located in some way. The idea is first to locate the high-density regions of the optimal distribution and then to imitate sampling from the optimal

kernel by drawing from some distribution with more heavy tails, like multidimensional t-distribution with l degrees of freedom. The mean of such instrumental distribution is fitted to the mode of the optimal distribution and covariance matrix is set to the negative inverse of the Hessian of the optimal log-density evaluated at the mode.

For further developments on the choice of the proposal distribution we refer to recent works by Olsson *et al.* (2007) and Cornebise *et al.* (2008). These papers are based on the two-stage sampling procedure, which serves to form an idea of the values of the importance weights before the particles are mutated, and introduces an additional resampling pass in order to select most promising particles.

2.3.3 Two–Stage Sampling

To set up the two-stage sampling procedure we have to reformulate the problem of estimating joint smoothing distribution discussed in the section 3.2., following the lines of Cappé *et al.* (2005), chapter 8.

Recall the recursive form (2.13) of the joint smoothing density and rewrite it inserting the normalised version of the optimal kernel,

$$\varphi_{0:n+1|n+1}(dx_{0:n+1}) = \varphi_{0:n|n}(dx_{0:n}) \left(\frac{L_{n+1}}{L_n} \right)^{-1} \gamma_n(x_n) T_n(x_n, dx_{n+1}), \quad (2.19)$$

where $\gamma_n(x_n) \triangleq \int Q(x_n, dx_{n+1}) g_{n+1}(x_{n+1})$. Since the equation above contains an unknown likelihood ratio, it is preferable to write the expression in auto-normalised form,

$$\varphi_{0:n+1|n+1}(dx_{0:n+1}) = \frac{\varphi_{0:n|n}(dx_{0:n}) \gamma_n(x_n) T_n(x_n, dx_{n+1})}{\int \dots \int \varphi_{0:n|n}(dx_{0:n}) \gamma_n(x_n)}. \quad (2.20)$$

Now plug in the importance sampling estimate of the joint smoothing distribution at time n, $\hat{\varphi}_{0:n|n,N}(dx_n) = \sum_{i=1}^{N} \frac{\omega_n^i}{\sum_{j=1}^{N} \omega_n^j} \delta_{\xi_{0:n}^i}(dx_{0:n})$, yielding

$$\bar{\varphi}_{0:n+1|n+1}(dx_{0:n+1}) \triangleq \sum_{i=1}^{N} \frac{\omega_n^i \gamma_n(\xi_n^i)}{\sum_{j=1}^{N} \omega_n^j \gamma_n(\xi_n^j)} \delta_{\xi_{0:n}^i}(dx_{0:n}) T_n(\xi_n^i, dx_{n+1}). \quad (2.21)$$

The equation above defines a finite mixture distribution which restriction to $n+1$ component is a weighted empirical distribution with support $\left\{ \xi_{0:n}^i \right\}_{i=1}^{N}$

and weights proportional to $\omega_n^i \gamma_n(\xi_n^i)$. Sampling from this mixture is carried out by first sampling the trajectory $\xi_{0:n}^I$ with probability proportional to $\omega_n^I \gamma_n(\xi_n^I)$ and appending a $(n+1)$-st component drawn from the optimal distribution $T_n(\xi_n^I, \cdot)$. The obvious attempt to simulate new particle trajectories from this mixture distribution, unfortunately, cannot be accomplished so easily because optimal kernels are usually not avaliable in the closed form. Instead one can again employ the importance sampling approach, approximating the finite mixture distribution (2.21) as close as possible. A reasonable choice is a measure of form

$$\rho_{0:n+1}(dx_{0:n+1}) = \sum_{i=1}^{N} \frac{\omega_n^i v_n^i}{\sum_{j=1}^{N} \omega_n^j v_n^j} \delta_{\xi_{0:n}^i}(dx_{0:n}) R_n(\xi_n^i, dx_{n+1}), \tag{2.22}$$

where v_n^i are some strictly positive weights called *adjustment multiplier weights* as in Pitt and Shephard (1999), and R_n is a Markovian transition kernel. Assume that optimal kernels T_n are absolutely continuous with respect to the kernels R_n, then the target distribution that defined in (2.21) is dominated by the instrumental distribution $\rho_{0:n+1}$ with Radon-Nikodym derivative

$$\begin{aligned} \frac{d\bar{\varphi}_{0:n+1|n+1}}{d\rho_{0:n+1}}(x_{0:n+1}) &= C_n \sum_{i=1}^{N} \mathbb{1}_{\{\xi_{0:n}^i\}}(x_{0:n}) \frac{\gamma_n(\xi_n^i)}{v_n^i} \frac{dT_n(\xi_n^i, \cdot)}{dR_n(\xi_n^i, \cdot)}(x_{n+1}) \\ &= C_n \sum_{i=1}^{N} \mathbb{1}_{\{\xi_{0:n}^i\}}(x_{0:n}) \frac{g_{n+1}(x_{n+1})}{v_n^i} \frac{dQ(\xi_n^i, \cdot)}{dR_n(\xi_n^i, \cdot)}(x_{n+1}), \end{aligned} \tag{2.23}$$

where

$$C_n = \frac{\sum_{j=1}^{N} \omega_n^j v_n^j}{\sum_{j=1}^{N} \omega_n^j \gamma_n(\xi_n^j)}.$$

The adjustment multiplier weights serve to sample in the first stage particle trajectories that are most likely under $\bar{\varphi}_{0:n+1|n+1}$, and usually depend on the new observation. The suggestion of Pitt and Shephard (1999) is to set the adjustment multiplier weights to the conditional density of the observation given the mean of the predictive distribution corresponding to each particle,

$$v_n^i = g_{n+1}(\int x Q(\xi_n^i, dx)). \tag{2.24}$$

The general two-stage sampling procedure is described in the following scheme.

Algorithm 2: two-stage algorithm

First stage sampling

- Draw $I_n^1, \ldots, I_n^{M_N}$ conditionally i.i.d. given $\{\xi_{0:n}^i\}_{i=1}^N$ with probabilities $\mathbb{P}(I_n^1 = j) \propto \omega_n^j v_n^j$, for $j = 1, \ldots, M_N$.
- Draw $\bar{\xi}_{n+1}^1, \ldots, \bar{\xi}_{n+1}^{M_N}$ conditionally independently given $\{\xi_{0:n}^i\}_{i=1}^N$ and $\{I_n^i\}_{i=1}^{M_N}$ from the importance distribution $\bar{\xi}_{n+1}^i \sim R_n(\xi_n^{I_n^i}, \cdot)$. Set $\bar{\xi}_{0:n+1}^i = (\xi_{0:n}^{I_n^i}, \bar{\xi}_{n+1}^i)$ for $i = 1, \ldots, M_N$.
- For $i = 1, \ldots, M_N$ compute the second-stage weights

$$\tau_n^j = \frac{g_{n+1}(\bar{\xi}_{n+1}^i)}{v_n^{I_n^i}} \frac{dQ(\xi_n^{I_n^i}, \cdot)}{dR_n(\xi_n^{I_n^i}, \cdot)}(\bar{\xi}_{n+1}^i).$$

Second-stage resampling

- Draw $J_{n+1}^1, \ldots, J_{n+1}^N$ conditionally i.i.d. given $\{\bar{\xi}_{0:n+1}^i\}_{i=1}^M$ with probabilities $\mathbb{P}(J_n^1 = j) \propto \tau_n^j$, for $j = 1, \ldots, M_N$.
- Set $\xi_{0:n+1}^i = \bar{\xi}_{0:n+1}^{J_{n+1}^i}$ and $\omega_{n+1}^i = 1$ for $i = 1, \ldots, N$.

The theoretical properties of the weighted sample produced in the two-stage sampling procedure have been investigated by Olsson *et al.* (2006). In this work authors established several convergence results for the estimates obtained by the two-stage sampling algorithm and suggested the possible improvements of the basic scheme leading to the decrease in the asymptotic variance of the estimates.

2.4 Convergence Results

Several convergence results have been developed for the estimates produced by the SISR algorithm during last years. In this section we will consider the filtering problem only, despite the fact that asymptotic for sequential Monte Carlo estimates is avaliable on much more general level within the framework of interacting particle systems, which is extensively explored by Del Moral (2004). Some

existing results related to the filtering algorithms are briefly described in Crisan and Doucet (2002).

Consider the SISR estimate $\hat{\varphi}_{n|n,N}$ of the filtering distribution $\varphi_{n|n}$ at time n. We start from the almost sure convergence, firstly explored by Del Moral (1996). For the kernel K on $(\mathsf{X}, \mathcal{X})$ and real measurable function f on $(\mathsf{Y}, \mathcal{Y})$ define the real measurable function Kf on $(\mathsf{X}, \mathcal{X})$ by

$$Kf(x) \triangleq \int_{\mathsf{X}} K(x, dy) f(y).$$

Theorem 2.1. *Assume that the densities of the observations g_n are bounded, continuous and strictly positive for all $n \geq 0$, and for any continuous bounded function g, Qg is also a continuous and bounded function.*
Then

$$\lim_{N \to \infty} \hat{\varphi}_{n|n,N} = \varphi_{n|n} \tag{2.25}$$

almost surely.

Next question to address is the magnitude of the estimation error. Results concerning bounds on the bias of the estimates were developed in Del Moral and Guionnet (2001) and latter bounds on L^p errors were obtained in Del Moral and Miclo (2000). Denote the L_p-norm of a random variable η by $\|\eta\|_p \triangleq (\mathbb{E}|\eta|^p)^{1/p}$.

Theorem 2.2. *For any time-point $n \geq 0$ and any $p \geq 1$ there exists a finite constant $C_n^{(p)}$ such that*

$$\|\hat{\varphi}_{n|n,N}(f) - \varphi_{n|n}(f)\|_p \leq \frac{1}{\sqrt{N}} C_n^{(p)} \|f\|_\infty \tag{2.26}$$

for all real-valued bounded measurable functions f on $(\mathsf{X}, \mathcal{X})$, where $\|f\|_\infty$ is the supremum norm.

Under certain assumptions on the transition kernel it is possible to derive a time-uniform bound for the L^p-norm, i.e. to find D such that $D_n = D \; \forall n$. These assumptions are typically fulfilled when X is compact and will not be considered here. Recently, bounds on the bias of the bootstrap particle filter were studied in the work by Olsson and Rydén (2006).

Weak convergence of the particle filtering estimates was extensively explored by Chopin (2004), who established the central limit theorem as in following.

Theorem 2.3. *Assume that the conditional densities of the observations, g_n, are bounded for $n \geq 0$. Then for any measurable function $f \in \mathsf{L}^2(\mathsf{X}, \varphi_{n-1|n-1})$*

$$\sqrt{N}(\hat{\varphi}_{n|n,N} - \varphi_{n|n}) \xrightarrow{D} N(0, \sigma^2(f)), \text{ as } N \to \infty, \tag{2.27}$$

where $\mathsf{L}^2(\mathsf{X}, \varphi_{n-1|n-1})$ is the set of functions which second order moments with respect to distribution $\varphi_{n-1|n-1}$ are finite.

All asymptotic results for particle filtering estimates are now days avaliable as a special case of more general theoretical developments concerning the asymptotic properties of weighted samples. For further reading in this topic we refer to the recent paper by Douc and Moulines (2005) and to the book by Cappé *et al.* (2005), chapter 9.

3 Location of Mobile Devices in Wireless Networks

3.1 Wireless Location: Definition and Applications

The general term *wireless location* refers to the determining the position of a Mobile Station (MS) in Wireless Local Area Networks (WLAN) or in cellular networks, operated by Base Transceiver Stations (BTS). In practice distinguish two main cases of location: *self-positioning*, when MS determines its own coordinates, and *target tracking*, where the aim is to determine the unknown target position. Such dichotomy is irrelevant for the present section and in following we will joint these terms under the name "wireless location" describing the general problem of estimating the MS position based on the measurements of a certain type, avaliable either on observer (network) or target side.

In the network–based positioning the estimates of a mobile coordinates are obtained using measurement avaliable on the BTSs. This technology relies on an existing networks and does not involve the MS into the positioning process, with an advantage of using existing handsets without any modification. An overview of existing network–based location methods can be found in Sayed *et al.* (2005).

In the mobile–based positioning the MS determines its own position using measurements either form BTS or form the global positioning system (GPS). The advantage of this method is that it allows to use some additional information, for example, the own speed in the positioning of a moving object. However, the implementation of the mobile–based techniques requires the replacement of an existing equipment and in some cases may increase cost, size and battery consumption of a handset. The costly MS with built-in GPS receivers are able to estimate the own position with high degree of accuracy in outdoor applications, but the accuracy degrades in indoor or urban environment.

Broad and active research in wireless location was provoked by the order of United States Federal Communication Commission (FCC) issued in 1996. According to the FCC requirements all wireless service providers have to be able to report accurate location information of an emergency caller to the emergency answering points. Due to the widespread of mobile phones among the people, a hight rate of emergency calls originate from mobile stations. In contrast to a fixed-network user, the location of a mobile caller is unknown, which leads to the lower quality of an emergency assistance. Thus the positioning problem becomes very important in public safety sector. Another application in this area is *asset tracking*, for example, location of lost children, patients or pets.

The wireless location can also be used to track personnel in a hospital or a manufacturing site to provide more efficient management. It can serve as a base for interactive tour guides, smart shopping guides that direct shoppers based on their location in a store, and traffic controls in parking structures that guides cars to free parking slots.

Many fleet operators, such as police forces, emergency cars and shuttle or taxi companies may use wireless location for tracking and operating their vehicles in order to minimize response time. The tracking of drivers on roads and highways, which carry the mobile phones while driving might be used to provide real-time traffic information and maintain the transportation safety. Another application is location sensitive billing, when wireless services providers can offer flexible call plans or rates or services based on the caller location.

Finally, wireless location may also serve to maintain WLAN security. For example, by using location information one allows access to files or databases only for users from certain physical areas.

A detailed survey of the different wireless location methods and applications can be found in Drane *et al.* (1998). For more recent overview see Gustaffson and Gunnarsson (2005) and references therein.

3.2 Solution of the Wireless Location Problem

Let us translate the location problem into more formal language. As it was mentioned in the previous section, the location of a mobile device is based on the noisy measurements of a certain kind. These measurements include received signal strength, when the transmitted and received signal power are known to the system and thus the channel attenuation can be computed. Other types of measurements provide directional (angle of arrival), temporal (time of arrival, time difference of arrival) or spatial (digital map, position estimates from GPS) information about the MS position. The relationship between measurements and position is described using a stochastic model.

First assume that the MS is fixed. Denote the two-dimensional position (coordinates) of a MS at time n by $X_n = (X_{1,n}, X_{2,n})^{\mathrm{T}}$ and the measurements collected from this position by Y_n. The general model which relates the measurements with the position and noise can be written as follows,

$$Y_n = h(X_n, W_n), \tag{3.1}$$

where h is an arbitrary vector-function and W_n is a random variable with known

probability distribution. In general there are no specific assumptions on the form of the h function or on the distribution of the measurement noise. For example, the relationship between received signal strength from the BTS j with known position and the MS position at time n is usually described by the empirical Okumura-Hata model (Hata (1980))

$$Y_{n,j} = K - 10\alpha \log_{10} |X_n - X^j_{\mathrm{BTS}}| + W_n, \tag{3.2}$$

where $|\cdot|$ denotes Euclidian distance, $Y_{n,j}$ is the received power on the log-scale, K is a known constant determined by the environment and antenna configuration, α is the damping parameter and W_n are i.i.d. Gaussian random variables. In paper A we used an alternative non–linear version of this model,

$$Y_{n,j} = K' \cdot |X_n - X^j_{\mathrm{BTS}}|^{-\alpha} \cdot W_n \tag{3.3}$$

with the received power on the original scale and with W_n being i.i.d. random variables with exponential distribution. Another example of non-linear relationship is the bearings only tracking problem, where the measurement equation includes the tangent function.

The measurement model with additive noise,

$$Y_n = h(X_n) + W_n,$$

is widely applied in practice, often with further restriction on measurement noise to follow Gaussian distribution. This restriction is mainly motivated by the central limit theorem and by the convenient properties of the Gaussian distribution.

The positioning problem in case of additive model consists in computing the coordinates X_n that minimise a given norm of the difference between actual measurements and the measurement model. Setting this norm to the quadratic form (weighted with the inverse of covariance matrix Σ_n in case of correlated measurements) results in the weighted least squares estimates,

$$\hat{X}_n = \arg\min (Y_n - h(X_n))^{\mathrm{T}} \Sigma_n^{-1} (Y_n - h(X_n)). \tag{3.4}$$

Alternatively, given the probability density of the measurement noise, $\mathbb{p}_{\mathrm{w}}$, one can use maximum likelihood approach, minimising the $-\log \mathbb{p}_{\mathrm{w}}(Y_n - h(X_n))$.

The analytical solution of the minimisation problem is possible only in case of Gaussian measurement noise, which explains wide use of the additive Gaussian models in practice. For non-Gaussian case numerical approximations are

avaliable, for example, the gradient or Gauss-Newton algorithms (see Dennis and Schnabel (1983)).

Now we make a step further and consider the problem of positioning a moving target. We add another model which describes the time changes of coordinates and relies on physical relationship between position, speed and acceleration. In the simplest case such model is given as follows,

$$\begin{cases} X_{1,n+1} = X_{1,n} + \Delta t \cdot V_{1,n+1} \\ X_{2,n+1} = X_{2,n} + \Delta t \cdot V_{2,n+1}, \end{cases} \tag{3.5}$$

where $V_{1,n+1}$ and $V_{2,n+1}$ correspond to horisonal and vertical velocities, and Δt is the time difference between measurements n and $n+1$. Another option is to model the time changes in speeds as well, using corresponding accelerations, i.e.

$$\begin{cases} X_{k,n+1} = X_{k,n} + \Delta t \cdot V_{k,n} + \frac{(\Delta t)^2}{2} A_{k,n+1}, \\ V_{k,n+1} = V_{k,n} + \Delta t \cdot A_{k,n+1}. \end{cases} \tag{3.6}$$

for $k = 1, 2$. Motion model may use the speed as an input in case of self-positioning or incorporate altitude information in aircraft navigation and tracking.

In papers A and B of this thesis we used a polar approach in motion model, where the coordinates are related to the velocity v_n of the MS and the direction φ_n of movement. The velocity and the directions are modeled as linear Markov chains with random accelerations and turns,

$$\begin{cases} v_{n+1} = v_n + \Delta t \cdot a_{n+1}, \\ \varphi_{n+1} = \varphi_n + \psi_{n+1}, \end{cases} \tag{3.7}$$

and coordinates are then given by

$$\begin{cases} X_{1,n+1} = X_{1,n} + \Delta t \cdot v_{n+1} \cdot \cos\varphi_{n+1}, \\ X_{2,n+1} = X_{2,n} + \Delta t \cdot v_{n+1} \cdot \sin\varphi_{n+1}. \end{cases} \tag{3.8}$$

We show that this approach gives better results in terms of mean squared error, but at the same time is more computationally intensive.

Polar model was also used in paper B, where we applied particle filtering algorithms in multiple-input multiple-output (MIMO) settings.

Write the general form of the motion model as

$$X_{n+1} = f(X_n, V_{n+1}) \tag{3.9}$$

without any specific restrictions on the vector-function f and the process noise V_n. Suppose that at each time point we observe a measurement Y_n and at time point n are given a sequence of measurements $y_{0:n}$. Combining the motion model (3.9) with the measurement equation (3.1) we immediately recognise the state-space model formulation (1.4) or (1.2)–(1.3) with the kernel Q determined by the process noise and the density g determined by the observation (measurement) noise. Indeed, the movement of a mobile is a process hidden from the observer, and one has to make an inference about this process using the measurements and use this inference to estimate a position. Thus, the positioning problem corresponds to the problem of filtering in state-space model formulation and in non–linear non–Gaussian settings can be solved using SMC methods.

Since early 20th, particle filters have been widely applied in wireless positioning, location and tracking. For early examples see Jwa *et al.* (2000) or Gustafsson *et al.* (2002). A review of the particle filter algorithms for non–linear non–Gaussian tracking problems can be found in the tutorial by Arulampalam *et al.* (2002) and in the book by Ristic *et al.* (2004). For more recent developments we refer to the collection of papers in doctoral thesis by Schön (2006).

4 Overview of the Papers

In this section we give a brief outline of the papers included in the thesis.

Paper A is a report about a simulation study on particle filtering performance in mobile positioning using received signal strength measurements. We use two different approaches: polar and Cartesian – to model the mobile movement, combined with two different models for the received signal strength: model with constant propagation coefficient and model where the propagation coefficient depends on the distance between the mobile and the base station. We found, that the particle filters based on a power model with varying propagation coefficient show better performance compared to filters based on a power model with constant propagation coefficient. The difference between polar and Cartesian approaches for the mobile movement is not so clear: filters based on the polar model have smaller error, but larger resampling rate, i.e. degenerate too often.

In paper B we apply particle filters for positioning in multiple-input multiple-output (MIMO) systems, where both the transmitter and the receiver have more than one antenna element. Increasing the system capacity, MIMO technologies have received a lot of attention as one of the most promising approaches for high data-rate wireless systems. As in paper A, we use polar and Cartesian approaches for mobile movement, but now combined with the geometrical model for the MIMO propagation channel, proposed by Molisch (2004). A simulation study shows that all tested filters are able to provide position estimates that satisfy the FCC requirements both for network and mobile–based positioning.

In paper C we investigate an algorithm for particle filtering for multi-dimensional state–space models which are decomposable in the coordinates. We call the model *decomposable*, if there exists a natural decomposition of the state space into two disjoint sub–spaces and if there exists a similar decomposition of the measurement equations into two independent parts. Instead of sampling multi-dimensional particles we propose to sample a smaller set of particles in each dimension and then to combine the resulting estimates. We demonstrate using the simulations that this approach effectively reduces the computation time without a large precision loss.

It is known that the quality of SMC estimates depends on the number of particles involved in the approximation. Several authors have studied the adaptation of the number of particles along the estimation procedure. For example, Fox (2003) suggested to increase the sample size until the Kullback–Leibler divergence between the true and the estimated target distribution is below a given

threshold. In paper D we explore two different strategies to increase the size of an instrumental sample: correlated sampling and observation–driven sampling and compare them with the independent sampling approach. Simulation examples do not show any improvement of the suggested methods over the naive one, but some issues are pointed out for future research.

Finally, in paper E we continue work on the correlated sampling scheme for SMC methods. We establish convergence result for this method and employ the idea of using *antithetic* variates – a well known method to reduce the variance of the standard Monte Carlo estimates. We show, based on the theoretical developments, that under certain conditions the antithetic sampling approach reduces the asymptotic variance of SMC estimates and illustrate our findings on numerical examples within the state–space models framework.

References

Anderson, B. D. O., and Moore, J. B. (1979) *Optimal Filtering*. Prentice-Hall Information and System Sciences Series.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002) A Tutorial on Particle Filters for On–line Non–linear/Non–Gaussian Bayesian Tracking. *IEEE Trans. Signal Proc.*, **50:2**, pp. 174–188.

Baum, L. E., Petrie, T. P., Soules, G., and Weiss, N. (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov Chains. *Ann. Math. Statist.*, **41**, pp. 164–171.

Brockwell, P. J., and Davis, R. A. (2002) *Introduction to Time Series and Forecasting, Second Edition.* Springer-Verlag New-York Inc.

Bucy, R. S., and Senne, K. D. (1971) Digital synthesis of Nonlinear Filters. *Automatica*, **7**, pp. 287–298.

Cappé, O., Moulines, É., and Rydén, T. (2005) *Inference in Hidden Markov Models.* New York: Springer.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999) An improved particle filter for non-linear problems. *IEEE Proc., Radar Sonar Navigation*, **146**, pp. 2–7.

Casella, G., and Berger, R. L. (1990) *Statistical Inference.* 2nd edition, Brooks/Cole.

Chopin, N. (2004) Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Ann. Stat.*, **32**, pp. 2385–2411.

Cornebise, J., Moulines, É, and Olsson, J. (2008) Adaptive Methods for Sequential Importance Sampling with Application to State Space models. Technical report, Preprints in Mathematical Sciences 2008:4, Lund University.

Crisan, D., and Doucet, A. (2002) A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Proc.*, **50:3**, pp. 736–746.

Del Moral, P. (1996) Nonlinear Filtering: Interacting Particle Solution *Markov Processes. Related Fields.*, **2**, pp. 555-579.

Del Moral, P. (2004) *Feyman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. New York: Springer.

Del Moral, P. and Guionnet, A. (2001) On the stability of interacting processes with applications to filtering and genetic algorithms. *Ann. Inst. H. Poincaré, Probab. et Stat.*, **37:2**, pp. 155-194.

Del Moral, P. and Miclo, L. (2000) Branching and Interacting Particle Systems Approximations of Feyman-Kac Formulae with Applications to Non-Linear Filtering. Lecture Notes in Mathematics, **1729**. Springer-Verlag, Berlin.

Dennis, J. E., and Schnabel, B. (1983) *Numerical Methods for Unconstrained Optimization and Non-linear equations*. Prentice-Hall series in Computational Mathematics. Englewood Cliffs, NJ:Prentice-Hall.

Douc, R., and Moulines, É. (2005) Limit theorems for weighted samples with applications to sequential Monte Carlo methods. To appear in *Ann. Stat.*

Doucet, A., de Freitas, N., and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice*. New York: Springer.

Drane, C., Macnaughtan, M. and Scott, C.(1998) Positioning GSM Telephones. *IEEE. Comm. Mag.*, pp. 46–59.

Fox, D. (2003) Adapting the sample size in particle filters through KLD–sampling. *International Journal of Robotic Research*,**22**, pp. 985 –1004.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993) Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEEE Proc. Comm. Radar Signal Proc.*, **140**, pp. 107–113.

Gustafsson, F., and Gunnarsson, F. N.(2005) Mobile Positioning Using Wireless Network: possibilities and fundamental limitations based on avaliable wireless network measurements. *IEEE Signal Process. Mag.*, **22:4**, pp. 41–53.

Gustafsson, F., Gunnarsson, F. Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund. P.-J. (2002) Particle Filters for Positioning, Navigation, and Tracking. *IEEE Trans. Signal Proc.*, **50:2**, pp 425–437,

Handschin, J. (1970). Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, **6**, pp. 555–563.

Hata, M. (1980) Empirical formula for propagation loss in land mobile radio services. *IEEE Trans. Veh. Tech.*, **50:2**, pp. 317–325.

Jwa, H., Kim, S., Cho, X. and Chun, J. (2000) Position tracking of mobiles in a cellular radio network using the constrained bootstrap filter. *Proc. National Aerospace Electronic Conference*, Dayton, OH, USA.

Kalman, R.E. (1960) A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic. Eng.*, **82**, pp. 35–45

Kalman, R. E., and Bucy, R. (1961). New results in linear filtering and prediction theory. *Trans. ASME J. Basic. Eng.*, **83**, pp. 95–108.

Kitagawa, G. (1996) Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comp. Graph. Statist.*, **1**, pp. 1–25.

Kong, A., Liu, J., and Wong, W. H. (1994) Sequential imputations and Bayesian missing data problems. *J. Am. Statist. Assoc.*, **89**, pp. 278–288.

Liu, J., and Chen, R. (1998) Sequential Monte-Carlo methods for dynamic systems. *J. Am. Statist. Assoc.*, **93**, pp. 1032–1044.

MacDonald, I. L., and Zucchini, W. (1997) Hidden Markov and Other Models for Discrete–Valued Time Series *Monographs on Statistics and Applied Probability 70* Chapman & Hall.

Molisch, A. F. (2004) A Generic Model for MIMO Wireless Propagation Channels in Macro- and Microcells. *IEEE Trans. Sig. Proc.*, **52:1**, pp. 61–71.

Olsson, J., Douc, R., and Moulines, É. (2006) Improving the performance of the two-stage sampling particle filter: a statistical perspective. In *On bounds and Asymptotic of Sequential Monte Carlo Methods for Filtering, Smoothing, and Maximum Likelihood Estimation in State Space Models*, pp. 143-181. Doctoral Thesis, Lund University.

Olsson, J., and Rydén, T. (2006) The Bootstrap Particle Filtering Bias. In *On bounds and Asymptotics of Sequential Monte Carlo Methods for Filtering, Smoothing, and Maximum Likelihood Estimation in State Space Models*, pp. 31–53. Doctoral Thesis, Lund University.

Olsson, J., Moulines, E., and Douc, R. (2007) Improving the performance of the two-stage sampling particle filter: a statistical perspective. *Proc. IEEE/SP 14th Workshop on Statistical Signal Processing*, Madison, USA, pp. 284–288.

Pitt, M. K., and Shephard, N. (1999) Filtering via simulation: Auxiliary particle filters. *J. Am. Statist. Assoc.*, **87**, pp. 493–499.

Rabiner, L. R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, pp. 257–285.

Ristic, B., Arulampalam, S., and Gordon, N. (2004) *Beyond the Kalman Filter: particle filters for tracking applications*. Artech House, London, UK.

Rubin, D. B. (1987) A noninterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modest: the SIR algorithm (discussion of Tanner and Wong). In *J. Am. Statist. Assoc.*, **82**, pp. 534–546.

Sayed, A. H., Tarighat, A., and Khajehnouri, N. (2005) Network-Based Wireless Location: challenges faced in developing techniques for accurate wireless location information. *IEEE Signal Process. Mag.*, **22:4**, pp. 24–40.

Schön, T. B. (2006) *Estimation of Nonlinear Dynamic Systems - Theory and Applications*, Linköping Studies in Science and Technology, Dissertations. No. 998.

Sorenson, H. W., and Alspach, D. L. (1971). Recursive Bayesian estimation using Gaussian sums. *Automatica* **7**, pp. 465–479.

A

Paper A

Maximum Likelihood and Particle Filter-based Mobile Positioning from Signal Strength Measurements

Svetlana Bizjajeva, Tobias Rydén

Abstract

In this paper we implement particle filtering methods for mobile positioning using received signal strength measurements. We used two different approaches for modeling the mobile movement and two different models for the received signal strength, resulting in four different particle filters. The performance of the filters was investigated in a simulation study, and compared to the results of maximum-likelihood estimation. The superiority of particle filtering over maximum-likelihood estimation is clearly demonstrated. Particle filters, based on a power model with varying propagation coefficient showed better performance compared to filters based on a power model with constant propagation coefficient. The difference between polar and Cartesian approaches for the mobile movement was not so clear: filter based on the polar model had smaller error, but larger resampling rate, i.e. degenerated too often.

Key words: Mobile positioning, particle filtering, simulations

1 Introduction

In wireless communication networks the positioning of mobile units is a key technology for location-based services such as location-sensitive billing, efficient handoff, or emergency calls. In recent years several mobile location methods using various types of measurements have been explored. Widely used types of measurements include time of arrival (TOA), time difference of arrival (TDOA) and

angle of arrival (AOA). These can be accompanied by the received signal strength of signals transmitted with known power, and by map information. Exhaustive overviews of different methods can be found in Rappaport *et al.* (1996) or Drane *et al.* (1998).

Once measurements are avaliable, the problem to filter out the mobile position arises. One possible solution is to linearize the models and use a Gaussian approximation for the noise, so that the Kalman filter can be applied. More advanced techniques include extended Kalman filtering (Mark and Zaidi (2002)) or Bayesian bootstrap filtering (Jwa *et al.* (2000)) to overcome the non-linearity problem. Recently, sequential Monte Carlo techniques, also known as particle filters (Doucet *et al.* (2001)), have been applied for positioning problems, see for example Nordlund *et al.* (2002).

In this paper we will explore the performance of particle filtering techniques in the self-positioning problem, i.e. when the mobile itself makes appropriate signal measurements and uses these measurements to determine its own position. A solution to the positioning problem when the only avaliable information is the received signal strength allows developing mobile-based equipment for positioning which could work independently of the mobile network operator.

2 Models for the mobile movement and for the received signal strength

Suppose the mobile moves along an arbitrary trajectory within the coverage area of several Base Transceiver Stations (BTS's) with known positions. The location of the mobile at a certain time point is unknown, and is subject to determine. We assume that the only available information is the received power from all the BTS's that are able to communicate with the mobile at the given time point.

The position of the mobile at time point t can be expressed in the common coordinate system by abscise x_t and ordinate y_t. The polar motion model relates these quantities to the velocity v_t of the mobile and the direction φ_t of movement. The velocity is generally unknown, but is assumed to follow a simple linear Markovian model with random acceleration a_t. The direction is defined by an angle with axis Ox, and is assumed to follow a linear model with random turns ψ_t. Thus the changes in speed and direction of the movement during time interval

Δt can be described by the following system of equations:

$$\begin{cases} v_t = v_{t-1} + \Delta t \cdot a_t, \\ \varphi_t = \varphi_{t-1} + \psi_t. \end{cases} \tag{2.1}$$

In this model the acceleration, a_t, and the size of turn, ψ_t, are assumed to be sequences of independent random variables with $N(0, \sigma_a^2)$ and $U[-\pi, \pi]$ distributions, respectively.

The mobile's coordinates at time point t are completely determined by the coordinates at the previous time point and by the velocity and direction in (2.1). Indeed,

$$\begin{cases} x_t = f_p(v_t, \varphi_t) = x_{t-1} + \Delta t \cdot v_t \cdot \cos\varphi_t, \\ y_t = g_p(v_t, \varphi_t) = y_{t-1} + \Delta t \cdot v_t \cdot \sin\varphi_t. \end{cases} \tag{2.2}$$

Alternatively, the motion of the mobile might be described using horizontal and vertical velocities $\dot{x}_t$ and $\dot{y}_t$. Both are generally unknown and assumed to follow a simple linear Markovian model with corresponding random accelerations. The direction and speed of the movement are then determined by the size and sign of the horizontal and vertical velocities.

Thus in the Cartesian model the changes in the velocities during the time interval Δt are described by the following equations:

$$\begin{cases} \dot{x}_t = \dot{x}_{t-1} + \Delta t \cdot a_{x,t}, \\ \dot{y}_t = \dot{y}_{t-1} + \Delta t \cdot a_{y,t}. \end{cases} \tag{2.3}$$

In this model the accelerations $a_{x,t}$ and $a_{y,t}$ are sequences of independent random variables with $N(0, \sigma_x^2)$ and $N(0, \sigma_y^2)$ distributions, respectively. The mobile's coordinates at time t are determined by the previous coordinates and by the velocities given in (2.3):

$$\begin{cases} x_t = f_c(\dot{x}_t, \dot{y}_t) = x_{t-1} + \Delta t \cdot \dot{x}_t, \\ y_t = g_c(\dot{x}_t, \dot{y}_t) = y_{t-1} + \Delta t \cdot \dot{y}_t. \end{cases} \tag{2.4}$$

The power received at time t from the j-th BTS is modeled as

$$P_{t,j} = P_{0,j} \cdot d_{t,j}^{-\alpha} \cdot \varepsilon_{t,j}, \tag{2.5}$$

where $d_{t,j}$ is the Euclidian distance between the mobile and the BTS, and the noise $\varepsilon_{t,j}$ are independent random variables from an exponential distribution with mean λ. In the simplest case the propagation coefficient α is fixed to some value between 2 and 4, while in a more complicated case

$$\alpha_t = \begin{cases} 2, & \text{if } d_{t,j} \leq 50\,\text{m}; \\ \alpha_0 + \alpha_1 d_{t,j}, & \text{if } 50 < d_{t,j} < 1000\,\text{m}; \\ 4, & \text{if } d_{t,j} \geq 1000\,\text{m}, \end{cases} \tag{2.6}$$

where α_0 and α_1 are set as to make α_t continuous in $d_{t,j}$. If there are M BTS's, the received powers are described by the following system of equations:

$$\begin{cases} P_{t,1} &= P_{0,1} \cdot d_{t,1}^{-\alpha} \cdot \varepsilon_{t,1}, \\ P_{t,2} &= P_{0,2} \cdot d_{t,2}^{-\alpha} \cdot \varepsilon_{t,2}, \\ &\vdots \\ P_{t,M} &= P_{0,M} \cdot d_{t,M}^{-\alpha} \cdot \varepsilon_{t,M}. \end{cases} \tag{2.7}$$

We assume that the transmitted power is the same for all BTS's,

$$P_{0,1} = P_{0,2} = \cdots = P_{0,M}. \tag{2.8}$$

Also note that the noise variables $\varepsilon_{i,j}$ for different time points and for different BTS's are independent and identically distributed.

3 Maximum Likelihood Estimation

This section presents the ML solution to the positioning problem. Consider first the power model with constant propagation coefficient. The conditional distribution of the power received from j-th BTS at time t, given the mobile's coordinates, is exponential,

$$P_{t,j} \sim \text{Exp}\left(\lambda P_0 d_{t,j}^{-\alpha}\right), \tag{3.1}$$

with density

$$p(u_{t,j}) = \frac{d_{t,j}^{\alpha}}{P_0 \lambda} \exp\left(-\frac{u_{t,j} d_{t,j}^{\alpha}}{P_0 \lambda}\right), \tag{3.2}$$

where

$$d_{t,j} = \sqrt{(x_t - x_{BS_j})^2 + (y_t - y_{BS_j})^2} \tag{3.3}$$

and x_{BS_j} and y_{BS_j} are abscise and ordinate of the j-th BTS. The log-density is then

$$\log p(u_{t,j}) = \alpha \log d_{t,j} - \log(P_0\lambda) - \frac{u_{t,j} d_{t,j}^{\alpha}}{P_0\lambda}. \tag{3.4}$$

Suppose there are n observations taken at the same point (or, at the same distance from all BTS's)

$$\mathbf{u}_{t,j} = (u_{t,j,1}, \ldots, u_{t,j,n})'. \tag{3.5}$$

Then the log-likelihood function is given by

$$\begin{aligned} l(\mathbf{u}_{t,j}\,;\, x_t, y_t) &= \sum_{k=1}^{n} \left(\alpha \log d_{t,j} - \log(P_0\lambda) - \frac{u_{t,j,k} d_{t,j}^{\alpha}}{P_0\lambda} \right) \\ &= \alpha n \log d_{t,j} - n \log(P_0\lambda) - \frac{d_{t,j}^{\alpha}}{P_0\lambda} \sum_{k=1}^{n} u_{t,j,k}. \end{aligned} \tag{3.6}$$

Since the power measurements for the different BTS's are independent, the log-likelihood for the data $\mathbf{u}_t$ from all BTS's at time t is expressed by the sum over j.

The score equations are then

$$\begin{cases} \frac{\partial}{\partial x_t} l(\mathbf{u}_t\,;\, x_t, y_t) = \sum_{j=1}^{M} \frac{\alpha(x_t - x_{BS_j}) \left(P_0\lambda n - d_{t,j}^{\alpha} \sum_{k=1}^{n} u_{t,j,k} \right)}{P_0\lambda d_{t,j}^2} = 0, \\ \frac{\partial}{\partial y_t} l(\mathbf{u}_t\,;\, x_t, y_t) = \sum_{j=1}^{M} \frac{\alpha(y_t - y_{BS_j}) \left(P_0\lambda n - d_{t,j}^{\alpha} \sum_{k=1}^{n} u_{t,j,k} \right)}{P_0\lambda d_{t,j}^2} = 0. \end{cases} \tag{3.7}$$

If the propagation coefficient depends linearly on the distance,

$$\alpha_{t,j} = \alpha_0 + \alpha_1 d_{t,j}, \tag{3.8}$$

the score equations take form

$$\begin{cases} \frac{\partial}{\partial x_t} l(\mathbf{u}_t\,;\, x_t, y_t) = \sum_{j=1}^{M} \frac{\tilde{\alpha}_{t,j}(x_t - x_{BS_j})\left(P_0 \lambda n - d_{t,j}^{\alpha_0 + \alpha_1 d_{t,j}} \sum_{k=1}^{n} u_{t,j,k}\right)}{P_0 \lambda d_{t,j}^2} = 0, \\ \\ \frac{\partial}{\partial y_t} l(\mathbf{u}\,;\, x_t, y_t) = \sum_{j=1}^{M} \frac{\tilde{\alpha}_{t,j}(y_t - y_{BS_j})\left(P_0 \lambda n - d_{t,j}^{\alpha_0 + \alpha_1 d_{t,j}} \sum_{k=1}^{n} u_{t,j,k}\right)}{P_0 \lambda d_{t,j}^2} = 0, \end{cases} \tag{3.9}$$

where

$$\tilde{\alpha}_{t,j} = \alpha_1 d_{t,j} \log d_{t,j} + \alpha_0 + \alpha_1 d_{t,j}. \tag{3.10}$$

The ML estimates of the position are obtained by solving the score equation system w.r.t. x_t and y_t.

Using the Fisher information matrix, it is possible to construct approximate confidence ellipses for the ML estimate $(\hat{x}_t, \hat{y}_t)$. The approximate 95% confidence ellipse is given by the equation

$$(x - \hat{x}_t,\, y - \hat{y}_t) B(\hat{x}_t, \hat{y}_t) (x - \hat{x}_t,\, y - \hat{y}_t)' = \chi^2_{0.95}(2), \tag{3.11}$$

where $B(\hat{x}_t, \hat{y}_t)$ is the Fisher information matrix

$$B(x_t, y_t) = \begin{pmatrix} -\mathsf{E}\left[\frac{\partial^2}{\partial x_t^2} l(\mathbf{u}_t\,;\, x_t, y_t)\right] & -\mathsf{E}\left[\frac{\partial^2}{\partial x_t \partial y_t} l(\mathbf{u}_t\,;\, x_t, y_t)\right] \\ \\ -\mathsf{E}\left[\frac{\partial^2}{\partial y_t \partial x_t} l(\mathbf{u}_t\,;\, x_t, y_t)\right] & -\mathsf{E}\left[\frac{\partial^2}{\partial y_t^2} l(\mathbf{u}_t\,;\, x_t, y_t)\right] \end{pmatrix}, \tag{3.12}$$

evaluated at $(\hat{x}_t, \hat{y}_t)$, and $\chi^2_{0.95}(2)$ is 95% quantile of the chi-square distribution with 2 degrees of freedom.

For constant propagation coefficient α the Fisher information matrix is

$$B(x_t, y_t) = \alpha^2 n \begin{pmatrix} \sum_{j=1}^{M} \frac{(x_t - x_{BS_j})^2}{d_{t,j}^4} & \sum_{j=1}^{M} \frac{(x_t - x_{BS_j})(y_t - y_{BS_j})}{d_{t,j}^4} \\ \\ \sum_{j=1}^{M} \frac{(x_t - x_{BS_j})(y_t - y_{BS_j})}{d_{t,j}^4} & \sum_{j=1}^{M} \frac{(y_t - y_{BS_j})^2}{d_{t,j}^4} \end{pmatrix}, \tag{3.13}$$

while if the propagation coefficient depends on the distance as in (3.8), it is

$$B(x_t, y_t) = n \begin{pmatrix} \sum_{j=1}^{M} \frac{(x_t - x_{BS_j})^2 \tilde{\alpha}_{t,j}^2}{d_{t,j}^4} & \sum_{j=1}^{M} \frac{(x_t - x_{BS_j})(y_t - y_{BS_j}) \tilde{\alpha}_{t,j}^2}{d_{t,j}^4} \\ \sum_{j=1}^{M} \frac{(x_t - x_{BS_j})(y_t - y_{BS_j}) \tilde{\alpha}_{t,j}^2}{d_{t,j}^4} & \sum_{j=1}^{M} \frac{(y_t - y_{BS_j})^2 \tilde{\alpha}_{t,j}^2}{d_{t,j}^4} \end{pmatrix}. \quad (3.14)$$

4 Particle filtering

Equations (2.1)–(2.2) or (2.3)–(2.4) and (2.7) define a state-space model. Suppose that the values of the power received from M BTS's up to time t are known:

$$\mathbf{P}_{1:t} = (\mathbf{P}_1, \ldots, \mathbf{P}_t) = \begin{pmatrix} P_{1,1} & \ldots & P_{1,t} \\ \vdots & \ddots & \vdots \\ P_{M,1} & \ldots & P_{M,t} \end{pmatrix}. \quad (4.1)$$

We are interested in the coordinates of the mobile at this time point, $\mathbf{r}_t = (x_t, y_t)'$, which are defined using either the polar (2.2) or the Cartesian (2.4) models.

Write in general

$$\mathbf{r}_t = h(\mathbf{z}_t), \quad (4.2)$$

where $\mathbf{z}_t$ denotes the state vector at time t:

$$\mathbf{z}_t = \begin{cases} (v_t,\ \varphi_t, x_t, y_t)' & \text{for the polar model (2.1)–(2.2),} \\ (\dot{x}_t,\ \dot{y}_t, x_t, y_t)' & \text{for the Cartesian model (2.3)–(2.4).} \end{cases} \quad (4.3)$$

The best guess, in the sense of mean squared loss, about the value of $h(\mathbf{z}_t)$ at time t is the expected value of this function w.r.t. the joint conditional distribution of states,

$$I = \mathsf{E}_{p(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})}\left[h(\mathbf{z}_t)\right] = \int h(\mathbf{z}_t) p(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})\, d(\mathbf{z}_{0:t}). \quad (4.4)$$

Integrals like these can be approximated using Monte Carlo methods. If it is possible to simulate from $p(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})$, then usual Monte Carlo estimates can be calculated. This is not the case in our situation, so the *importance sampling* method was applied.

The key idea of importance sampling is to introduce an importance distribution, $\pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})$, which is easy to simulate from. Then the integrals can be written as weighted expectations w.r.t. to the importance distribution,

$$\begin{aligned} I &= \int h(\mathbf{z}_t) p(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})\, d(\mathbf{z}_{0:t}) \\ &= \int h(\mathbf{z}_t) \frac{p(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})}{\pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})} \pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})\, d(\mathbf{z}_{0:t}) \\ &= \mathsf{E}_{\pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})}[f(\mathbf{z}_t)\omega^*(\mathbf{z}_{0:t})], \end{aligned} \tag{4.5}$$

where

$$\omega^*(\mathbf{z}_{0:t}) = \frac{p(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})}{\pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})}. \tag{4.6}$$

Suppose there are N i.i.d. samples from the importance distribution. Then an estimate of the integral is the weighted sample mean

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} h(\mathbf{z}_t^{(i)}) \omega_t^{*(i)}, \tag{4.7}$$

with importance weights

$$\omega_t^{*(i)} = \omega^*(\mathbf{z}_{0:t}^{(i)}) = \frac{p(\mathbf{z}_{0:t}^{(i)}|\mathbf{P}_{1:t})}{\pi(\mathbf{z}_{0:t}^{(i)}|\mathbf{P}_{1:t})} = \frac{p(\mathbf{P}_{1:t}|\mathbf{z}_{0:t}^{(i)}) p(\mathbf{z}_{0:t}^{(i)})}{p(\mathbf{P}_{1:t})\, \pi(\mathbf{z}_{0:t}^{(i)}|\mathbf{P}_{1:t})}. \tag{4.8}$$

Since the importance weights contain the normalizing constant $p(\mathbf{P}_{1:t})$, which is typically unknown, the estimate (4.7) can in general not be computed.

However, the integral can be expressed as the ratio of two expectations,

$$I = \frac{\mathsf{E}_{\pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})}[h(\mathbf{z}_t)\omega(\mathbf{z}_{0:t})]}{\mathsf{E}_{\pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})}[\omega(\mathbf{z}_{0:t})]}, \tag{4.9}$$

with weights

$$\omega(\mathbf{z}_{0:t}) = \frac{p(\mathbf{P}_{1:t}|\mathbf{z}_{0:t}) p(\mathbf{z}_{0:t})}{\pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t})}. \tag{4.10}$$

Hence an estimate of the integral is given by

$$\hat{I}_N = \frac{\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{z}_{0:t}^{(i)}) \omega_t^{(i)}}{\frac{1}{N}\sum_{j=1}^{N} \omega_t^{(j)}} = \sum_{i=1}^{N} f(\mathbf{z}_t^{(i)}) \tilde{\omega}_t^{(i)}, \tag{4.11}$$

where the importance weights are

$$\omega_t^{(i)} = \omega(\mathbf{z}_{0:t}^{(i)}) = \frac{p(\mathbf{P}_{1:t}|\mathbf{z}_{0:t}^{(i)})p(\mathbf{z}_{0:t}^{(i)})}{\pi(\mathbf{z}_{0:t}^{(i)}|\mathbf{P}_{1:t})}, \tag{4.12}$$

and $\tilde{\omega}_t^{(i)}$ are normalized importance weights,

$$\tilde{\omega}_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{j=1}^{N} \omega_t^{(j)}}. \tag{4.13}$$

Restricting the importance distribution to the form

$$\pi(\mathbf{z}_{0:t}|\mathbf{P}_{1:t}) = \pi(\mathbf{z}_{0:t-1}|\mathbf{P}_{1:t-1})\pi(\mathbf{z}_t|\mathbf{z}_{0:t-1},\, \mathbf{P}_{1:t-1}), \tag{4.14}$$

allows for evaluating the importance weights sequentially in time. If the prior distribution of the states (system dynamics) is taken as the importance distribution, then the updating coefficients are equal to the values of the conditional density of the observations, evaluated at the observed values,

$$\omega_t^{(i)} = \omega_{t-1}^{(i)} p(\mathbf{P}_{1:t}|\mathbf{z}_{0:t}^{(i)}) = \omega_{t-1}^{(i)} \prod_{j=1}^{M} p(P_{t,j}), \tag{4.15}$$

where p, in our setting, corresponds to the exponential density with mean $\lambda P_0 d_{t,j}^{-\alpha_{t,j}}$.

To avoid degeneracy of the importance sampling method, the resampling procedure according to the normalized weights was included, namely, the systematic resampling scheme (Carpenter *et al.* (1999)). The effective sample size estimate (Kong *et al.* (1994))

$$N_e = \left[\sum_{i=1}^{N} [\tilde{\omega}_t^{(i)}]^2 \right]^{-1} \tag{4.16}$$

was used as the measure of degeneracy, and resampling was carried out if N_e fell below a fixed threshold. In our algorithm the threshold was set to 60% of the total number of particles.

The filtering algorithm is specified as follows:

1. *Initialization*, $t = 0$
   • For $i = 1$ to N, sample

   $$(x_0^{(i)}, y_0^{(i)}) \sim \text{Uniform[coverage area]},$$
   $$v_0^{(i)} \sim \text{Uniform}[20,\ 70],\ \varphi_0^{(i)} \sim \text{Uniform}[0, 2\pi]$$
   or
   $$\dot{x}_0^{(i)} \sim \text{Uniform}[20,\ 70],\ \dot{y}_0^{(i)} \sim \text{Uniform}[20,\ 70]$$

   • Set $t = 1$

2. *Propagation and Resampling*
   • For $i = 1$ to N, move current samples (or particles) according to (2.1)–(2.2) or (2.3)–(2.4) to obtain $\mathbf{z}_t^{(i)}$.
   • For $i = 1$ to N and for $j = 1$ to M, evaluate

   $$d_t^{(ij)} = \sqrt{(x_t^{(i)} - x_{BS_j})^2 + (y_t^{(i)} - y_{BS_j})^2}.$$

   • For $i = 1$ to N, update the importance weights according to (4.15), normalize to obtain normalized weights $\tilde{\omega}_t^{(i)}$ and calculate the effective sample size.
   • Take as the position estimate

   $$\hat{x}_t = \sum_{i=1}^{N} \tilde{\omega}_t^{(i)} x_t^{(i)}, \quad \hat{y}_t = \sum_{i=1}^{N} \tilde{\omega}_t^{(i)} y_t^{(i)}.$$

   - If $N_e < 0.6N$, resample with replacement N particles

   $$\{\mathbf{z}_t^{(i)};\ i = 1, \ldots, N\}$$

   according to the normalized importance weights. Set all weights equal to $1/N$.

   • Set $t = t + 1$ and go to step 2.

5 Simulations

In the first simulation the mobile moves within the coverage area of three BTS's during 3 minutes with constant speed and constant direction:

$$v_0 = 60\,\text{km/h},\ v_0 = v_1 = \cdots = v_T;$$
$$\varphi_0 = \pi/6,\ \varphi_0 = \varphi_1 = \cdots = \varphi_T.$$

The sampling rate is 3 s^{-1}, so $\Delta t = 1/3$ s and $T = 540$. The BTS's are located at the points (0, 500), (–500, 500) and (500, –500), and the initial position was chosen randomly within the square $[-1000, 1000] \times [-1000, 1000]$.

The transmitted power was set to 16 dB and the power noise was simulated from an exponential distribution with mean 2 dB.

The received power from all BTS's was calculated by (2.7) first with the fixed propagation coefficient $\alpha = 2$ and then with the propagation coefficient depending on the distance as in (2.6).

Figure 1 shows the simulated track and 95% confidence ellipses for three selected points along the track. The ellipses were computed using the Fisher information matrix (3.13) or (3.14) around the corresponding ML estimates. These ellipses thus display the uncertainty of a single measurement taken at this point, disregarding any dynamics of the movement. As can be expected, the ML estimates for the model with varying propagation coefficient are more precise about the true values, than the ML estimates for the model with constant propagation coefficient. The estimation becomes more uncertain when the mobile moves outside the area between the BTS's.

For the on-line estimation of the true trajectory, four different filters were applied.

- **Filter 1:** Polar model (2.1)–(2.2) for states and fixed propagation coefficient $\alpha = 2$ in the power model (2.7);

- **Filter 2:** Polar model (2.1)–(2.2) for states and propagation coefficient determined by (2.6) in the power model (2.7);

- **Filter 3:** Cartesian model (2.3)–(2.4) for states and fixed propagation coefficient $\alpha = 2$ in the power model (2.7);

- **Filter 4:** Cartesian model (2.3)–(2.4) for states and propagation coefficient determined by (2.6) in the power model (2.7).

All four filters were run using 1000 particles and the appropriate received power input. In filters 1 and 2 the acceleration $a_t \sim N(0, 1)$, the random angle $\varphi_t \sim U[-\pi, \pi]$ and the power noise $\varepsilon_{i,j} \sim \text{Exp}(2\,\text{dB})$. In filters 3 and 4 the random accelerations $a_{x,t} \sim N(0, 1)$, $a_{y,t} \sim N(0, 1)$ and the power noise $\varepsilon_{i,j} \sim \text{Exp}(2\,\text{dB})$.

The RMSE based on 100 runs ($R = 100$) was calculated in order to estimate the over-time performance for each filter,

$$RMSE_t = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left[(x_t - \hat{x}_{t,r})^2 + (y_t - \hat{y}_{t,r})^2\right]}, \quad t = 1, \ldots, T. \qquad (5.1)$$

Filtering results for the one run and the RMSE are displayed in Figure 2.

The plot of RMSE shows better performance for filters 2 and 4, which are based on the model with varying propagation coefficient. Among these two, filter 4, based on the Cartesian model for states, has the smallest RMSE for almost all time points. It takes around 5 s for all filters to find the true trajectory, and during this time the RMSE drops to around one third of its initial value. After 90 s the first filter loses the track and diverges drastically, with the over-time mean of RMSE of 1 km. The other three filters follow the true path, and start to diverge after around 150 s when the mobile moves outside the area between the BTS's. After discarding the first 15 (burn-in) and last 90 measurements, filter 4 gives a mean error of 34.3 m, which is several times smaller that the mean error for filters 2 (71.9 m) and 3 (190.4 m).

The resampling rates for four filters are 13.7%, 29.4%, 6.7% and 11.8%, respectively. Filter 2 allows particles to explore the state space more actively, as its resampling rate is about 2–3 times higher than for the other filters.

Figure 3 displays the filtering results with different standard deviations for the acceleration in models (2.1) and (2.3). Filters based on the polar model for the states are able to track the true trajectory even with a relatively small standard deviation (0.5 m/s^2), whereas Filters 1 and 2 diverge in this case. A quite large standard deviation (5 m/s^2) allows all filters to find the true position, but the filters based on the model with constant propagation coefficient are less precise. Filters 2 and 4 show more similar results as the standard deviation increases.

Figure 4 displays changes in the resampling rate depending on the standard deviation in the models. The resampling rate shows how often the filter starts to degenerate. This rate also depends on an arbitrarily chosen threshold of 60% in the condition for resampling. At this point we consider a resampling rate between 20% and 25% as optimal. For filter 1 the resampling rate seems to bee too small. In this case the filter resamples in around 15% of steps, so most of the time it uses quite a lot of different particles. But if all these particles are far away from the truth, then the filter might be forced to follow the wrong path. On the

contrary, filter 3 resamples too often, in around 30% of steps. The resampling rate for filter 2 changes rapidly as the standard deviation increases to 2 m/s^2. It might be an indication that the filter is trying to find an optimal rate for the given standard deviation and resampling threshold. As the standard deviation increases, the resampling rates for filters 2 and 4 become more close to each other.

Results of the filtering with $\sigma_a = 3$, $\sigma_x = \sigma_y = 3$ and of the ML estimation are compared in Figure 5. The superiority of particle filtering is clear for filters 2 and 4 (model with varying propagation coefficient). Particles are highly concentrated around the true value, and 95% confidence ellipses are more narrow than the ML confidence ellipses.

In the second simulation the mobile moves in the area between same BTS's during 3 minutes with constant speed of 60 km/h along a track with $\pi/2$ and $\pi/4$ turns. All other parameters are the same as in the first simulation. Again, four filters were applied, with exactly the same setup.

Figure 6 shows the true trajectory and the filtering results for four different filters and the over-time RMSE for each filter. The superiority of filter based on the polar model for states, with varying propagation coefficient in the power model, is clear from both plots. After discarding the first 15 and last 90 measurements, the mean RMSE for this filter is 40.7 m, comparing to the mean errors of 147.9 m, 108.2 m and 50.1 m for filters 1, 3 and 4, respectively. The price to pay for the best performance is the highest resampling rate, 31.1%, compared to 12.7%, 10.5% and 20.7% for other three filters.

6 Discussion

Two state models and two power models, resulting in four different particle filters, were compared in the simulation study. Filters based on the power model with varying propagation coefficient show better performance compared to the filters based on the power model with constant propagation coefficient. The difference between polar and Cartesian models for states is not so clear. Filters based on the polar model estimate the position with smaller error. At the same time they might resample too often (up to 65%), if the standard deviation of the acceleration in the state model is small. The resampling rates for the filters based on the Cartesian model is more stable and are close to the optimal when the standard deviation is more than 2 m/s^2.

For comparison, consider the filter applied for the positioning in Gustafsson

et al. (2002). In this paper the motion model uses the velocity as an avaliable input signal. The received power from two BTS's is modeled on log-scale with the constant propagation coefficient $\alpha \in [2, 5]$, and with additive noise from $N(0, \sigma_e^2)$, where $\sigma_e = 6\,\mathrm{dB}$. The implemented particle filter with 66.6% resampling threshold yields an RMSE of 36 m. This error is similar to the results of filtering with varying propagation coefficient in the power model. Hence, inclusion of more information about the distance into the power model allows us to apply more general model for the states without significant loss in the precision.

It seems reasonable to consider the filter with smallest RMSE as the best one, but the resampling rate has also be taken into account. The best filter should therefore combine the small estimation error with reasonable resampling frequency. The choice of the standard deviations in the state model and of the resampling threshold in order to have an optimal resampling rate requires further investigation.

References

Carpenter, J., Clifford, P., and Fearnhead, P. (1999) Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proc.*, **146**.

Doucet, A. (1998) On sequential simulation-based methods for Bayesian filtering, Technical Report CUED-F-ENG-TR310, University of Cambridge, Dept. of Engineering.

Doucet, A., Godsill, S. J., and Andrieu, C. (2000) On Sequential Monte-Carlo Sampling Methods for Bayesian Filtering. *Stat. Comput.*, **10:3**, pp. 197–208.

Doucet, A., de Freitas, N., and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice.* New York: Springer.

Drane, C., Macnaughtan, M., and Scott, C.(1998) Positioning GSM Telephones. *IEEE Communications Magazine,*, pp. 46–59.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund. P.-J. (2002) Particle Filters for Positioning, Navigation, and Tracking. *IEEE Trans. Signal Processing*, **50:2**, pp. 425–437,

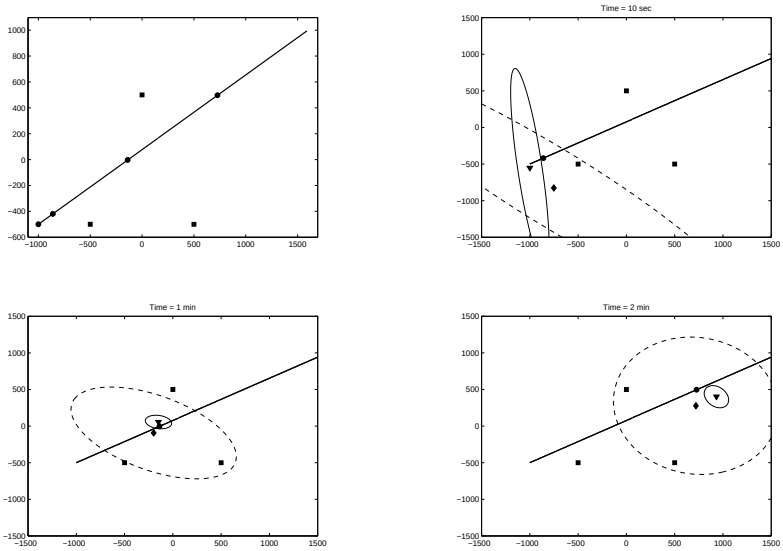


Figure 1: The simulated track (top left) and ML-estimates with 95% confidence ellipses for the selected points (other plots). Diamonds and dashed lines correspond to the ML estimates and ellipses for the power model with fixed propagation coefficient; triangles and solid lines correspond to the ML estimates and ellipses for the power model with varying propagation coefficient. Circles on track indicate the true mobile position, squares show positions of BTS's

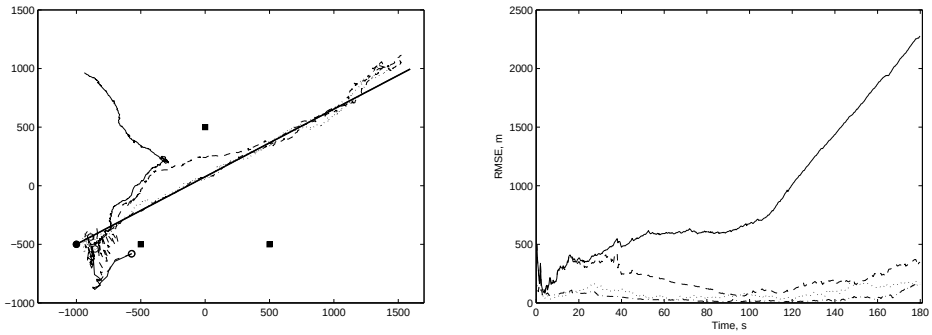


Figure 2: Left: Simulated trajectory (thick solid line) and filtered ones for the four different filters. Circle indicates the initial position of mobile, squares show positions of BTS's. Right: RMSE based on 100 runs. Thin solid lines in both plots correspond to filtered track and RMSE for Filter 1; dotted lines - Filter 2; dashed lines - Filter 3; dash-dotted lines - Filter 4.

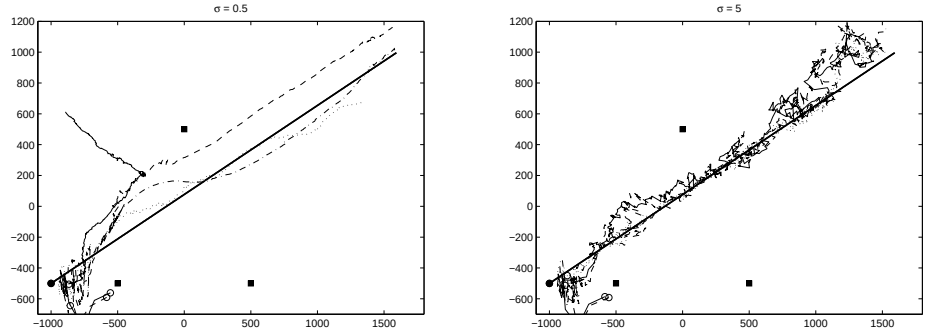


Figure 3: Filtering results for different standard deviations. Line code: thin solid - Filter 1; dotted - Filter 2; dashed - Filter 3; dash-dotted - Filter 4.

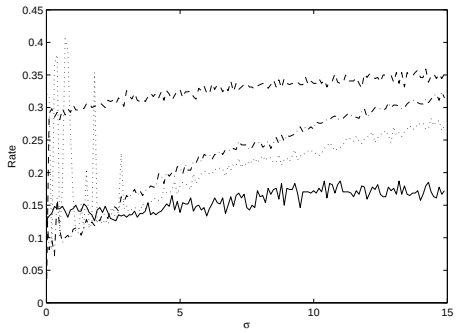


Figure 4: Resampling rates depending on the standard deviation. Line code: solid - Filter 1; dotted - Filter 2; dashed - Filter 3; dash-dotted - Filter 4.

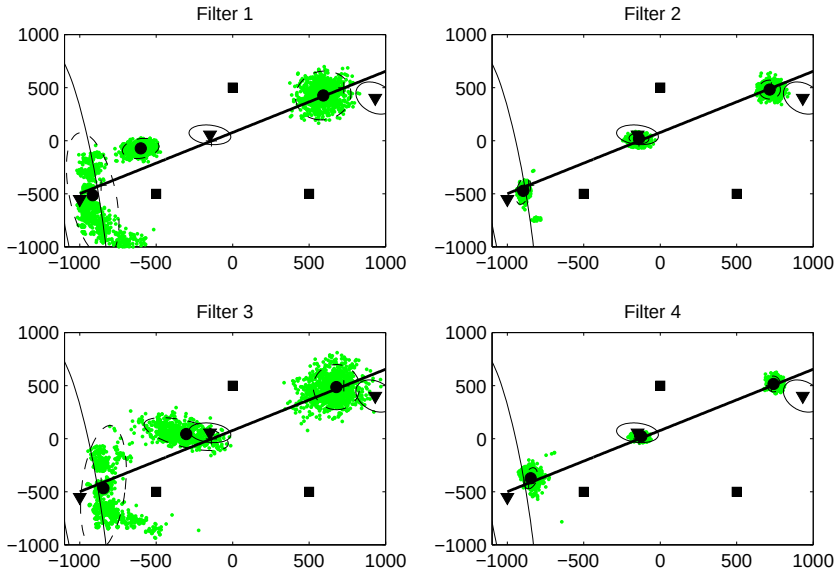


Figure 5: Particle clouds and confidence ellipses for three selected points along the track. Circles and dashed lines correspond to filtered positions and 95% confidence ellipses based on the covariance estimate; Triangles and solid lines correspond to ML estimates and confidence ellipses for the power model with varying propagation coefficient. Black crosses denote the true position.

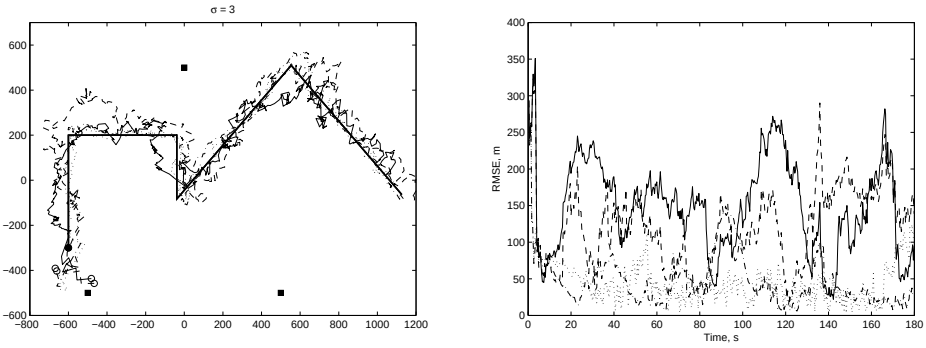


Figure 6: Left: Simulated trajectory with turns and the filtered one for the four different filters. Right: RMSE based on 100 runs. Line code: thin solid - Filter 1; dotted - Filter 2; dashed - Filter 3; dash-dotted - Filter 4.

Jwa, H., Kim, S., Cho, X., and Chun, J. (2000) Position tracking of mobiles in a cellular radio network using the constrained bootstrap filter. *Proc. National Aerospace Electronic Conference*, Dayton, OH, USA.

Kitagawa, G. (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space model, *Journal of Computational and Graphical Statistics*, **5(1)**, pp. 1–25.

Kong, A., Liu, J. S., and Wong, W. H. (1994) Sequential imputations and Bayesian missing data problems. *J. Am. Stat. Assoc.*, **89**, pp. 278–288.

Mark, B., and Zaidi, Z. (2002) Robust mobility Tracking for Cellular Networks. *Proc IEEE International Communication Conference*, New-York, USA.

Nordlund, P.-J., Gunnarsson, F., and Gustafsson, F. (2002) Particle filters for positioning in wireless networks, *Proc. of EUSIPCO*, Toulouse, France.

Rappaport, T. S., Reed, J. H., and Woerner B. D. (1996) Position location using wireless communications on highways of the future. *IEEE Communications Magazine*, **34:10**, pp. 33–41.

B

Paper B

Mobile Positioning in MIMO System Using Particle Filtering

Svetlana Bizjajeva, Tobias Rydén, Ove Edfors

Abstract

This paper represents the results of a simulation study on positioning of a mobile unit in MIMO settings. We used two different approaches for modeling the mobile movement, combined with a simple geometrical model for the MIMO channel. Three different particle filters were implemented for the position estimation. The results show that all three filters are able to achieve estimation accuracy required by Federal Communication Commission. The dimensionality of the particle filter state space is independent of the number of antenna elements, and it is possible to increase the number of antennas and use more sophisticated channel models without changing the filtering algorithms.

Key words: MIMO, mobile positioning, channel modeling, particle filtering, simulations

1 Introduction

Wireless systems are now used worldwide to help people and machines to communicate with each other irrespectively of their location. In a global perspective, wireless stands to be a method most people will use to connect to the Internet. New generation wireless communication systems (4G) should be able to provide clients with all the benefits associated with the World Wide Web: multimedia, e-commerce, unified messages, peer-to-peer network etc. To increase system performance is thus very important.

Another goal of 4G systems is to allow switching between networks of systems that gives "the best" connection at the moment. Mixing various connections from

satellites to local area networks may result in a crowded frequency spectrum and requires a signaling strategy that is spectrally efficient.

Using multiple antennas at both transmitter and receiver can solve these issues. The MIMO (multiple-input multiple-output) technology, proposed by Paulraj and Kailath in Paulraj and Kailath (1994), increases the spectral efficiency of a system. It enables high capacities suited for Internet and multimedia services and also dramatically increases range and reliability. In the last few years, MIMO systems have emerged as one of the most promising approaches for high data-rate wireless systems. For more details about the MIMO technology see, for example, Paulraj *et al.* (2003).

The positioning of a mobile unit in MIMO settings is a challenging problem. During the last decade various location technologies have been invented using either cellular network-based, mobile-based, or hybrid approaches. A comprehensive overview of different positioning methods can be found in Syrjärinne (2001). Most known and widely used is the satellite Global Positioning System (GPS), which is based on measurements of time difference of arrival. The propagation time of signals is measured simultaneously from satellites at known locations and the distance between a satellite and a user receiver is obtained by multiplying the propagation time with the speed of light, assuming the line of sight (LOS). In most applications however, the LOS signal is succeeded by multi-path components that arrive to the receiver with a short delay. This introduces significant errors in the LOS path time of arrival and gain estimation, especially in urban environments with many reflections from buildings and other objects. On the contrary, MIMO systems can use the information from multipath components to improve the accuracy of the estimation.

The key concept for the positioning in MIMO settings is the selection of an appropriate model for the propagation channel. With the proper channel model the location problem can be solved using sequential Monte Carlo methods, also called particle filtering, see Doucet *et al.* (2001). In this paper we will investigate the performance of particle filtering in MIMO system settings.

2 State-space model and particle filtering

The positioning problem in a MIMO setting consists in the estimation of the receive antenna coordinates at time t given the signal strength measurements at the receiver end up to this time, if the transmitted signal is known. In a state-

space model framework this problem corresponds to computation of the filtering probability density function and estimation of the expected values of the state variables.

Consider a discrete state-space model with additive noise,

$$\begin{cases} z_{t+1} &= f(z_t) + \varepsilon_t, \\ u_t &= h(z_t) + e_t, \end{cases} \tag{2.1}$$

where the process noise ε_t and the measurement noise e_t are independent random variables with known probability density functions $p_\varepsilon(\varepsilon_t)$ and $p_e(e_t)$, respectively. Arbitrary, often non-linear functions $f(z_t) : \mathbb{R}^n \to \mathbb{R}^n$ and $h(z_t) : \mathbb{R}^n \to \mathbb{R}^m$ describe the evolution of the state variables, z_t, and the measurements, u_t, over time.

Suppose that the measurements up to time t, $u_{0:t}$, are avaliable. Then the filtering probability density for the state variables, $p(z_t|u_{0:t})$, is derived using Bayes' formula,

$$p(z_t|u_{0:t}) = \frac{p(u_t|z_t)p(z_t|u_{0:t-1})}{p(u_t|u_{t-1})}, \tag{2.2}$$

where

$$p(u_t|u_{t-1}) = \int p(u_t|z_t)p(z_t|u_{0:t-1})\, dz_t.$$

This density can be used to estimate the expected values of the state variables according to

$$I(g(z_t)) = \mathsf{E}_{p(z_t|u_{0:t})}\left(g(z_t)\right) = \int g(z_t)p(z_t|u_{0:t})\, dz_t. \tag{2.3}$$

The integrals involved in (2.2) and (2.3) can be analytically evaluated only in a limited number of cases. The most important special case is the linear Gaussian state-space model, when the Kalman filtering technique is applicable. Many popular algorithms for the non-linear/non-Gaussian case, like the extended Kalman filter and Gaussian sum filter, rely on analytical approximations of the integrals Anderson and Moore (1979). The great computational power of modern computers however allows using numerical methods based on Monte Carlo integration. A complete description of sequential Monte Carlo methods can be found

in Doucet *et al.* (2000). In the next paragraphs we shall briefly explain the basic steps in the derivation of the particle filtering algorithm.

The particle filter approximates the density by a large set of M samples (particles), $\{z_{t|t}^{(i)},\ i = 1 \dots M\}$, where each particle has an associated normalized weight, $\tilde{w}_t^{(i)}$, such that $\tilde{w}_t^{(i)} \geq 0$ for all i and $\sum_{i=1}^{M} \tilde{w}_t^{(i)} = 1$. An empirical estimate of the filtering probability density function is then given by

$$\hat{p}(z_t|u_{0:t}) \approx \sum_{i=1}^{M} \tilde{w}_t^{(i)} \delta(z_t - z_{t|t}^{(i)}),$$

where $\delta(\cdot)$ is the Dirac delta function. Further, an estimate of the integral (2.3) is the weighted sample mean,

$$\hat{I}_M(g(z_t)) = \sum_{i=1}^{M} \tilde{w}_t^{(i)} g(z_{t|t}^{(i)}).$$

The particles are initialized at random points of the state space. The filter updates the particle locations and weights each time a new observation is avaliable. Firstly, the particle location is obtained by passing the current particles through the system dynamics:

$$z_{t+1|t}^{(i)} = f(z_{t|t}^{(i)}) + \varepsilon_{t+1}^{(i)}, \quad \text{where } \varepsilon_{t+1}^{(i)} \sim p_\varepsilon(\varepsilon_{t+1}).$$

The unnormalized weights $w_t^{(i)}$ are usually updated sequentially in time, and the updating coefficients are equal to the values of the conditional density of the observations, evaluated at the observed values,

$$w_{t+1}^{(i)} = w_t^{(i)} p(u_{t+1}|z_{t+1|t}^{(i)}).$$

This updating mechanism has the serious drawback that normalized weights tend to degenerate with time, in the sense that after few steps of the algorithm all but one of the normalized weights are very close to zero. As a result, a large computational effort is spent on updating trajectories with very small contribution to the final estimate. To avoid the degeneracy problem, a resampling procedure is introduced. This step consists of resampling with replacement among the predictive particles, according to the updated and normalized weights,

$$\mathbb{P}(z_{t+1}^{(i)} = z_{t+1|t}^{(j)}) = \tilde{w}_{t+1}^{(j)},\ i = 1 \dots M,$$

where normalized weights are given by

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{M} \omega_t^{(i)}}.$$

After resampling all weights are set to $1/M$.

The weight of a particle reflects how likely the obtained measurement is, given the present state. Particles with large weights have high probabilities to be drawn from the true distribution and thus have high probabilities of being resampled. At the same time, particles with low weights appear to come from the wrong distribution and have to be discarded. There are several resampling algorithms proposed in the literature, namely, simple random resampling, stratified resampling, systematic sampling and residual sampling, see Kitagawa (1996) and Liu and Chen (1998).

If the state-space model (2.1) contains a linear Gaussian sub-structure, the estimates can be improved by using a marginalized particle filter (Schön *et al.* (2006)). Consider a state-space model, that is linear in all states and with additive Gaussian noise for some states. The state vector z_t can then be split into two parts,

$$z_t = (z_t^k,\, z_t^p)',$$

where z_t^k corresponds to the states with Gaussian dynamics and z_t^p corresponds to the rest of the states. Similarly we split the vector of errors $\varepsilon_t = (\varepsilon_t^k,\, \varepsilon_t^p)'$, where $\varepsilon_t^k \sim N(0, \Sigma_\varepsilon)$ and $\varepsilon_t^p \sim p_\varepsilon(\varepsilon_t^p)$.

Then the model (2.1) can be rewritten in the following way:

$$\begin{cases} z_{t+1}^k &= A_t^k z_t^k + A_t^p z_t^p + \varepsilon_{t+1}^k, \\ z_{t+1}^p &= B_t^k z_{t+1}^k + B_t^p z_t^p + \varepsilon_{t+1}^p, \\ u_t &= h_t(z_t^p) + C_t z_t^k + e_t. \end{cases} \tag{2.4}$$

This split of the state variables and errors allows us to split the filtering probability density into two components,

$$p(z_t|u_{0:t}) = p(z_t^k|z_t^p,\, u_{0:t}) p(z_t^p|u_{0:t}).$$

Here the first term can be evaluated analytically by the Kalman filter and the second term can be estimated using particle filtering. Such a combination reduces the computational complexity of the algorithm (Karlsson *et al.* (2005)) and allows

to obtain better estimates with the same number of particles. More about this splitting technique, also called Rao-Blackwellization, can be found in Doucet *et al.* (2000).

3 Models for positioning in MIMO settings

3.1 Movement models

The transmit antenna is assumed kept at a fixed point with a fixed orientation. The receiver moves along an arbitrary trajectory and the receiving antenna turns randomly at some time points. The position of the receiver at time t is expressed by the abscise $x_{R,t}$ and the ordinate $y_{R,t}$ of the reference point.

In the Cartesian approach, these quantities are related to the horizontal and vertical velocities $\dot{x}_{R,t}$ and $\dot{y}_{R,t}$. These velocities, in turn, are assumed to follow a simple linear Markovian model with random accelerations. The evolution of these four states is described by the system of equations

$$\begin{cases} \dot{x}_{R,t+1} = \dot{x}_{R,t} + \Delta t \cdot a_{x,t+1}, \\ \dot{y}_{R,t+1} = \dot{y}_{R,t} + \Delta t \cdot a_{y,t+1}, \\ x_{R,t+1} = x_{R,t} + \Delta t \cdot \dot{x}_{R,t+1}, \\ y_{R,t+1} = y_{R,t} + \Delta t \cdot \dot{y}_{R,t+1}, \end{cases} \tag{3.1}$$

where $a_{x,t}$ and $a_{y,t}$ are independent random variables with $N(0,\ \sigma_x^2/\Delta t)$ and $N(0,\ \sigma_y^2/\Delta t)$ distributions, respectively.

In the polar approach the coordinates of the receiver are related to the speed of the receiver, $v_{R,t}$, and the direction of movement, $\varphi_{R,t}$. Both the velocity and the direction are assumed to follow simple linear Markovian models with random acceleration and turns. This gives the evolution equations

$$\begin{cases} v_{R,t+1} = v_{R,t} + \Delta t \cdot a_{t+1}, \\ \varphi_{R,t+1} = \begin{cases} \varepsilon_{t+1} & \text{with probability } \delta, \\ \varphi_{R,t} + \sqrt{\Delta t} \cdot \varepsilon_{t+1} & \\ \qquad & \text{with probability } 1-\delta, \end{cases} \\ x_{R,t+1} = x_{R,t} + \Delta t \cdot v_{R,t+1} \cos\left(\varphi_{R,t+1}\right), \\ y_{R,t+1} = y_{R,t} t + \Delta t \cdot v_{R,t+1} \sin\left(\varphi_{R,t+1}\right). \end{cases} \tag{3.2}$$

The acceleration, a_t, and the turns, ε_t, are assumed to be sequences of independent random variables with $N(0,\ \sigma_a^2/\Delta t)$ and $U[-\pi,\ \pi]$ distributions, respectively.

The dynamics of the receiving antenna orientation was modeled as

$$\psi_{R,t+1} = \begin{cases} \nu_{t+1} & \text{with probability } \delta', \\ \psi_{R,t} + \sqrt{\Delta t} \cdot \nu_{t+1} & \\ & \text{with probability } 1-\delta', \end{cases} \tag{3.3}$$

where the random turns ν_t are independent variables from a $U[-\pi,\ \pi]$ distribution.

3.2 MIMO propagation channel model

Recently, many different models for MIMO propagation channels have been proposed. An overview of these models can be found in Yu and Ottersten (2002), and the research on this topic is continued. In our simulation study we will model the channel using the geometrical approach introduced in Molisch (2004). The basic idea of this approach is to place scatterers at random and then emulate the propagation process from the transmitter to the receiver, taking into account the effect of scattering.

Consider a MIMO system with N_T transmit elements and N_R receive elements. At time t the relationship between input and output can be expressed as

$$u_t = H_t v_t + e_t, \tag{3.4}$$

where v_t is the N_T-vector of the transmitted signal, u_t is the N_R-vector of the received signal and e_t is an additive Gaussian noise term. The channel is described by the deterministic $N_R \times N_T$ channel response matrix H_t. A single element of this matrix, $h_{nm,t}$, is the impulse response from the m-th transmit to the n-th receive antenna element. Suppose there are N_S scatterers around the receiver and the transmitter. Then the impulse response is determined by

$$h_{nm,t} = \sum_{s=1}^{N_s} A_{s,t} \exp(j\varphi_s) \exp(j\zeta_{ms}) \exp(j\eta_{ns,t}). \tag{3.5}$$

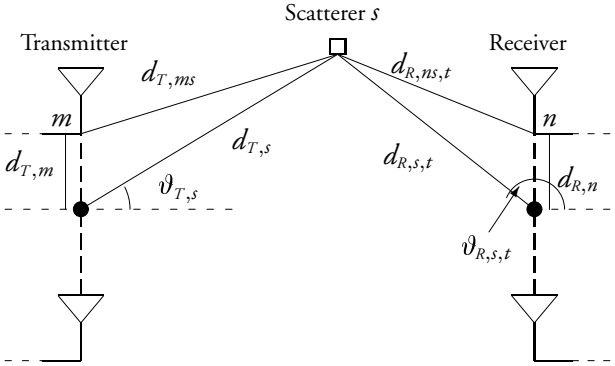


Figure 1: The distances and angular parameters used in the phase shift derivations.

Here $A_{s,t}$ denotes the amplitude damping for the path between the transmitter, scatterer s and the receiver,

$$A_{s,t} = d_{s,t}^{-\nu} \cdot \alpha_s, \tag{3.6}$$

with the total traveling distance $d_{s,t}$, random damping α_s at the scatterer and propagation coefficient ν. Note that we assume no LOS and single scattering for all paths.

The phase shift is composed of three components: random phase shift φ_s at scatterer s, phase shift ζ_{ms} at the m-th transmit antenna element and phase shift $\eta_{ns,t}$ at the n-th receive antenna element. These are derived from simple geometrical relationships and given by

$$\begin{aligned} \zeta_{ms} = \frac{2\pi}{\lambda} \cdot d_{T,m} \cdot |\sin(\vartheta_{T,s})| \\ \cdot \operatorname{sign}(d_{T,s} - d_{T,ms}), \end{aligned} \tag{3.7}$$

$$\begin{aligned} \eta_{ns,t} = \frac{2\pi}{\lambda} \cdot d_{R,n} \cdot |\sin(\vartheta_{R,s,t} + \frac{\pi}{2} - \psi_{R,t})| \\ \cdot \operatorname{sign}(d_{R,ns,t} - d_{R,s,t}), \end{aligned} \tag{3.8}$$

where λ denotes the wave length and the angular parameters and the distances are explained in Figure 1.

4 Particle filtering algorithms

Equations (3.1)–(3.3)–(3.4) and (3.2)–(3.3)–(3.4) define a state-space model with five states. The position of the receiver is involved in the measurement equation in a highly non-linear way. It influences both the amplitude damping and the phase shifts, since the coordinates of a reference point at the receiver are used in the calculation of all distances in (3.6), (3.7), and (3.8). In addition, the direction of arrival, $\vartheta_{R,s,t}$, depends on the position of the receiver at time t. In the Cartesian model the velocities, however, have linear Gaussian dynamics and do not appear in the measurement equation.

The distribution of the measurement noise e_t is the N_T-dimensional complex Gaussian distribution with zero mean and covariance matrix $\mathbf{\Sigma}_e$.

The particle filter weights equal the conditional density of the observations, given the states:

$$\begin{aligned} w_t^{(i)} &= p(u_t|z_{t|t-1}) = p_e(u_t - h(z_t)) \\ &= \frac{1}{\pi^{N_T}|\mathbf{\Sigma}_e|}\times \\ &\times \exp\left\{-(\overline{u_t - H_{i,t}v_t})'\mathbf{\Sigma}_e^{-1}(u_t - H_{i,t}v_t)\right\} \end{aligned} \tag{4.1}$$

The estimates of the position of the receiver are simply the weighted sample means over M particles, evaluated with the normalized weights:

$$(\hat{x}_{R,t},\, \hat{y}_{R,t}) = \left(\sum_{i=1}^{M} \tilde{w}_{t,i}x_{R,i,t},\, \sum_{i=1}^{M} \tilde{w}_{t,i}y_{R,i,t}\right). \tag{4.2}$$

The estimate of the direction of the receiver is the circular mean direction (Fisher (1993)):

$$\begin{aligned} S &= \sum_{i=1}^{M} \sqrt{\tilde{w}_{t,i}} \cos\psi_{R,i,t}\,, \\ C &= \sum_{i=1}^{M} \sqrt{\tilde{w}_{t,i}} \sin\psi_{R,i,t}\,, \\ \hat{\psi}_{R,t} &= \begin{cases} \tan^{-1}(S/C), \text{ if } S > 0, C > 0; \\ \tan^{-1}(S/C) + \pi, \text{ if } C < 0; \\ \tan^{-1}(S/C) + 2\pi, \text{ if } S < 0,\ C > 0. \end{cases} \end{aligned} \tag{4.3}$$

To measure the degeneracy of the algorithm we used the effective sample size estimate (Kong *et al.* (1994)), defined as

$$\hat{N}_e = \left[\sum_{i=1}^{N} [\tilde{w}_t^{(i)}]^2 \right]^{-1} . \tag{4.4}$$

Resampling was carried out if $\hat{N}_e$ fell below 60% of the total number of particles.

The filtering algorithm is specified as follows:

1. *Initialization*, $t = 0$

   (a) For $i = 1$ to M, sample

   $$(x_{R,i,0},\ y_{R,i,0})' \sim U\left([a, b] \times [a', b']\right)$$

   $$\psi_{R,i,0} \sim U[0,\ 2\pi]$$

   $$\dot{x}_{R,i,0} \sim U[c,\ d],\ \dot{y}_{R,i,0} \sim U[c,\ d]$$

   or

   $$v_{R,i,0} \sim U[c,\ d],\ \varphi_{R,i,0} \sim U[0,\ 2\pi]$$

   (b) Set $t = 1$

2. *PF time update*

   For $i = 1$ to M, move current particles according to (3.2) or (3.1) and turn the antenna according to (3.3).

3. *PF measurement update*

   (a) For $i = 1$ to M evaluate the channel matrix $H_{i,t}$, update the weights, normalize, calculate the effective sample size estimate. Estimate the position and the direction according to (4.2).

   (b) If the effective sample size estimate is less than $0.6M$, resample with replacement M particles according to the normalized weights using systematic sampling. Set all weights equal to $1/M$.

4. Set $t = t + 1$ and iterate to step 2.

The algorithm for the marginalized particle filtering combines the Kalman filter for the velocities in the Cartesian model $(\dot{x}_{R,t}, \dot{y}_{R,t})$ with the particle filter for the coordinates and the direction of the antenna.

1. *Initialization*, $t = 0$
   (a) For $i = 1$ to M, sample
   $$(x_{R,i,0},\ y_{R,i,0})' \sim U\left([a, b] \times [a', b']\right)$$
   $$\psi_{R,i,0} \sim U[0,\ 2\pi]$$
   $$\dot{x}_{R,i,0} \sim U[c,\ d],\ \dot{y}_{R,i,0} \sim U[c,\ d]$$
   (b) Set $t = 1$
2. *Resampling (PF measurement update)*
   (a) For $i = 1$ to M evaluate the channel matrix $H_{i,t}$, update the weights, normalize, calculate the effective sample size estimate. Estimate the position and the direction according to (4.2).
   (b) If the effective sample size estimate is less than $0.6M$, resample with replacement M particles according to the normalized weights using the systematic sampling procedure. Set all weights equal to $1/M$.
3. *PF time update and KF update*
   (a) KF measurement update:
   $$(\dot{x}_{R,i,t|t}, \dot{y}_{R,i,t|t}) = (\dot{x}_{R,i,t|t-1}, \dot{y}_{R,i,t|t-1}),$$
   $$P_{t|t} = P_{t|t-1}.$$
   (b) PF time update:
   For $i = 1$ to M, move current particles according to (3.2) or (3.1) and turn the antenna according to (3.3).
   (c) KF time update:
   $$\dot{x}_{R,i,t+1|t} = \frac{1}{\Delta t}\left(x_{R,i,t+1|t} - x_{R,i,t|t}\right),$$
   $$\dot{y}_{R,i,t+1|t} = \frac{1}{\Delta t}\left(y_{R,i,t+1|t} - y_{R,i,t|t}\right),$$
   $$P_{t+1|t} = P_{t|t} = \Delta t \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}.$$

4. Set $t = t + 1$ and iterate to step 3.

5 Simulations

The transmitting antenna is linear with $N_T = 3$ elements distanced by the half wave length, and the wave length $\lambda = 3/20$ m corresponding to 2 GHz frequency. The transmitter is located at the origin, $(x_T,\ y_T)' = (0, 0)'$, and the orientation of the transmit antenna is 90°.

The receiving antenna is also linear with three elements, $N_R = 3$, distanced by the half wave length. The receiver starts moving at (100, 0) with speed 6 km/h and direction 0°. The initial orientation of the antenna is 45°. The receiver moves along the trajectory with turns of size 90° and 45° during 3 minutes, and turns the antenna by 90° or 45° at some time points.

There are 45 scatterers, placed randomly within the area [0, 300] × [0, 250] m. The amplitude damping at each scatterer is simulated from a Rayleigh distribution with mean -6 dB, and the phase shifts are uniformly distributed between 0 and 2π. All these are fixed over time, and used as the filter input.

The standard deviation for signal noise is set to 10^{-8} for each antenna element in order to keep signal to noise ratio between 10 and 30 dB. Noises on different antenna elements are assumed to be independent, which gives the diagonal covariance matrix

$$\Sigma_e = \begin{pmatrix} 10^{-16} & 0 & 0 \\ 0 & 10^{-16} & 0 \\ 0 & 0 & 10^{-16} \end{pmatrix},$$

for the calculation of weights by (4.1).

The sampling rate is 100 times per sec, so $\Delta t = 0.01$ sec and $T = 18,000$. Propagation coefficient ν is set to 3.5.

Three different filters are applied: common particle filters with either Cartesian (3.1) or polar (3.2) model for the mobile movement (Filters 1 and 2, respectively) and the marginalized particle filter, based on the Cartesian model (Filter 3). All filters are run using 500 particles, with $\sigma_x^2 = \sigma_y^2 = 3$, $\sigma_a^2 = 3$ in common particle filters, and with $\sigma_x^2 = \sigma_y^2 = 2$ in the marginalized particle filter. The probabilities δ and δ' in the models of the direction of movement (3.2) and of the antenna orientation (3.3) are set to 0.01. In all three filters initial positions are sampled within the area $[95, 105] \times [-5, 5]$ m, and initial speeds are sampled from $U[1, 6]$ km/h.

To estimate the over-time performance for each filter we calculated the RMSE, based on R runs for the position estimates,

$$RMSE_t = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left[(x_t - \hat{x}_{t,r})^2 + (y_t - \hat{y}_{t,r})^2\right]}, \quad t = 1, \ldots, T,$$

and the arithmetic sample mean over R runs for the angle estimates.

Figures 2 and 3 show the true trajectory and the true direction of the receiver together with the results for three different filters. All filters seem to perform quite well and are able to follow the track, with the estimation error not more than 25 m. Filter 1 has the largest over-time mean estimation error in the position, 4.9 m, compared to 3.5 m and 3.4 m for filters 2 and 3, respectively.

The estimation error in the orientation of the receiver is quite large at the turning times, but at the next time point decreases to less than 25° for all three filters. Over-time mean errors for all three filters are around 4°.

Figure 4 displays the RMSE for these filters based on 68 runs. Over-time performance for all filters corresponds with the results from one run. Filter 1 has largest over-time mean RMSE of 16.3 m, whereas for the filters 2 and 3 over-time mean RMSE is 5.4 m and 3.9 m, respectively. The mean estimation error in the antenna orientation is displayed in Figure 5. All three filters have similar precision of estimation, with the mean error staying below 20° most of the time.

Table 1 shows the Federal Communication Commission (FCC) performance requirements for the mobile location, expressed in error probability. For example, at least 67% of the positioning errors should be smaller than 100 m. To compare our results with these requirements, we have calculated the positioning error at each time point for 68 runs of three different filters, and then evaluated 67 and 95 percentiles for each time point. Maximal over-time values for three filters are given in Table 2. Comparing these two tables, we see that the estimation accuracy for all three filters fits the FCC requirements for both network-based and mobile-based positioning.

We have also applied these three filters in more sophisticated situations, where the receiver moves along the circular or sinusoidal track and the receiving antenna makes a full round during the movement.

Results are displayed in Figure 6 and show good performance of all three filters in the estimation of both the position and the antenna orientation.

6 Conclusions

Three different particles filters were applied for mobile positioning in a MIMO settings: Filter 1, based on the Cartesian model (3.1) for the states variables; Filter 2, based on the polar model (3.2) for states, and Filter 3, marginalized particle filter. Results, averaged over 68 independent runs of these filters, show good performance, satisfying the FCC performance requirements for the mobile location in network-based positioning as well as in mobile-based positioning. The marginalized particle filter, being the combination of Kalman filter and particle filter, shows the best performance with over-time mean RMSE of 3.9 m and 95% of positioning errors below 25 m.

In our simulation we used a very simple channel model with a small number of transmit and receive antennas. It is possible to increase the number of antennas and use more sophisticated channel models (including e.g. effects of multiple scattering) without changing the filtering algorithms. In addition, the dimensionality of the particle filter state space is independent of the number of antenna elements, as well as of the number of scatterers.

Note however, that the positioning with particle filters in these settings requires large computational power. There are two reasons for that. First, the calculations involve high-dimensional matrices, with one dimension equal to the number of particles. Second, all tested filters have high resampling rate about 98%, which means that the filters degenerate and need to resample at almost every step. The practical solution to the first problem can be to reduce the number of particles. It will increase the positioning error, but at the same time decrease the computation time. A solution for the second problem is somewhat more difficult. In our filtering algorithms we sampled particles according to the system dynamics. In other words, we chose the prior distribution of states as the sampling distribution This choice gives a simple expression for the calculation of weights, but filter may perform badly if the likelihood is peaked. As a solution, one can use so-called auxiliary particle filter, discussed by Pitt and Shephard in Doucet *et al.* (2001).

References

Anderson, B. D. O., and Moore, J. B. (1979) *Optimal Filtering.* Englewood Cliffs.

Table 1: FCC requirements for mobile- and network-based positioning, expressed in error probability.

Error %	**Mobile-based**	**Network-based**
67	50 m	100 m
95	150 m	300 m

Table 2: The maximal percentiles for 68 runs of three different filters,with the maximum is taken over all time points.

Error %	**Filter 1**	**Filter 2**	**Filter 3**
67	17 m	16 m	12 m
95	34 m	24 m	25 m

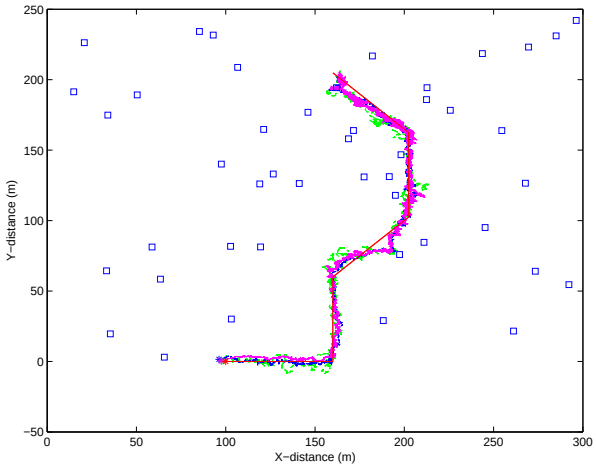


Figure 2: True trajectory and the filtering results based on one run. Stars corresponds to the initial position of the mobile (red) and to the starting positions of three filters (green: Filter 1, blue: Filter 2 and magenta: Filter 2. Squares show positions of scatterers. Green line shows the results for Filter 1, blue: Filter 2, magenta: Filter 3.

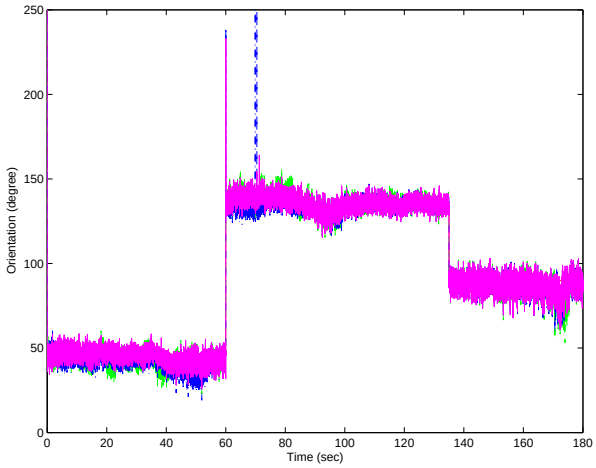


Figure 3: True orientation of antenna (red line) and the filtered one for three different filters. Over-time mean errors are 4.4°, 4.1° and 3.8° for Filter 1 (green line), Filter 2 (blue line) and Filter 3 (magenta line), respectively.

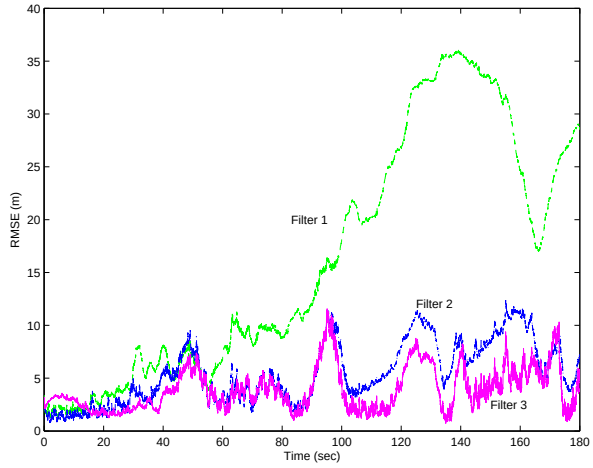


Figure 4: RMSE for the position estimate with three filters, based on 68 runs. Color code: green – Filter 1, blue – Filter 2, magenta – Filter 3.

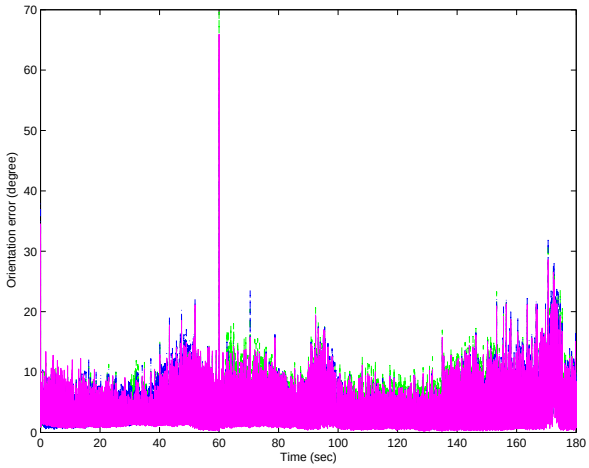


Figure 5: Mean error in the estimation of the antenna direction. Over-time mean is 5° for Filter 1 (green line), 12° for Filter 2 (blue line) and 4° for Filter 3 (magenta line).

Doucet, A., Godsill, S.,J., and Andrieu, C. (2000) On Sequential Monte-Carlo Sampling Methods for Bayesian Filtering. *Statist. Comput.*, **10:3**, pp. 197–208.

Doucet, A., de Freitas, N., and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice.* New York: Springer.

Fisher, N. I. (1993) *Statistical Analysis of Circular Data.* Cambridge University Press.

Karlsson, R, Schön, T., and Gustafsson, F. (2005) Complexity Analysis of the Marginalized Particle Filter. *IEEE Trans. Signal Proc.*, **53:11**, pp. 4408–4411.

Kitagawa, G. (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space model, *Journal of Computational and Graphical Statistics*, **5(1)**, pp. 1–25.

Kong, A., Liu, J. S., and Wong, W. H. (1994) Sequential imputations and Bayesian missing data problems. *J. Am. Stat. Assoc.*, **89**, pp. 278–288.

Liu, J. S., and Chen, E. (1998) Sequential Monte Carlo methods for dynamic systems. *J. Am. Stat. Assoc.*, **93**, pp. 1032–1044.

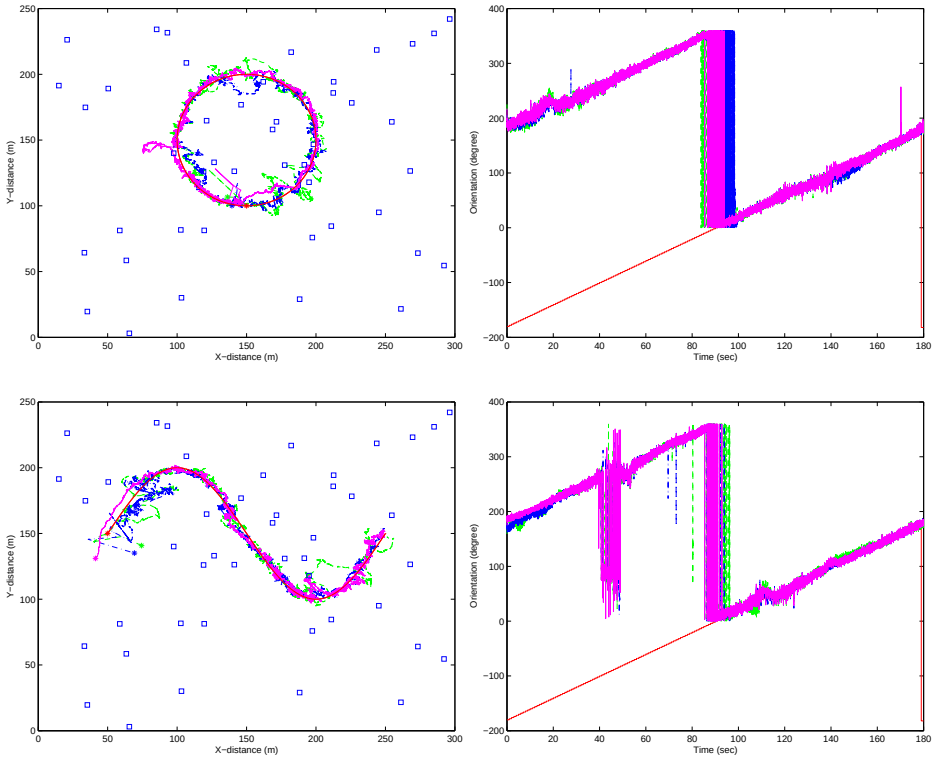


Figure 6: True trajectory (left) and true orientation of the antenna (right) together with the filtering results for the circular (upper panel) and the sinusoidal (lower panel) movement. Green lines show the results for Filter 1, blue: Filter 2, magenta: Filter 3. Over-time mean errors in the positioning do not exceed 10 m. Over-time mean errors in estimation of the antenna orientation are around 10°.

Molisch, A. F. (2004) A Generic Model for MIMO Wireless Propagation Channels in Macro- and Microcells. *IEEE Trans. Sig. Proc.*, **52:1**, pp. 61–71.

Paulraj, A. and Kailath, T. (1994) Increasing capacity in wireless broadcast systems using distributed transmission/directional reception. US Patent, 5 345 599.

Paulraj, A., Nabar, R., and Core, D. (2003) *Introduction to Space-Time Wireless Communications.* Cambridge University Press.

Schön, T., Karlsson, R., and Gustafsson, F. (2006) The Marginalized Particle Filter in Practice. *Proc. IEEE Aerospace Conf.*, Big Sky, USA.

Syrjärinne, J. (2001) Studies of modern techniques for personal positioning. PhD thesis, Tampere University of Technology.

Yu, K., and Ottersten, B. (2002) Models for MIMO propagation channels: a review. *Wireless Communications and Mobile Computing*, **2**, pp. 653–666.

C

Paper C

Sequential Monte Carlo Methods and Decomposable State–Space Models

Svetlana Bizjajeva, Tobias Rydén

Abstract

Sequential Monte Carlo methods (particle filters) for non-linear filtering have been applied since early nineties in many different areas. The convergence results and computational load of the filtering algorithms depend on the number of particles, and often a very large number of particles is required in order to obtain good estimation. From the other side, using too many particles drastically slows down the algorithm. In this paper we consider a method for particle filtering for multi-dimensional models which are decomposable in the coordinates, allowing a reduced number of particles without a large precision loss.

Key words: Particle filtering, simulations, multi-dimensional state–space models.

1 Introduction

Many problems in applied statistics, from signal processing to modeling of biological sequences, can be formulated in terms of a state-space model. The *transition equation* describes the evolution of *states*, i.e. the Markov chain $\{X_t\}_{t\geq 0}$, which is subject to interest but not observable directly. Instead one can observe another stochastic process $\{Y_t\}_{t\geq 0}$ that is linked to this chain in a way that the conditional distribution of Y_t is determined by X_t and described by the *observation equation*. The goal is to estimate the conditional distribution of the states given the observations and often one of its marginals, the *filtering distribution* of the state at time t, given the observations up to this time. This problem is known as a Bayesian filtering or optimal filtering problem. Except for a very few cases (linear Gaussian

systems and discrete finite state space), analytical solution of this problem is not possible and numerical methods have to be employed.

Particle filters, first suggested in Gordon *et al.* (1993) and throughly described in Doucet *et al.* (2000), have been shown to be an effective solution to the optimal filtering problem. They consist in approximating filtering distributions by the empirical distribution of a weighted set of particles. This method is relatively easy to implement and can be applied to wide class of models with non-linear/non-Gaussian structure; for examples, see Doucet *et al.* (2001). Summary descriptions of particle filtering (or sequential Monte Carlo) methods together with convergence results can be found in Doucet *et al.* (2001). It is known that the number of particles is very important in determining the performance of particle filters. The number of particles and the processing rate directly affect the hardware complexity of the implemented filter. Moreover, in applications like mobile tracking the measurements are collected with very high sampling rate and the filter has to work very fast in order to perform on-line estimation. For an efficient implementation it is important to use as few particles as possible at any given time.

Existing methods for changing the number of particles during filtering include the Kullback-Leibler distance (KLD) sampling (Fox (2001)) and its improved version (Soto (2005)). These methods are based on the idea of propagating a small number of particles when the density is concentrated in a small region and propagating a large number of particles in the case of a wide density region. Decreasing the number of particles becomes essential when the state space has more than one dimension. Taking a large number of particles for each dimension drastically increases the dimension of the filter and notably affects its working time.

In this paper we consider a method for particle filtering in the case when there exists a natural decomposition of the state space and a similar decomposition of the measurement equations. Instead of sampling multi-dimensional particles we sample a smaller set of particles in each dimension and then combine the resulting estimates. We demonstrate that this effectively reduces the computation time without a large precision loss.

2 Monte Carlo solution to the smoothing problem in state-space models

Consider a Markov chain $\{X_t\}_{t\geq 0}$, taking values in some measurable state space $(\mathsf{X}, \mathcal{X})$ at the discrete time-points $t \in \mathbb{Z}_+$. Let Q and ν denote the probability transition kernel, $Q(x, dz) = \mathbb{P}\left(X_{t+1} \in dz \,|\, X_t = x\right)$, and the initial distribution of the chain, respectively. This chain is partially observed through a stochastic process on Y, $\{Y_t\}_{t\geq 0}$, in a way that given $\{X_t\}$, $\{Y_t\}$ is a sequence of independent random variables such that the conditional distribution of Y_t depends on the value of X_t only. Denote by $g_t(x_t)$ the density function of Y_t given $X_t = x_t$ w.r.t. some measure μ on $(\mathsf{Y}, \mathcal{Y})$. This density and the transition kernel Q describe the state-space model in the following way:

$$\begin{aligned} X_{t+1} \,|\, X_t = x_t &\sim Q(x_t, \cdot), \\ Y_t \,|\, X_t = x_t &\sim g_t(x_t). \end{aligned} \tag{2.1}$$

Statistical inference about the hidden Markov chain $\{X_t\}_{t\geq 0}$ can be done using the observations $\{Y_t\}_{t\geq 0}$ only. The general problem, usually referred as *joint smoothing*, consists in the evaluation of the joint conditional distribution of hidden states $X_{0:s}$, given the observations $y_{0:t} \doteq (y_0, y_1, \ldots, y_t)$, $t \geq s$. The special case of the marginal distribution with $t = s$ is called the *filtering problem*, i.e. evaluating the conditional distribution of the hidden state at time t given the observations up to this time. Denote the joint smoothing distribution by $\varphi_{0:t|t}$, and the filtering distribution by $\varphi_{t|t}$.

For the particular sequence of observations $y_{0:t}$ the joint smoothing distribution satisfies

$$\varphi_{0:t|t}(dx_{0:t}) = L_t^{-1}\, \nu(dx_0) g_0(x_0) \prod_{s=1}^{t} Q(x_{s-1}, dx_s) g_s(x_s), \tag{2.2}$$

where L_t denotes the full likelihood of the observations,

$$L_t(y_0, \ldots, y_t) = \int \ldots \int \nu(dx_0) g_0(x_0) \prod_{s=1}^{t} Q(x_{s-1}, dx_s) g_s(x_s).$$

This gives a recursive update for the joint smoothing distribution. Starting with

$$\varphi_0(dx_0) = \frac{g_0(x_0)\ \nu(dx_0)}{\int g_0(x)\ \nu(dx)},$$

at time $t \geq 0$ we have

$$\varphi_{0:t+1|t+1}(dx_{0:t+1}) = \varphi_{0:t|t}(dx_{0:t})\, T_t(x_t, dx_{t+1}), \tag{2.3}$$

where T_t is the unnormalized transition kernel on $(\mathbf{X}, \mathcal{X})$ defined by

$$T_t(x, dx') = \left(\frac{L_{t+1}}{L_t}\right)^{-1} Q(x, dx') g_t(x'). \tag{2.4}$$

Except in some special cases (models with linear/Gaussian structure) the likelihood ratio in (2.4) can not be computed in closed form. This makes analytical evaluation of T_t and $\varphi_{0:t|t}$ impossible. However, it is possible to approximate the distribution of interest using a set of samples from this distribution. Moreover, if the target distribution is known only up to a constant, one can sample from an *importance distribution* which is absolutely continuous w.r.t. the target and which is easy to sample from. Then the appropriately weighted samples from the importance distribution serve for the approximation of the target one.

Let $\{\rho_{0:t}\}_{t\geq 0}$ be the family of probability measures associated with the inhomogeneous Markov chain with initial distribution ρ_0 and transition kernels $\{R_t\}_{t\geq 0}$,

$$\rho_{0:t}(dx_{0:t}) = \rho_0(dx_0) \prod_{s=0}^{t-1} R_s(x_s, dx_{s+1}). \tag{2.5}$$

This family can be used to compute importance sampling estimates for the joint smoothing distribution $\varphi_{0:t|t}$. Assume that $\varphi_0 \ll \rho_0$ and $T_t(x, \cdot) \ll R_t(x, \cdot)$ for all $t \geq 0$. Then $\varphi_{0:t|t} \ll \rho_{0:t}$ with Radon-Nikodym derivative (Billingsley (1986))

$$\frac{d\varphi_{0:t}}{d\rho_{0:t}}(x_{0:t}) = \frac{d\varphi_0}{d\rho_0}(x_0) \prod_{s=0}^{t-1} \frac{dT_s(x_s, \cdot)}{dR_s(x_s, \cdot)}(x_{s+1}). \tag{2.6}$$

Suppose we have N independent samples $\xi_{0:t}^1, \ldots, \xi_{0:t}^N$ from $\rho_{0:t}$. The importance sampling estimate of the joint smoothing distribution $\varphi_{0:t|t}(dx_{0:t})$ is given by

$$\hat{\varphi}_{0:t|t}(dx_{0:t}) = \frac{\sum_{i=1}^N \omega_t^i\, \delta_{\xi_{0:t}^i}(dx_{0:t})}{\sum_{i=1}^N \omega_t^i}, \tag{2.7}$$

where the unnormalized importance weights ω_t^i are computed recursively as

$$\omega_0^i = \frac{d\varphi_0}{d\rho_0}(\xi_0^i) \quad \text{for } i = 1, \ldots, N,$$

and

$$\omega_{t+1}^i = \omega_t^i \frac{dT_t(\xi_t^i, \cdot)}{dR_t(\xi_t^i, \cdot)}(\xi_{t+1}^i) \quad \text{for } t \geq 0,\ i = 1, \ldots, N. \tag{2.8}$$

Due to the normalization of the weights in (2.7), the importance sampling estimator remains unchanged if the weights are evaluated up to a constant only. This allows us to omit the unknown scaling factor in T_t and makes possible the practical implementation of the recursive importance sampling procedure: at each timepoint, first sample a set of *particles* $\left(\xi_{t+1}^1, \ldots, \xi_{t+1}^N\right)$ conditionally independently given the *history* $\left\{\xi_{0:t}^i,\ i = 1, \ldots, N\right\}$ from the distribution $\xi_{t+1}^i \sim R_t(\xi_t^i, \cdot)$, and then update the importance weights according to

$$\omega_{t+1}^i = \omega_t^i \times g_{t+1}(\xi_{t+1}^i) \frac{dQ(\xi_t^i, \cdot)}{dR_t(\xi_t^i, \cdot)}(\xi_{t+1}^i) \quad \text{for } i = 1, \ldots, N. \tag{2.9}$$

The main drawback of this procedure is that sequentially updated importance weights tend to degenerate and after few iterations the estimate of the joint smoothing distribution will be based in fact on one particle only. To avoid degeneracy, one has to introduce a *resampling* step, which serves to eliminate particle trajectories with low importance weights and duplicate trajectories with lage importance weights. This is achieved by sampling with replacement among the particles with probability of sampling a particle equal to its normalized importance weight.

3 Decomposition of the joint smoothing distribution

Consider the case where the state space X is a product of two spaces, $\mathsf{X} = \mathsf{X}^\alpha \times \mathsf{X}^\beta$, and the kernel Q can be decomposed in two components Q^α and Q^β defined on $(\mathsf{X}^\alpha, \mathcal{X}^\alpha)$ and $(\mathsf{X}^\beta, \mathcal{X}^\beta)$ respectively:

$$Q^\alpha(x, dz) = \mathbb{P}(X_{t+1}^\alpha \in dz \,|\, X_t^\alpha = x) \quad \text{for } x, z \in X^\alpha,$$
$$Q^\beta(x, dz) = \mathbb{P}(X_{t+1}^\beta \in dz \,|\, X_t^\beta = x) \quad \text{for } x, z \in X^\beta,$$

so that

$$Q\left((x^\alpha,\, x^\beta),\, (dz^\alpha,\, dz^\beta)\right) = Q^\alpha(x^\alpha, dz^\alpha)Q^\beta(x^\beta, dz^\beta).$$

Assume also that a similar decomposition exists for the initial distribution ν, with components ν^α and ν^β defined on X^α and X^β respectively. Finally assume that we can decompose the density function of the observations:

$$\begin{aligned} g_t(x_t) = g_t\left((x_t^\alpha, x_t^\beta)\right) &= \mathbb{P}\left((Y_t^\alpha,\, Y_t^\beta)|(X_t^\alpha, X_t^\beta) = (x_t^\alpha, x_t^\beta)\right) \\ &= \mathbb{P}(Y_t^\alpha|X_t^\alpha = x_t^\alpha)\mathbb{P}(Y_t^\beta|X_t^\beta = x_t^\beta) = g_t^\alpha(x_t^\alpha)g_t^\beta(x_t^\beta). \end{aligned}$$

In other words, the model decomposes into two separate independent submodels evolving on X^α and X^β, respectively.

Then the joint smoothing distribution has product form:

$$\begin{aligned} \varphi_{0:t|t}(dx_{0:t}) &= \varphi_{0:t|t}(dx_{0:t}^\alpha \times dx_{0:t}^\beta) \\ &= L_t^{-1}\nu^\alpha(dx_0^\alpha)g_0^\alpha(x_0^\alpha)\prod_{s=1}^{t} Q^\alpha(x_{s-1}^\alpha, dx_s^\alpha)g_s^\alpha(x_s^\alpha) \\ &\times \nu^\beta(dx_0^\beta)g_0^\beta(x_0^\beta)\prod_{s=1}^{t} Q^\beta(x_{s-1}^\beta, dx_s^\beta)g_s^\beta(x_s^\beta) \\ &\doteq \varphi_{0:t}^\alpha(dx_{0:t}^\alpha)\varphi_{0:t}^\beta(dx_{0:t}^\beta), \end{aligned} \tag{3.1}$$

and for the recursion we have

$$\begin{aligned} \varphi_0(dx_0) &= \frac{g_0^\alpha(x_0^\alpha)\nu^\alpha(dx_0^\alpha)g_0^\beta(x_0^\beta)\nu^\beta(dx_0^\beta)}{\int g_0^\alpha(x^\alpha)\nu^\alpha(dx^\alpha)g_0^\beta(x^\beta)\nu^\beta(dx^\beta)}, \\ \varphi_{0:t+1|t+1}(dx_{0:t+1}) &= \varphi_{0:t|t}(dx_{0:t})T_t^\alpha(x_t^\alpha, dx_{t+1}^\alpha)T_t^\beta(x_t^\beta, dx_{t+1}^\beta), \end{aligned} \tag{3.2}$$

where

$$\begin{aligned} T_t^\alpha(x^\alpha, dz) &\propto Q^\alpha(x^\alpha, dz)g_t^\alpha(z) \quad \text{for } x^\alpha \in \mathsf{X}^\alpha, \\ T_t^\beta(x^\beta, dz) &\propto Q^\beta(x^\beta, dz)g_t^\beta(z) \quad \text{for } x^\beta \in \mathsf{X}^\beta. \end{aligned} \tag{3.3}$$

It follows that a convenient importance distribution should also be on product form. Let $\{R_t^\alpha\}_{t\geq 0}$ and $\{R_t^\beta\}_{t\geq 0}$ be Markov transition kernels on $(\mathsf{X}^\alpha, \mathcal{X}^\alpha)$ and

$(\mathsf{X}^\beta, \mathcal{X}^\beta)$, respectively, and let ρ_0^α and ρ_0^β be probability distributions on X^α and X^β, respectively.

Take

$$\rho_0 = \rho_0^\alpha \times \rho_0^\beta \quad \text{and} \quad R_t = R_t^\alpha \times R_t^\beta \quad \text{for all } t \geq 0.$$

Then the probability measures $\{\rho_{0:t}\}_{t\geq 0}$ associated with the non-homogeneous Markov chain with initial distribution ρ_0 and transition kernels $\{R_t\}_{t\geq 0}$ are given by

$$\begin{aligned}\rho_{0:t}(dx_{0:t}) &= \rho_0(dx_0)\prod_{s=0}^{t-1} R_s(x_s, dx_{s+1}) \\ &= \rho_0^\alpha(dx_0^\alpha)\prod_{s=0}^{t-1} R_s^\alpha(x_s^\alpha, dx_{s+1}^\alpha) \times \rho_0^\beta(dx_0^\beta)\prod_{s=0}^{t-1} R_s^\beta(x_s^\beta, dx_{s+1}^\beta) \\ &\doteq \rho_{0:t}^\alpha(dx_{0:t}^\alpha)\rho_{0:t}^\beta(dx_{0:t}^\beta).\end{aligned} \tag{3.4}$$

Assume that

$$\begin{aligned}&\varphi_0^\alpha \ll \rho_0^\alpha,\ \varphi_0^\beta \ll \rho_0^\beta, \\ &T_t^\alpha(x^\alpha, \cdot) \ll R_t^\alpha(x^\alpha, \cdot),\ T_t^\beta(x^\beta, \cdot) \ll R_t^\beta(x^\beta, \cdot) \quad \forall\, t \geq 0,\ x^\alpha \in \mathsf{X}^\alpha,\ x^\beta \in \mathsf{X}^\beta.\end{aligned}$$

Then for any $t \geq 0$,

$$\begin{aligned}\varphi_{0:t|t}(dx_{0:t}) = &\frac{d\varphi_0^\alpha}{d\rho_0^\alpha}(x_0^\alpha)\left\{\prod_{s=0}^{t-1}\frac{dT_s^\alpha(x_s^\alpha, \cdot)}{dR_s^\alpha(x_s^\alpha, \cdot)}(x_{s+1}^\alpha)\right\}\rho_{0:t}^\alpha(dx_{0:t}^\alpha) \\ &\times \frac{d\varphi_0^\beta}{d\rho_0^\beta}(x_0^\beta)\left\{\prod_{s=0}^{t-1}\frac{dT_s^\beta(x_s^\beta, \cdot)}{dR_s^\beta(x_s^\beta, \cdot)}(x_{s+1}^\beta)\right\}\rho_{0:t}^\beta(dx_{0:t}^\beta).\end{aligned} \tag{3.5}$$

Suppose we have N independent samples from each of the marginal importance distributions,

$$\xi_{0:t}^{\alpha,1}, \ldots, \xi_{0:t}^{\alpha,N} \sim \rho_{0:t}^\alpha \quad \text{and} \quad \xi_{0:t}^{\beta,1}, \ldots, \xi_{0:t}^{\beta,N} \sim \rho_{0:t}^\beta.$$

All possible combinations of $\xi_{0:t}^{\alpha,i}$ and $\xi_{0:t}^{\beta,j}$ give us a sample of size N^2 from the joint importance distribution,

$$(\xi_{0:t}^{\alpha,i}, \xi_{0:t}^{\beta,j}) \sim \rho_{0:t} \quad \text{for } i,j = 1, \ldots, N.$$

Unnormalized importance weights for this sample are calculated using the corresponding Radon-Nikodym derivatives:

$$\omega_0^{ij} = \frac{d\varphi_0^\alpha}{d\rho_0^\alpha}(\xi_0^{\alpha,i})\frac{d\varphi_0^\beta}{d\rho_0^\beta}(\xi_0^{\beta,j}) \doteq \omega_0^{\alpha,i}\omega_0^{\beta,j} \quad \text{for } i,j = 1,\dots,N, \tag{3.6}$$

and for $t \geq 0$,

$$\begin{aligned}\omega_{t+1}^{ij} &= \omega_t^{\alpha,i}\frac{dT_s^\alpha(\xi_s^{\alpha,i},\cdot)}{dR_s^\alpha(\xi_s^{\alpha,i},\cdot)}(\xi_{s+1}^{\alpha,i}) \times \omega_t^{\beta,j}\frac{dT_s^\beta(\xi_s^{\beta,j},\cdot)}{dR_s^\beta(\xi_s^{\beta,j},\cdot)}(\xi_{s+1}^{\beta,j}) \\ &= \omega_{t+1}^{\alpha,i}\omega_{t+1}^{\beta,j} \quad \text{for } i,j = 1,\dots,N.\end{aligned} \tag{3.7}$$

Denote

$$\Omega_t \doteq \sum_{i,j}\omega_t^{ij}, \quad \Omega_t^\alpha \doteq \sum_i \omega_t^{\alpha,i}, \quad \Omega_t^\beta \doteq \sum_j \omega_t^{\beta,j},$$

so that $\Omega_t = \Omega_t^\alpha\Omega_t^\beta$.

Then the importance sampling estimate of the joint smoothing distribution $\varphi_{0:t|t}$ is given by

$$\begin{aligned}\hat{\varphi}_{0:t|t}(dx_{0:t}) &= \frac{\sum_{i,j=1}^N \omega_t^{ij}\delta_{(\xi_{0:t}^{\alpha,i},\xi_{0:t}^{\beta,j})}(dx_{0:t}^\alpha \times dx_{0:t}^\beta)}{\Omega_t} \\ &= \sum_{i,j=1}^N \frac{\omega_t^{\alpha,i}}{\Omega_t^\alpha}\delta_{\xi_{0:t}^{\alpha,i}}(dx_{0:t}^\alpha)\frac{\omega_t^{\beta,j}}{\Omega_t^\beta}\delta_{\xi_{0:t}^{\beta,j}}(dx_{0:t}^\beta) = \hat{\varphi}_{0:t|t}^\alpha(dx_{0:t}^\alpha)\hat{\varphi}_{0:t|t}^\beta(dx_{0:t}^\beta),\end{aligned} \tag{3.8}$$

and one can estimate the two components of joint smoothing distribution separately, using two samples $\{\xi_{0:t}^{\alpha,i},\ i = 1,\dots,N\}$ and $\{\xi_{0:t}^{\beta,j},\ j = 1,\dots,N\}$ with importance weights $\omega_t^{\alpha,i}$ and $\omega_t^{\beta,j}$, respectively.

In this case the importance sampling procedure with optional resampling step consists of the following steps:

1. *Initialization, $t = 0$.*
   Sample independently $\xi_0^{\alpha,1},\dots,\xi_0^{\alpha,N}$ from ρ_0^α and $\xi_0^{\beta,1},\dots,\xi_0^{\beta,N}$ from ρ_0^β, and set

$$\omega_0^{\alpha,i} = g_0^\alpha(\xi_0^{\alpha,i})\frac{d\nu^\alpha}{d\rho_0^\alpha}(\xi_0^{\alpha,i}), \quad \omega_0^{\beta,j} = g_0^\beta(\xi_0^{\beta,j})\frac{d\nu^\beta}{d\rho_0^\beta}(\xi_0^{\beta,j}) \quad \text{for } i,j = 1,\dots,N.$$

Set $t = 1$.

2. *Propagation and updating weights*

   (a) Sample new sets of particles

   $$\{\xi_{t+1}^{\alpha,i},\ i = 1,\dots,N\} \quad \text{and} \quad \{\xi_{t+1}^{\beta,j},\ j = 1,\dots,N\}$$

   conditionally independently given previous history from the distributions $R_t^{\alpha}(\xi_t^{\alpha,i},\cdot)$ and $R_t^{\beta}(\xi_t^{\beta,j},\cdot)$ respectively. Form $\xi_{0:t+1}^{\alpha,i} = (\xi_{0:t}^{\alpha,i},\ \xi_{t+1}^{\alpha,i})$ and $\xi_{0:t+1}^{\beta,j} = (\xi_{0:t}^{\beta,j},\ \xi_{t+1}^{\beta,j})$.

   (b) Compute the updated importance weights $\omega_{t+1}^{\alpha,i}$ and $\omega_{t+1}^{\beta,i}$ as in (2.9).

   (c) Calculate importance sampling estimates using (3.8) and combine them to obtain an estimate of the joint distribution.

3. *Resampling*
   Resample with replacement N particles from each of sets $\{\xi_{0:t+1}^{\alpha,i}, i = 1,\dots,N\}$ and $\{\xi_{0:t+1}^{\beta,j}, j = 1,\dots,N\}$ according to the normalized importance weights $\frac{\omega_{t+1}^{\alpha,i}}{\Omega_{t+1}^{\alpha}}$ and $\frac{\omega_{t+1}^{\beta,j}}{\Omega_{t+1}^{\beta}}$, respectively. Reset all importance weights to a constant value.

4. Set $t = t + 1$ and go to step 2.

4 Application in target tracking

Consider a linear Gaussian state-space model for target movement observed in noise. The position of the target at time-point t is described by its coordinates at the plane, (X_t, Y_t), and related vertical and horizontal velocities which change in time due to the random accelerations. The true position is unknown because of measurement noise.

The resulting model has four states

$$\begin{cases} \dot{X}_{t+1} = \dot{X}_t + \Delta t \cdot \ddot{X}_{t+1} \\ \dot{Y}_{t+1} = \dot{Y}_t + \Delta t \cdot \ddot{Y}_{t+1} \\ X_{t+1} = X_t + \Delta t \cdot \dot{X}_t \\ Y_{t+1} = Y_t + \Delta t \cdot \dot{Y}_t, \end{cases} \tag{4.1}$$

and two observations

$$\begin{cases} M_{x,t} = X_t + \varepsilon_{x,t} \\ M_{y,t} = Y_t + \varepsilon_{y,t}, \end{cases} \tag{4.2}$$

where the accelerations $\ddot{X}_t$ and $\ddot{Y}_t$ are independent Gaussian random variables with zero mean and standard deviations $\sigma_x/\sqrt{\Delta t}$ and $\sigma_y/\sqrt{\Delta t}$, respectively. The measurement errors $\varepsilon_{x,t}$ and $\varepsilon_{y,t}$ are independent Gaussian random quantities with zero mean and standard deviations τ_x and τ_y, respectively.

In the simulation study the target started at (–600 m, 1400 m) and moved with constant speed 90 km/h along a trajectory with turns of size $\pi/4$ and $\pi/2$. The standard deviations of the measurement errors were set to 200 m for both observations, $\tau_x = \tau_y = 200$ m. Observations were collected 3 times per second ($\Delta t = 1/3$) during 3 minutes. Figure 1 (left panel) shows the true trajectory and the observations.

Our goal was to estimate the true position of the target at time t, given the observations up to this time. In this case the two-dimensional estimate of the coordinates is the weighted expectation w.r.t. the two-dimensional filtering distribution. Suppose we have a sample from this distribution; then the estimate is simply the weighted sample mean.

The states in the model (4.1) have natural split into two independent pairs $(\dot{X}_t, X_t)$ and $(\dot{Y}_t, Y_t)$. This enabled us to work with two one-dimensional Gaussian distributions, propagating and resampling particles independently. For the instrumental distribution we took $R_t = Q$ for all t. In this case, the importance weights are proportional to the conditional density of the observations, given the current position.

We applied two particle filters for this problem: a filter with N particles, based on the decomposed model and the common bootstrap filter with N^2 particles. The results were compared with the results of the Kalman filter, which is optimal for the linear Gaussian model.

The algorithm for the decomposed filter is given below.

1. *Initialization,* $t = 0$.
   Sample independently

$$\dot{X}_0^1, \ldots, \dot{X}_0^N \sim U[85, 95], \quad \dot{Y}_0^1, \ldots, \dot{Y}_0^N \sim U[85, 95],$$
$$X_0^1, \ldots, X_0^N \sim U[-700, -500], \quad Y_0^1, \ldots, Y_0^N \sim U[1300, 1500]$$

and set

$$\omega_0^{X,i} = 1/N, \quad \omega_0^{Y,j} = 1/N \qquad \text{for } i,j = 1,\dots,N.$$

Set $t = 1$.

2. *Propagation and updating weights*

   (a) Sample

   $$\dot{X}_t^i \sim N(\dot{X}_{t-1}^i, \Delta t \cdot \sigma_x^2), \quad \dot{Y}_t^i \sim N(\dot{Y}_{t-1}^i, \Delta t \cdot \sigma_y^2) \text{ for } i = 1,\dots,N.$$

   Calculate $\left(X_t^1,\dots,X_t^N\right)$ and $\left(Y_t^1,\dots,Y_t^N\right)$ according to (4.1).

   (b) Given the measurements $\left(M_{x,t}, M_{y,t}\right)$, compute the importance weights

   $$\omega_t^{X,i} = g_X(X_t^i), \quad \omega_t^{Y,j} = g_Y(Y_t^j),$$

   where $g_X(X_t^i)$ and $g_Y(Y_t^i)$ correspond to the densities of $N(X_t^i, \tau_x^2)$ and $N(Y_t^i, \tau_y^2)$, respectively.

   (c) Calculate the position estimates

   $$\hat{X}_t = \sum_{i=1}^{N} \frac{\omega_t^{X,i}}{\Omega_t^X} X_t^i \quad \text{and} \quad \hat{Y}_t = \sum_{i=1}^{N} \frac{\omega_t^{Y,j}}{\Omega_t^Y} Y_t^j.$$

3. *Resampling*
   Resample with replacement N particles from $\{(\dot{X}_t, X_t)^1,\dots,(\dot{X}_t, X_t)^N\}$ and N particles from $\{(\dot{Y}_t, Y_t)^1,\dots,(\dot{Y}_t, Y_t)^N\}$ according to the normalized importance weights, using systematic sampling (Carpenter *et al.* (1999)). Reset all importance weights to $1/N$.

4. Set $t = t + 1$ and go to step 2.

Note that we estimate marginal filtering distribution at time t and hence do not have to save the full particle trajectories up to this time. The algorithm was implemented on-line, i.e. we updated current estimates at the moment when a new observation became avaliable.

In step 2(b) in the equivalent common bootstrap filter we evaluated joint importance weights according to

$$\omega_t^i = g_X(X_t^i) g_Y(Y_t^i) \quad \text{for } i = 1,\dots,N^2,$$

and calculated the position estimates using these weights:

$$\hat{X}_t^1 = \sum_{i=1}^{N} \frac{\omega_t^i}{\Omega_t} X_t^i \quad \text{and} \quad \hat{Y}_t^1 = \sum_{i=1}^{N} \frac{\omega_t^i}{\Omega_t} Y_t^i.$$

In step 3 we resampled with replacement N^2 particles from

$$\{(\dot{X}_t, \dot{Y}_t, X_t, Y_t)^1, \ldots, (\dot{X}_t, \dot{Y}_t, X_t, Y_t)^{N^2}\}$$

according to the normalized importance weights using systematic sampling. The initialization and propagation steps remain unchanged.

Filtering results with $N = 50$, $\sigma_x = \sigma_y = 10$ are displayed in Figure 1 (right panel). The decomposed particle filter had a larger over-time mean error, 93 m, compared to 87 m and 85 m for the bootstrap filter and the Kalman filter, respectively. However, due to the smaller number of particles, the decomposed filter required much less computational power than the common bootstrap filter. In this simulation the CPU working time (2.39 GHz) for the bootstrap filter with 2500 particles was 131 s, whereas the decomposed filter with 50 particles took only 1.5 s of CPU working time. Figure 2 displays the mean CPU working time, evaluated over 100 runs, and the average RMSE for different number of particles. The decomposed filter with 100 particles took on average 2 s, compared to 22 minutes for the corresponding bootstrap filter, and had an average RMSE only 5 meters larger.

In the second simulation we worked with exactly the same settings except that the observational noise was simulated form a t-distribution with 5 degrees of freedom, which has more heavy tails than the Gaussian one. Figure 3 displays the mean CPU working time, evaluated over 100 runs, and the average RMSE for different number of particles.

We close this section with a brief discussion of the above results in a theoretical framework. Denote by $\varphi_{t|t}^N$ the estimate of the filtering distribution $\varphi_{t|t}$ with N particles and by $\|\xi\|_p \doteq (\mathbb{E}|\xi|^p)^{1/p}$ the L_p-norm of a random variable ξ. According to Del Moral and Miclo (2000), for any time-point $t \geq 0$ and any $p \geq 1$ there exists a finite constant $C_t^{(p)}$ such that

$$\|\varphi_{t|t}^N(f) - \varphi_{t|t}(f)\|_p \leq \frac{1}{\sqrt{N}} C_t^{(p)} \|f\|_\infty \tag{4.3}$$

for all real-valued bounded measurable functions f on $(\mathsf{X}, \mathcal{X})$, where $\|f\|_\infty$ is the supremum norm.

For an $\mathbf{R}^d$-valued random variable ξ, put $\|\xi\|_p \doteq (\mathbb{E}|\xi|_p^p)^{1/p}$. Then for a vector $f = (f_1, \ldots, f_d)'$ of real-valued bounded measurable functions, we have

$$\begin{aligned} \|\varphi_{t|t}^N(f) - \varphi_{t|t}(f)\|_p^p &= \mathbb{E}\sum_{i=1}^d |\varphi_{t|t}^N(f_i) - \varphi_{t|t}(f_i)|^p \\ &\leq \sum_{i=1}^d N^{-p/2}(C_t^{(p)})^p \|f_i\|_\infty^p, \end{aligned}$$

so that

$$\|\varphi_{t|t}^N(f) - \varphi_{t|t}(f)\|_p \leq \frac{1}{\sqrt{N}} C_t^{(p)} |(\|f_1\|_\infty, \ldots, \|f_d\|_\infty)|_p \leq \frac{1}{\sqrt{N}} C_t^{(p)} \|f\|_\infty$$

where we defined $\|f\|_\infty = \max_{1\leq i\leq d} \|f_i\|_\infty$.

For the bootstrap particle filter with N^2 particles we have

$$\|\varphi_{t|t}^{N^2}(f) - \varphi_{t|t}(f)\|_p \leq \frac{1}{N} C_t^{(p)} \|f\|_\infty.$$

For the corresponding decomposed particle filter, consider a vector-valued bounded measurable function f on product form, i.e.

$$f(x) = f((x^\alpha, x^\beta)) = f^\alpha(x^\alpha) \otimes f^\beta(x^\beta),$$

where f^α and f^β are real vector-valued bounded measurable functions on $(\mathsf{X}^\alpha, \mathcal{X}^\alpha)$ and $(\mathsf{X}^\beta, \mathcal{X}^\beta)$, respectively. Then the filtering distribution and its estimate have product form,

$$\varphi_{t|t}(f) = \varphi_{t|t}^\alpha(f^\alpha)\varphi_{t|t}^\beta(f^\beta), \qquad \varphi_{t|t}^N(f) = \varphi_{t|t}^{\alpha,N}(f^\alpha)\varphi_{t|t}^{\beta,N}(f^\beta).$$

Using the triangle inequality we obtain

$$\begin{aligned} \|\varphi_{t|t}^N(f) - \varphi_{t|t}(f)\|_p &= \|\varphi_{t|t}^{\alpha,N}(f^\alpha)\varphi_{t|t}^{\beta,N}(f^\beta) - \varphi_{t|t}^\alpha(f^\alpha)\varphi_{t|t}^\beta(f^\beta)\|_p \\ &\leq \|\varphi_{t|t}^{\alpha,N}(f^\alpha)\varphi_{t|t}^{\beta,N}(f^\beta) - \varphi_{t|t}^\alpha(f^\alpha)\varphi_{t|t}^{\beta,N}(f^\beta)\|_p \\ &+ \|\varphi_{t|t}^\alpha(f^\alpha)\varphi_{t|t}^{\beta,N}(f^\beta) - \varphi_{t|t}^\alpha(f^\alpha)\varphi_{t|t}^\beta(f^\beta)\|_p \\ &\leq \|f^\beta\|_\infty \frac{1}{\sqrt{N}} A_t^{(p)} \|f^\alpha\|_\infty + \|f^\alpha\|_\infty \frac{1}{\sqrt{N}} B_t^{(p)} \|f^\beta\|_\infty \\ &= \frac{1}{\sqrt{N}} \max(A_t^{(p)}, B_t^{(p)}) \|f^\alpha\|_\infty \|f^\beta\|_\infty, \end{aligned}$$

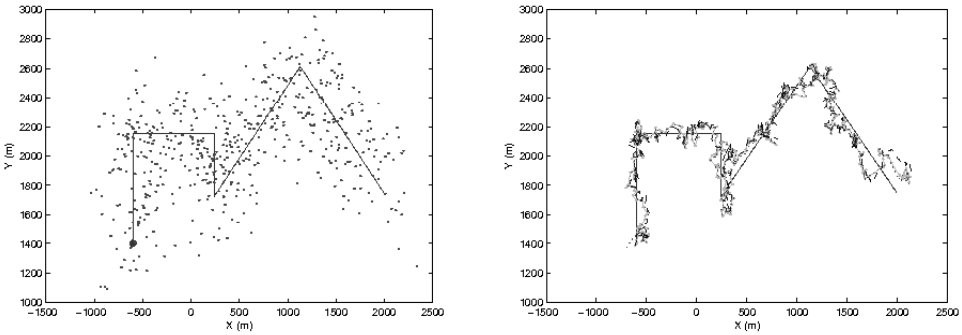


Figure 1: Left panel: True trajectory (line) and measurements (dots). The circle marks the starting point. Right panel: True trajectory (solid line) and filtered ones with the decomposed filter (dashed line), the bootstrap filter (dash-dotted line), Kalman filter (dotted line).

where $A_t^{(p)}$ and $B_t^{(p)}$ correspond to the constant $C_t^{(p)}$ for the spaces $\mathbf{X}^\alpha$ and $\mathbf{X}^\beta$ respectively (and the systems on these spaces). The heuristic is now that both $A_t^{(p)}$ and $B_t^{(p)}$ are (much) smaller than $C_t^{(p)}$, because the dimensions of $\mathbf{X}^\alpha$ and $\mathbf{X}^\beta$ are less than the dimension of $\mathbf{X}$.

In general, there are two sources that contribute to the error of the particle filter, relative to the actual true state at a given time point. First, the observations are noisy. Secondly, part of error is caused by the discrepancy between between particle filter and the exact filter; this error is what is discussed above. Increasing the number of particles helps to reduce the second source of error, but the size of the error also depends on the constants $C_t^{(p)}$ etc., as described above. Although the bootstrap filter converges to the optimal filter faster, at rate N if the filter has N^2 particles, it is still beaten by the decomposed particle filter which has only $2N$ particles, but whose constant is so much smaller that the resulting total error is only slightly larger that for the bootstrap filter, but at a much smaller computational cost.

References

Billingsley, P. (1986) *Probability and Measure*, 2nd edition. John Wiley & Sons.

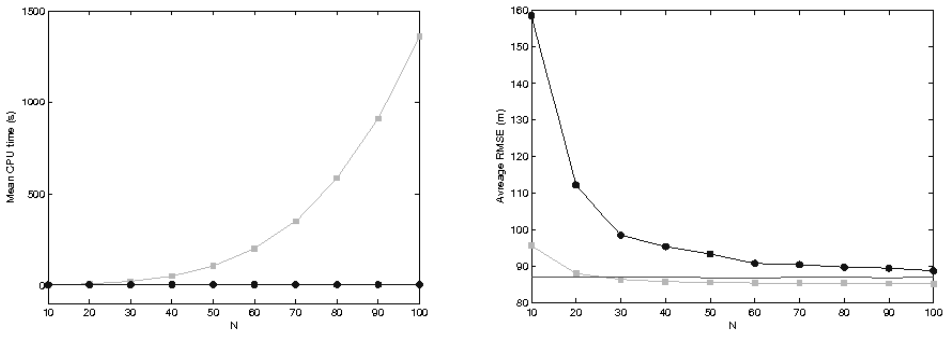


Figure 2: Left panel: Mean CPU time depending on the number of particles. Line with circles corresponds to the results for the decomposed filter, line with squares corresponds to the results for the common bootstrap filter. Right panel: Average RMSE for two different filters depending on the number of particles: the decomposed filter (line with circles), the bootstrap filter (line with squares). Results for the Kalman filter (solid line) are shown for reference.

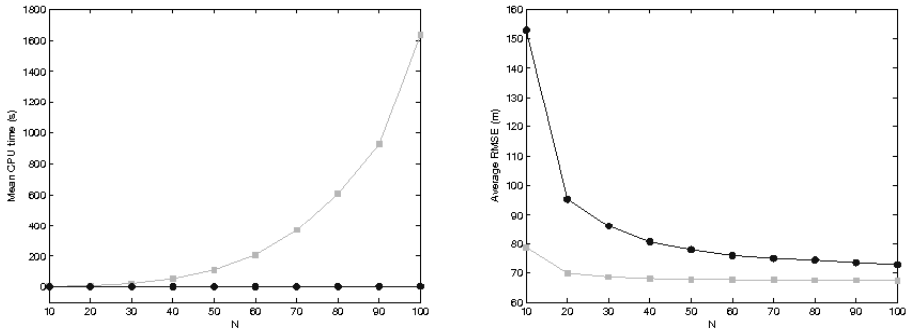


Figure 3: Results for the simulation with t-distribution. Left panel: Mean CPU time depending on the number of particles. Line with circles corresponds to the results for the decomposed filter, line with squares corresponds to the results for the common bootstrap filter. Right panel: Average RMSE for two different filters depending on the number of particles: the decomposed filter (line with circles), the bootstrap filter (line with squares).

Cappé, O., Moulines, É., and Rydén, T. (2005) *Inference in Hidden Markov Models.* New York: Springer.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999) Improved particle filter for nonlinear problems. *IEEE Proc. Radar, Sonar and Navigation*, **146**.

Del Moral, P., and Miclo, L. (2000) Branching and Interacting Particle Systems Approximations of Feyman-Kac Formulae with Applications to Non-Linear Filtering. *Lecture Notes in Mathematics*, **1729**, Springer-Verlag, Berlin.

Doucet, A., Godsill, S. J., and Andrieu, C. (2000) On Sequential Monte-Carlo Sampling Methods for Bayesian Filtering. Statist. Comput., **10:3**, pp. 197–208.

Doucet, A., de Freitas, N., and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice*. New York: Springer.

Fox, D. (2001) KLD-Sampling: Adaptive Particle Filters and Mobile Robot Localization. In Advances in Neural Information Processing Systems (NIPS), pp. 26–32.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993) Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEEE Proc. Comm. Radar Signal Proc.*, **140**, pp. 107–113.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002) Particle Filters for Positioning, Navigation and Tracking, *IEEE Trans. Signal Proc.*, **50:2**, pp. 425–437.

Kitagawa, G. (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space model, *Journal of Computational and Graphical Statistics*, **5:1**, pp. 1–25.

Kong, A., Liu, J. S., and Wong, W. H. (1994) Sequential imputations and Bayesian missing data problems. *J. Am. Stat. Assoc.*, **89**, pp. 278–288.

Soto, A. (2005) Self adaptive particle filter. *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (eds. L. P. Kaelbling and A. Saffiotti), Edinburgh, Scotland, pp. 1398–1406.

D

Paper D

Sequential Monte Carlo Methods: strategies for changing the instrumental sample size

Svetlana Bizjajeva, Tobias Rydén

Abstract

SMC methods constitute a class of IS methods which serve for the approximation of the target measure with the samples from the importance measure. The quality of the approximation naturally depends on the involved sample size. In this paper we introduce different strategies to increase the size of an importance sample and investigate performance of suggested algorithms on the simulation examples.

Key words: SMCM, state–space models, sample size, two–stage sampling

1 Introduction

During the last decades, sequential Monte Carlo methods have drawn much attention as a powerful tool for approximating the joint smoothing distributions for non-linear and non-Gaussian state-space models. Originating from pioneering work of Gordon *et al.* (1993), sequential Monte Carlo (or *particle filtering*) techniques have been widely applied in different areas such as signal processing or finance, see for instance Doucet *et al.* (2001).

Particle filtering algorithms approximate the target distribution by a properly weighted set of samples (also called *particles*) and are usually proceed in two main steps. In the first *mutation* step, one samples the collection of particles from the *instrumental* distribution and assigns to each particle an appropriate *importance* weight, which reflects how likely the current observation is, given this particle.

The resulting weighted sample approximates the target distribution. The second, *selection* step is optional and serves to eliminate particles with low weights and duplicate particles with high weights. This step consists in resampling with replacement among the particles according to their normalised importance weights.

The quality of the approximation with the particle filter depends on two key factors, namely, the number of particles used in the approximation and the choice of the instrumental distribution. Adaptation of the instrumental distribution is based on two stage sampling procedure, proposed by Pitt and Shephard (1999). Recently, Cornebise*et al.* (2008) proposed algorithms for adapting instrumental distribution using chi-square and Kullback-Leibler (KL) distances between the target and proposal distributions. They suggest an estimate of chi-square distance based on coefficient of variation, establish empirical estimate of the KL distance and propose an algorithm adapting the importance distribution by minimisation of the estimated distances.

The KL divergence have been employed by Fox (2003) in adapting the number of particles. The idea of so-called *KLD–sampling* is to increase the sample size until the KL distance between the true and estimated target distribution is below a given threshold. Another approach is introduced by Legland and Oudjane (2006) and is based on the idea to increase the size of the particle sample until the total weight mass reaches a positive threshold. Recently, Soto (2005) suggested a revised version of KLD–sampling, with the threshold corrected by the variance of the importance sampling estimates, and combined this algorithm with the adaptation of the importance sampling distribution.

Adaptive methods discussed above are concerned with the criteria for decision whether the sample size is enough to provide an adequate estimate or more particles are needed. In this paper we address the question how, given the collection of particles after resampling step, increase the sample size during mutation procedure. The first and the most obvious choice is simply to duplicate the initial particles and mutate independently from the enlarged collection. In addition we will try an algorithm introducing correlation between the particles, mutated from the same ancestor. In both these strategies the number of offspring is the same for all particles and is set a priori. Alternatively, one can mutate the particles independently, but with different number of offspring for each particle, which is determined using the observations.

The paper is organised in the following way. First, we describe the general state-space model and recall the standard algorithm for the particle filtering. In

the second section we review the two-stage sampling procedure. Next we introduce three different approaches for changing number of particles and formulate the algorithms. The last sections contains application of suggested algorithms to different models and we conclude this section with the discussion of the results.

2 Sequential Monte Carlo approximation to the joint smoothing distribution

Consider a general state-space model that is a stochastic process on two levels. On the first level we have a discrete time Markov chain $\{X_n,\ n \geq 0\}$ taking values in some measurable state space $(\mathsf{X}, \mathcal{X})$. The initial distribution of the chain ν and the transition probability kernel Q which describes the evolution of the chain in time are assumed to be known, but the chain itself is not observable. Instead one can observe another stochastic process $\{Y_n,\ n \geq 0\}$ taking values in some measurable space $(\mathsf{Y}, \mathcal{Y})$. This process is linked to the first level chain in the following way. Given $\{X_n\}_{n\geq 0}$, $\{Y_n\}_{n\geq 0}$ is a sequence of independent random variables such that the conditional distribution of Y_n depends on X_n only. Usually X is referred to as a *hidden* chain or *states*, and Y is called the *observation* process. The conditional density of the observations together with the initial distribution and the transition kernel of the hidden chain fully describe a state-space model,

$$\begin{aligned} X_0 &\sim \nu, \\ X_{n+1} \,|\, X_n = x_n &\sim Q(x_n, \cdot), \\ Y_n \,|\, X_n = x_n &\sim g_n(x_n). \end{aligned} \tag{2.1}$$

In general the structure of X and Y is not specified, but we can consider both X and Y being subsets of $\mathbb{R}^c$ and $\mathbb{R}^d$, respectively. Models with a countable X are usually referred as *hidden Markov models* in the literature.

Suppose we are given a sequence of observations, $y_1, \ldots, y_n$. The main interest is in estimating the joint smoothing distribution of the states given the observations, $\varphi_{0:n|n} \triangleq \mathbb{P}(X_{0:n}|y_{0:n})$ and often one of its marginals, the *filtering* distribution $\varphi_{n|n} \triangleq \mathbb{P}(X_n|y_{0:n})$.

Using Bayes' rule, one can derive recursive formulas for the joint smoothing

distribution,

$$\begin{aligned} \varphi_0(dx_0) &= \frac{g_0(x_0)\ \nu(dx_0)}{\int g_0(x)\ \nu(dx)}\,, \\ \varphi_{0:n+1|n+1}(dx_{0:n+1}) &= \varphi_{0:n|n}(dx_{0:n})\ T_n^u(x_n, dx_{n+1})\,, \end{aligned} \tag{2.2}$$

where T_n^u is the unnormalised transition kernel on $(\mathsf{X}, \mathcal{X})$ defined by

$$T_n^u(x, dx') = \left(\frac{L_{n+1}}{L_n}\right)^{-1} Q(x, dx') g_{n+1}(x') \tag{2.3}$$

and L_n denotes the full likelihood of the observations at time n. Unfortunately, the likelihood ratio in (2.3) cannot be computed in closed form except for a few cases – linear Gaussian models and models with countable state space. For models with more complicated structure, approximation methods have to be employed.

Denote by $\{\rho_{0:n}\}_{n\geq 0}$ the family of probability measures associated with the inhomogeneous Markov chain with initial distribution ρ_0 and transition kernels $\{R_n\}_{n\geq 0}$,

$$\rho_{0:n}(dx_{0:n}) = \rho_0(dx_0) \prod_{k=0}^{n-1} R_k(x_k, dx_{k+1}). \tag{2.4}$$

This family can be used as an instrumental (importance) distribution to compute estimates of the joint smoothing distribution $\varphi_{0:n|n}$ recursively in time. Assume that φ_0 is absolutely continuous with respect to ρ_0 and that $T_n^u(x, \cdot)$ is absolutely continuous with respect to $R_n(x, \cdot)$ for all $n \geq 0$. Then $\varphi_{0:n|n}$ is absolutely continuous with respect to $\rho_{0:n}$ with Radon-Nikodym derivative (Billingsley (1995))

$$\frac{d\varphi_{0:n}}{d\rho_{0:n}}(x_{0:n}) = \frac{d\varphi_0}{d\rho_0}(x_0) \prod_{k=0}^{n-1} \frac{dT_k^u(x_k, \cdot)}{dR_k(x_k, \cdot)}(x_{k+1}). \tag{2.5}$$

Suppose we have N independent samples $\left\{\xi_{0:n}^i\right\}_{i=1}^N$ from the importance distribution $\rho_{0:n}$ and associated importance weights $\left\{\omega_n^i\right\}_{i=1}^N$. The sequential importance sampling estimate of the joint smoothing distribution $\varphi_{0:n|n}(dx_{0:n})$ is given by

$$\hat{\varphi}_{0:n|n}(dx_{0:n}) = \sum_{i=1}^N \frac{\omega_n^i}{\sum_{j=1}^N \omega_n^j}\ \delta_{\xi_{0:n}^i}(dx_{0:n})\,, \tag{2.6}$$

where the unnormalised importance weights are computed recursively as

$$\omega_0^i = \frac{d\varphi_0}{d\rho_0}(\xi_0^i) \quad \text{for } i = 1, \ldots, N,$$

and

$$\omega_{n+1}^i = \omega_n^i \times g_{n+1}(\xi_{n+1}^i)\frac{dQ(\xi_n^i, \cdot)}{dR_n(\xi_n^i, \cdot)}(\xi_{n+1}^i) \quad \text{for } i = 1, \ldots, N. \tag{2.7}$$

The general algorithm for the sequential importance sampling with resampling is presented below.

Algorithm 1: standard SISR algorithm

`Initialisation: Draw an i.i.d. sample` $\xi_0^1, \ldots, \xi_0^N$ `from` ρ_0 `and set`

$$\omega_0^i = g_0(\xi_0^i)\frac{d\nu}{d\rho_0}(\xi_0^i), \quad i = 1, \ldots, N.$$

`For` $n = 0, 1, \ldots$

- `Sampling: Draw` $\tilde{\xi}_{n+1}^1, \ldots, \tilde{\xi}_{n+1}^N$ `conditionally independently given` $\{\xi_{0:n}^1\}_{i=1}^N$ `from the importance distribution`

  $$\tilde{\xi}_{n+1}^i \sim R_n(\xi_n^i, \cdot), \quad i = 1, \ldots, N$$

  `Compute the updated importance weights`

  $$\omega_{n+1}^i = \omega_n^i g_{n+1}(\tilde{\xi}_{n+1}^i)\frac{dQ_n(\xi_n^i, \cdot)}{dR_n(\xi_n^i, \cdot)}(\tilde{\xi}_{n+1}^i) \quad i = 1, \ldots, N.$$

- `Resampling (optional): Draw, conditionally independently given` $\{\xi_{0:n}^i, \tilde{\xi}_{n+1}^j\}_{i,j=1}^N$ `the indices` $I_{n+1}^1, \ldots, I_{n+1}^N$ `from the multinomial distribution with probabilities`

  $$\frac{\omega_{n+1}^1}{\sum_{i=1}^N \omega_{n+1}^i}, \ldots, \frac{\omega_{n+1}^N}{\sum_{i=1}^N \omega_{n+1}^i}.$$

  `Reset all importance weights to a constant value.`
  `If the resampling step is not applied, set`

  $$I_{n+1}^i = i, \text{ for } i = 1, \ldots, N.$$

- `Trajectory update:` Set $\xi^i_{0:n+1} = (\xi^{I^i_{n+1}}_{0:n}, \tilde{\xi}^{I^i_{n+1}}_{n+1})$.

The simplest case of the algorithm with resampling at each step and *prior* importance kernel $R_n = Q$ is usually called the *bootstrap filter*. It is very easy to implement, but sometimes it has poor performance especially when there is a large mismatch between the prior and the target distribution. Another choice for the importance kernel is the *optimal* kernel which is simply the normalised version of T^u_n,

$$T_n(x, dx') = \frac{Q(x, dx')g_{n+1}(x')}{\int Q(x, dx')g_{n+1}(x')}. \tag{2.8}$$

The optimal kernel incorporates the information both on the dynamics of states and on the current observations. However, sampling directly form the optimal kernel and computation of the associated importance weights

$$\gamma_n(x) = \int Q(x, dx')g_{n+1}(x')$$

might not be feasible. Different approaches to sampling from the optimal kernel include accept-reject algorithms and local approximation for the optimal kernel. Most known is the two-stage sampling procedure which was introduced as the *auxiliary particle filter* in the work of Pitt and Shephard (1999).

3 Two-stage sampling with prior kernel

Recall the recursive form of the joint smoothing distribution from (2.2) and rewrite it in the self-normalised form using the optimal kernel defined in (2.8),

$$\varphi_{0:n+1|n+1}(dx_{0:n+1}) = \frac{\varphi_{0:n|n}(dx_{0:n})\gamma_n(x_n)T_n(x_n, dx_{n+1})}{\int \varphi_{0:n|n}(dx_{0:n})\ \gamma_n(x_n)}.$$

Now plug in the importance sampling estimate defined in (2.6) to obtain

$$\tilde{\varphi}_{0:n+1|n+1}(dx_{0:n+1}) = \sum_{i=1}^{N} \frac{\omega^i_n \gamma_n(\xi^i_n)}{\sum_{j=1}^N \omega^j_n \gamma_n(\xi^j_n)}\ \delta_{\xi^i_{0:n}}(dx_{0:n})T_n(\xi_n, dx_{n+1}), \tag{3.1}$$

which defines a finite mixture distribution. Sampling from this distribution is carried out by first sampling the trajectory $\xi_{0:n}^I$ with probability proportional to $\omega_n^I \gamma_n(\xi_n^I)$ and then appending a $(n+1)$-st component drawn from the optimal distribution $T_n(\xi_n^I, \cdot)$.

In general this cannot be accomplished so easily because the kernels T_n are usually not available in closed form. Instead one can use importance sampling approach again, taking the prior kernel Q as an instrumental kernel. Construct a measure on X^{n+2} of the form

$$\rho_{0:n+1}(dx_{0:n+1}) = \sum_{i=1}^{N} v_i \delta_{\xi_{0:n}^i}(dx_{0:n})\, Q(\xi_n^i, x_{n+1}). \tag{3.2}$$

Here $v_1, \ldots, v_N$ are some normalised weights. Then the finite mixture distribution defined in (3.1) is dominated by this instrumental distribution with Radon-Nikodym derivative

$$\begin{aligned}\frac{d\tilde{\varphi}_{0:n+1|n+1}}{d\rho_{0:n+1}}(x_{0:n+1}) &= \sum_{i=1}^{N} \frac{\omega_n^i \gamma_n(\xi_n^i)}{v_i \sum_{j=1}^{N} \omega_n^j \gamma_n(\xi_n^j)} \mathbb{1}_{\{\xi_{0:n}^i\}}(x_{0:n}) \frac{T_n(\xi_n^i, \cdot)}{Q(\xi_n^i, \cdot)}(x_{n+1}) \\ &= \sum_{i=1}^{N} \frac{\omega_n^i}{v_i \sum_{j=1}^{N} \omega_n^j \gamma_n(\xi_n^j)} \mathbb{1}_{\{\xi_{0:n}^i\}}(x_{0:n}) g_{n+1}(x_{n+1}).\end{aligned}$$

Thus to a sampled trajectory $(\xi_0^i, \ldots, \xi_n^i, x_{n+1})$ is associated an importance weight proportional to $\omega_n^i g_{n+1}(x_{n+1}) v_i^{-1}$. When the weights v_i are uniform, we recover the bootstrap filter weights proportional to $\omega_n^i g_{n+1}(x_{n+1})$.

Once the joint smoothing distribution is estimated, the approximation of the filtering distribution is obtained by restricting the estimates to the last component. In the following we will focus on the filtering distribution, but the presented algorithms can be applied to the joint smoothing distribution as well.

4 Strategies for changing the number of particles

Suppose that we are given a weighted sample of size N approximating the filtering distribution at time n. We wish to transform this sample into a weighted sample of size αN approximating the filtering distribution at time $n+1$. The most obvious and simple choice is to duplicate each of the initial particles α times and mutate them independently according to the prior kernel. The use of multiple offspring

was suggested by Rubin (1987): an increase in the number of distinct particles in the importance sampling step will increase the number of distinct particles after the resampling step. For an illustrative example of this finding see Cappé *et al.* (2005), pp. 309–310. Moreover, we suggest a modification of the sampling step, introducing correlation between particles mutated from the same ancestor. In the one-dimensional case the correlation is introduced using a sample $(u_1, \ldots, u_\alpha)$ of correlated draws from the uniform distribution by the inverse transform method, so that the mutation step in the standard SISR algorithm is modified. Note that we resample at each time step in order to maintain the same number of particles in the mutation step.

Algorithm 2: Bootstrap filter with correlated mutation

Initialisation: Draw an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N$ from ν and set

$$\omega_0^i = g_0(\xi_0^i), \quad i = 1, \ldots, N.$$

For $n = 0, 1, \ldots$

- Sampling: For $i = 1, \ldots, N$ draw $u_0^i \sim \mathrm{U}(0,1)$ and set $\tilde{\xi}_{n+1}^{i,j} = F_Q^{-1}(u_j^i)$, where F_Q is the cumulative distribution function corresponding to the prior kernel $Q(\xi_n^i, \cdot)$, and

  $$u_j^i = \langle u_0^i + \frac{j-1}{\alpha} \rangle, \quad j = 1, \ldots, \alpha$$

  with $\langle x \rangle$ denoting the fractional part of x.
  Combine all particles into a large sample $\tilde{\xi}_{n+1}^1, \ldots, \tilde{\xi}_{n+1}^{\alpha N}$ and compute the importance weights

  $$\omega_{n+1}^i = g_{n+1}(\tilde{\xi}_{n+1}^i) \quad i = 1, \ldots, \alpha N.$$

- Resampling: Draw, conditionally independently given $(\{\xi_{0:n}^i\}_{i=1}^N$ and $\{\tilde{\xi}_{n+1}^j\}_{j=1}^{\alpha N})$ the indices $I_{n+1}^1, \ldots, I_{n+1}^N$ from the multinomial distribution with probabilities

  $$\frac{\omega_{n+1}^1}{\sum_{j=1}^{\alpha N} \omega_{n+1}^j}, \ldots, \frac{\omega_{n+1}^{\alpha N}}{\sum_{j=1}^{\alpha N} \omega_{n+1}^j}.$$

  Reset all importance weights to a constant value.

- Trajectory update: Set $\xi_{n+1}^i = \tilde{\xi}_{n+1}^{I_{n+1}^i}$.

We expect that the algorithm with correlated mutations will produce more stable estimates, especially in the case of multidimensional particles. For example, consider the positioning problem. In the common filter with α replications one moves a particle α times from the same initial position. Instead, in the correlated sampling approach we suggest to move a particle only once and obtain remaining proposals simply by reflecting this first draw in $\alpha - 1$ directions symmetrically around the initial position. In this way the algorithm explores the state space in a more systematic way, potentionally improving the performance of the filter.

In the mutation step of the previous algorithm we duplicate each particle the same number of times, α, and this number is a priori set to some constant value. But when choosing which particle to mutate we would like to favor particles that tend to locate their offspring in the regions where the density of the observation, g_{n+1} is large. The bootstrap filter does nothing particular to favor such particles. On the other hand we can use an auxiliary filtering approach adapting the weights v_i along the way as we are sampling, thus gradually increasing the weights of those particles that tend to give large g_{n+1} when mutated.

In order to do this we start by exploring the following scenario. Assume that we have sampled M trajectories from a measure ρ as in (3.2) with normalised weights v_i. Now we switch to a different importance sampling measure ρ' of the same form, by changing the weights to some other normalised weights v'_i and sampling one more trajectory. We thus have a total of $M+1$ sampled trajectories and one can think of them as being sampled from the measure

$$\rho'' = \frac{M}{M+1}\rho + \frac{1}{M+1}\rho',$$

with normalised weights

$$v''_j = \frac{M}{M+1}v_i + \frac{1}{M+1}v'_i.$$

With this new measure ρ'' the importance weight of a sampled trajectory is proportional to $\omega_n^I g_{n+1}(x)/v''_I$, where x is the current element (at time $n+1$) of the trajectory and I is the index of the particle from which this current position was mutated.

One could imagine now the following strategy for sampling. First mutate each existing particle once. This is equivalent to having $v^{1,i} = 1/N$. Then for

each particle ξ_n^i compute the mean of the g_{n+1} over all positions mutated from this particle so far. Denote this mean by $\bar{g}_{n+1}^i$, compute new weights $v^{1,i} \propto \bar{g}_{n+1}^i$ and use them to select a new particle to mutate, thus obtaining another sampled trajectory. After each new trajectory sampling re-compute the means $\bar{g}_{n+1}^i$, weights $v^{1,i}$ and update the first stage weights $v^{2,i}$. After sampling αN particles in total re-compute all importance weights using the final first stage weights $v^{2,i}$. We summarise the proposed strategy in the algorithm below.

Algorithm 3: Adaptive filter with prior kernel

Initialisation: Draw an i.i.d. sample $\xi_0^1, \ldots, \xi_0^N$ from ν and set

$$\omega_0^i = g_0(\xi_0^i), \quad i = 1, \ldots, N.$$

For $n = 0, 1, \ldots$

- Sampling:
  - Draw $\tilde{\xi}_{n+1}^1 \ldots, \tilde{\xi}_{n+1}^N$ conditionally independently given previous particles, from $Q(\xi_n^i, \cdot)$, $i = 1, \ldots, N$. Compute the selection weights

    $$v_0^{1,i} = \bar{g}_{n+1}^i, \; i = 1, \ldots, N,$$

    and set first stage weights $v_{n+1}^{2,i}$ to $1/N$ for $i = 1, \ldots, N$.
  - For $k = N+1, \ldots, \alpha N$
    Select I by drawing one element from multinomial distribution with probabilities equal to the normalised selection weights. Sample $\tilde{\xi}_{n+1}^k \sim Q(\xi_n^I, \cdot)$.
    Update the first stage weights

    $$v_{n+1,k}^{2,i} = \frac{k-1}{k} v_{n,k-1}^{2,i} + \frac{1}{k} v_{k-1}^{1,i}, \quad i = 1, \ldots, N.$$

    Recompute the mean $\bar{g}_{n+1}^I$ and the selection weight $v_k^{1,I}$.
  - Compute the importance weights

    $$\omega_{n+1}^i = \frac{1}{v_{n+1,\alpha N}^{2,i}} g_{n+1}(\tilde{\xi}_{n+1}^i) \quad i = 1, \ldots, \alpha N.$$

- Resampling: Draw, conditionally independently given $\{\xi_{0:n}^i\}_{i=1}^N$ and $\{\tilde{\xi}_{n+1}^j\}_{j=1}^{\alpha N}$ the indices $I_{n+1}^1, \ldots, I_{n+1}^N$ from the multinomial distribution with probabilities

  $$\frac{\omega_{n+1}^1}{\sum_{j=1}^{\alpha N} \omega_{n+1}^j}, \ldots, \frac{\omega_{n+1}^{\alpha N}}{\sum_{j=1}^{\alpha N} \omega_{n+1}^j}.$$

`Reset all importance weights to a constant value.`

- `Trajectory update: Set` $\xi_{n+1}^i = \tilde{\xi}_{n+1}^{I_{n+1}^i}$.

In the next section we compare the performance of the described schemes on a few examples.

5 Simulations

5.1 ARCH model

As a first example we consider the Gaussian autoregressive heteroscedasticity (ARCH) model observed in noise, which is described by the equations

$$\begin{aligned} X_{n+1} \,|\, X_n = x_n &\sim \mathrm{N}(0, \beta_0 + \beta_1 x_n^2), \\ Y_n \,|\, X_n = x_n &\sim \mathrm{N}(x_n, \sigma^2). \end{aligned} \tag{5.1}$$

We compared three different filters: the common bootstrap filter with α independent replications per particle, the bootstrap filter with α correlated replications per particle (algorithm 2) and the filter with α_i independent replications per particle (algorithm 3). The experiment was repeated for the case of informative and non-informative observations with parameters $(\beta_0 \ \beta_1 \ \sigma) = (0.9 \ 0.5 \ 0.2)$ and $(\beta_0 \ \beta_1 \ \sigma) = (0.9 \ 0.5 \ 4)$, respectively. In each case 16 observations were simulated according to the model (5.1) and the corresponding set of parameters. For these observational records, each filter approximated 250 filter means using $N = 500$ particles, and with αN particles in the mutation step, where α took values from 2 to 5. The resulted mean square errors based on reference values obtained with the standard bootstrap filter with 10^6 particles are displayed in Figures 1 and 2.

For the case of informative observations introducing correlation does not improve the performance except for some time points and $\alpha \geq 3$. Note that for these time points the improvement grows with number of replications, with the MSE almost 10 times smaller for $n = 10$ and $\alpha = 5$. The adaptive filter is more elaborative and more computationally intensive, but has larger MSE than the filters with independent or correlated replications. Moreover, it shows no improvement over the common bootstrap filter with 500 particles only and sometimes even performs worse. This might be explained by the fact that while choosing a particle

for the mutation the adaptive filter repeatedly selects only a few particular particles. As a consequence, after the second-stage resampling the resulted particle cloud is originated from a few ancestors.

If the observations are non-informative, the correlated sampling beats the filter with independent mutations, but the improvement is not very large, and again increases with the number of replications as in the informative case.

5.2 Two-dimensional tracking

Consider a linear Gaussian state-space model for target movement observed in noise. The position of the target at time point t is described by its coordinates in the plane, (X_n, Y_n), and is changing in time with random vertical and horizontal velocities. The true position is unknown because of measurement noise.

The resulting model has two states

$$\begin{cases} X_{n+1} = X_n + \Delta t \cdot \dot{X}_{n+1}, \\ Y_{n+1} = Y_n + \Delta t \cdot \dot{Y}_{n+1}, \end{cases} \tag{5.2}$$

and two observations

$$\begin{cases} M_{x,n} = X_n + \varepsilon_{x,n}, \\ M_{y,n} = Y_n + \varepsilon_{y,n}, \end{cases} \tag{5.3}$$

where the speeds $\{\dot{X}_n\}_{n\geq 1}$ and $\{\dot{Y}_n\}_{n\geq 1}$ are independent Gaussian random variables with zero mean and standard deviations $\sigma_x/\sqrt{\Delta t}$ and $\sigma_y/\sqrt{\Delta t}$, respectively. The measurement errors $\varepsilon_{x,n}$ and $\varepsilon_{y,n}$ are independent Gaussian random quantities with zero mean and standard deviations τ_x and τ_y, respectively.

In the simulation study the target started at the point (100 m, 800 m) and moved with constant speed of 60 km/h along a trajectory which includes turns and circular movement. The standard deviations of the measurement errors were set to 100 m for both observations, $\tau_x = \tau_y = 100$ m. Observations were collected 3 times per second ($\Delta t = 1/3$) during 3 minutes. Figure 3 (left panel) shows the true trajectory and the observations.

As in the previous example, three different filters were applied for this problem, each estimating the position of the target with $N = 250$ particles and with αN particles in the mutation step, $\alpha = 8$. The performance was compared using the root mean square error (RMSE) relatively to the estimates obtained by the

Kalman filter $(\bar{X}_n,\ \bar{Y}_n)$,

$$RMSE_n = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left[(\hat{X}_{n,r} - \bar{X}_n)^2 + (\hat{Y}_{n,r} - \bar{Y}_n)^2\right]},$$

with $R = 100$ runs.

The plot of RMSEs is displayed in the Figure 3, right panel. The adaptive filter shows similar performance as the filter with independent replications. The RMSE for the correlated filter is larger for all time points.

The overall conclusion we draw from the simulation study of the proposed algorithms is that both approaches are not significantly better than the standard one, at least for the considered examples. Selecting the most promising particles for further mutation may lead to the poor performance if only a few particles are selected. In this case large particle swarm is originated from just a couple of particles, which may increase the overall variance of the estimates. Regarding the filter with correlated mutations we notice that it improves the standard algorithm a little for the non-informative ARCH model. However, the scheme to introduce the dependence between the particles was proposed ad hoc, without detailed investigation of the correlation structure. We suggest that there exist some specific correlation structures that provide better estimates and recommend this topic for future research.

References

Billinglsey, P. (1995) *Probability and Measure.* Wiley, 3rd ed.

Cappé, O., Moulines, É., and Rydén, T. (2005) *Inference in Hidden Markov Models*. New York: Springer.

Cornebise, Moulines, É, Olsson, J. (2008) Adaptive Methods for Sequential Importance Sampling with Application to State Space models. *Preprints in Mathematical Sciences 2008:4*, Lund University.

Doucet, A., de Freitas, N., and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice.* New York: Springer.

Fox, D. (2003) Adapting the sample size in particle filters through KLD–sampling. *International Journal of Robotic Research*,**22**, pp. 985 –1004.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993) Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEEE Proc. Comm. Radar Signal Proc.*, **140**, pp. 107–113.

Legland, F., and Oudjane, N. (2006) A sequential particle algorithm that keeps the particle system alive. Rapport de Recherche 5836, INRIA. URL `ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-5826.pdf`.

Pitt, M. K., and Shephard, N. (1999) Filtering via simulation: Auxiliary particle filters. *J. Am. Statist. Assoc.*, **87**, pp. 493–499.

Rubin, D. B. (1987) A noninterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when the fraction of missing information is modest: the SIR algorithm (discussion of Tanner and Wong). In *J. Am. Statist. Assoc.*, **82**, pp 534–546.

Soto, A. (2005) Self adaptive particle filter. *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (eds. L. P. Kaelbling and A. Saffiotti), Edinburgh, Scotland, pp. 1398–1406.

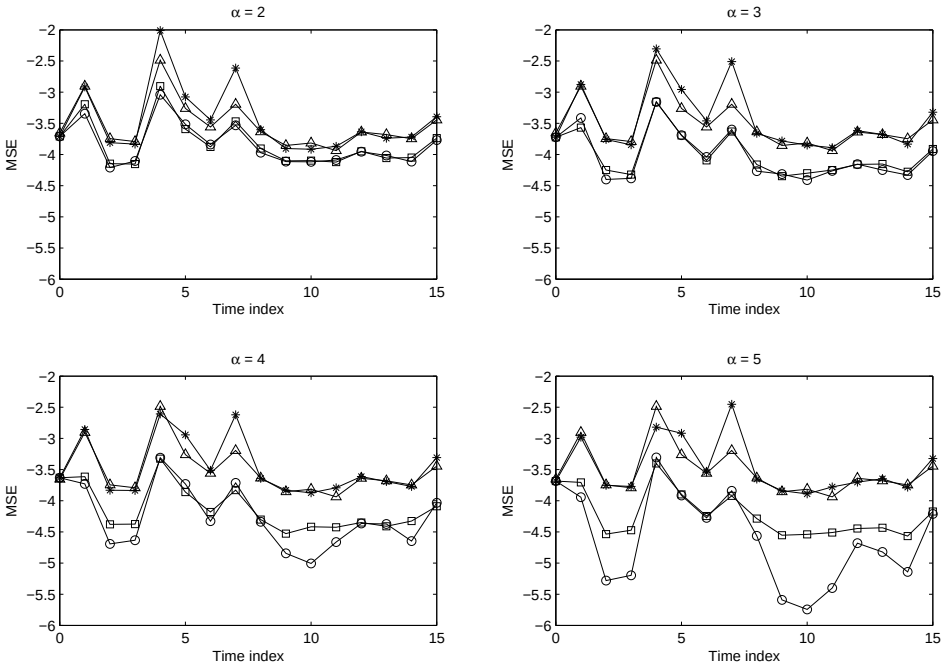


Figure 1: MSE on $\log_{10}$-scale for the bootstrap filter with replications(□), the correlated filter (○) and the adaptive filter (∗) for different α, informative ARCH model. The MSE values are computed using 500 particles and 250 runs of each algorithm. Results for the common bootstrap filter with 500 particles (△) are shown for the reference.

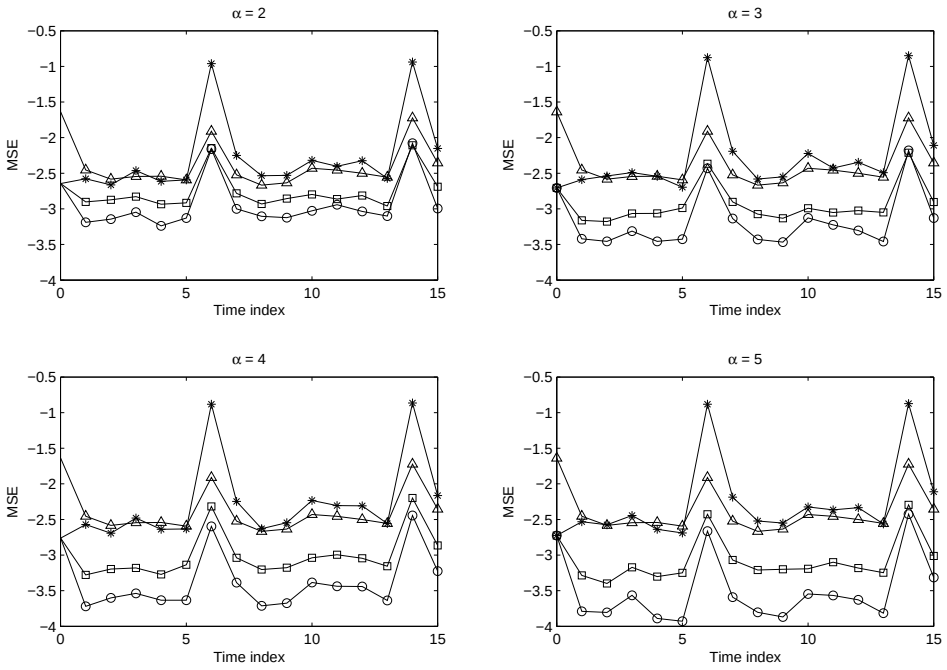


Figure 2: MSE on $\log_{10}$- scale for bootstrap filter with replications (□), the correlated filter (○) and the adaptive filter (∗) for different α, non-informative ARCH model. Results for the common bootstrap filter with 500 particles (△) are shown for the reference.

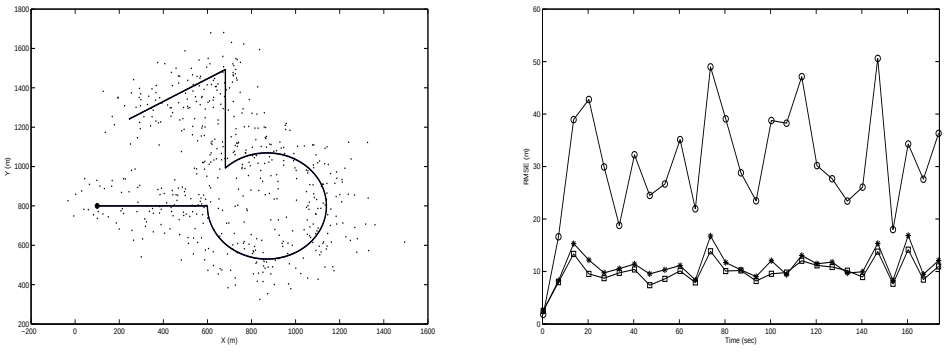


Figure 3: Left: True trajectory of the target (line) and measured positions (dots). The black circle denotes the starting position. Right: RMSE for bootstrap filter with replications (□), the correlated filter (○) and the adaptive filter (∗).

E

Paper E

Antithetic Sampling for Sequential Monte Carlo Methods with Application to State Space Models

Svetlana Bizjajeva, Jimmy Olsson

Abstract

In this paper we cast the idea of antithetic sampling, widely used in standard Monte Carlo simulation, into the framework of sequential Monte Carlo methods. A version of the standard auxiliary particle filter (Pitt and Shephard, 1999) is proposed where the particles are mutated blockwise in such a way that all particles within each block are, firstly, offspring of a common ancestor and, secondly, negatively correlated conditionally on this ancestor. By examining the weak limit of a central limit theorem describing the convergence of the algorithm, we conclude that the asymptotic variance of the produced Monte Carlo estimates can be straightforwardly decreased by means of antithetic techniques when the particle filter is close to fully adapted, which involves approximation of the so-called optimal proposal kernel. As an illustration, we apply the method to optimal filtering in state space models.

Key words: Antithetic sampling, central limit theorem, optimal filtering, optimal kernel, particle filter, permuted displacement method, state space models

1 Introduction

Sequential Monte Carlo (SMC) *methods* — alternatively termed *particle filters* — refer to a collection of algorithms which approximate recursively a sequence (often called the *Feynman-Kac flow*) of target measures by a sequence of empirical distributions associated with properly weighted samples of *particles*. These methods

have received a lot of attention during the last decade and are at present applied within a wide range of scientific disciplines. Doucet *et al.* (2001) provides a survey of recent developments of the SMC methodology from a practical viewpoint and a comprehensive treatment of theoretical aspects of basic SMC algorithms is given by Del Moral (2004).

In standard SMC methods two main operations are alternated. In the *mutation step* the particles are propagated according to a Markovian kernel and associated with importance sampling weights proportional to the Radon-Nikodym derivative of the target measure with respect to the instrumental distribution of the particles. In the subsequent *selection step* the particle sample is transformed by selecting new particles from the current (mutated) ones using the normalized importance weights as probabilities of selection. This step serves to eliminate or duplicate particles with small or large weights, respectively.

In this paper we propose a modification of the *auxiliary particle filter* (APF) (introduced originally by Pitt and Shephard, 1999) which relies on the classical idea of *antithetic sampling* used in standard Monte Carlo estimation: when estimating the expectation

$$I(f) \triangleq \int_{\mathbb{R}} f(x)p(x)\,\mathrm{d}x\ ,$$

where p is a probability density function and f is a given real-valued target function, the unbiased estimator

$$\hat{I}^N(f) \triangleq \frac{1}{2N}\sum_{i=1}^{N}[f(\xi_i)+f(\xi_i')]$$

of $I(f)$, where $\{\xi_i\}_{i=1}^N$ and $\{\xi_i'\}_{i=1}^N$ are two samples from p, is more efficient (has lower variance) than the standard Monte Carlo estimator based on a sample of $2N$ independent and identically distributed (*i.i.d.*) draws, if the variables $f(\xi_i)$ and $f(\xi_i')$ are *negatively correlated* for all $i \in \{1,\ldots,N\}$. In this setting, the variables $\{\xi_i'\}_{i=1}^N$ are referred to as *antithetic variables*. Antithetically coupled variables can be generated in different ways, and in Section 2 we discuss how this can be achieved by means of the well-known *permuted displacement method* (Arvidsen and Johnsson, 1982). In order to allow for antithetic acceleration within the SMC framework we introduce (in Section 2) a version of the standard APF where the particles are mutated *blockwise* in such a way that all particles within each block

are, firstly, offspring of a common ancestor and, secondly, statistically dependent conditionally on this ancestor. Moreover, in Section 3 we establish convergence results for our proposed method in the sense of convergence in probability and weak convergence. By examining the weak limit of the obtained central limit theorem (CLT) in Corollary 3.2 we conclude that the asymptotic variance of the produced Monte Carlo estimates is decreased when the particle filter is close to *fully adapted* (in which case close to uniform importance weights are obtained by means of approximation of the so-called *optimal kernel*, see Pitt and Shephard, 1999) and the inherent correlation structure of each block is negative. Finally, in the implementation part, Section 4, we apply our algorithm to optimal filtering in *state space models* and benchmark its performance on a noisily observed ARCH model as well as a univariate growth model. The outcome of the simulations indicates that introducing antithetically coupled particles provides, *besides* a lowered computational burden, a significant gain of precision for these models.

2 Auxiliary particle filter with blockwise correlated mutation

2.1 Notation and definitions

In order to state precisely our results and keep the presentation streamlined, we preface the description of the algorithm with some measure-theoretic notation. In the following we assume that all random variables are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. A state space Ξ is called *general* if it is equipped with a countably generated σ-field $\mathcal{B}(\Xi)$, and we denote by $\mathcal{P}(\Xi)$ and $\mathbb{B}(\Xi)$ the sets of probability measures on $(\Xi, \mathcal{B}(\Xi))$ and measurable functions from Ξ to $\mathbb{R}$, respectively. For any measure $\mu \in \mathcal{P}(\Xi)$ and function $f \in \mathbb{B}(\Xi)$ satisfying $\int_\Xi |f(\xi)|\,\mu(\mathrm{d}\xi) < \infty$ we let $\mu(f) \triangleq \int_\Xi f(\xi)\,\mu(\mathrm{d}\xi)$ denote the expectation of f under μ. A kernel K from $(\Xi, \mathcal{B}(\Xi))$ to some other state space $(\tilde{\Xi}, \mathcal{B}(\tilde{\Xi}))$ is called *finite* if $K(\xi, \tilde{\Xi}) < \infty$ for all $\xi \in \Xi$ and *Markovian* if $K(\xi, \tilde{\Xi}) = 1$ for all $\xi \in \Xi$. Moreover, a kernel K induces two operators, the first transforming a function $f \in \mathbb{B}(\Xi \times \tilde{\Xi})$ satisfying $\int_{\tilde{\Xi}} |f(\xi, \tilde{\xi})|\,K(\xi, \mathrm{d}\tilde{\xi}) < \infty$ into the function

$$\xi \mapsto K(\xi, f) \triangleq \int_{\tilde{\Xi}} f(\xi, \tilde{\xi})\,K(\xi, \mathrm{d}\tilde{\xi})$$

in $\mathbb{B}(\Xi)$; the other transforms any measure $\nu \in \mathcal{P}(\Xi)$ into the measure

$$A \mapsto \nu K(A) \triangleq \int_{\Xi} K(\xi, A)\, \nu(\mathrm{d}\xi)$$

in $\mathcal{P}(\tilde{\Xi})$. Finally, in order to describe lucidly joint distributions associated with Markovian transitions, we define the *outer product*, denoted by $K \otimes T$, of a kernel K from $(\Xi, \mathcal{B}(\Xi))$ to $(\tilde{\Xi}, \mathcal{B}(\tilde{\Xi}))$ and a kernel T from $(\Xi \times \tilde{\Xi}, \mathcal{B}(\Xi) \otimes \mathcal{B}(\tilde{\Xi}))$ to some other state space $(\bar{\Xi}, \mathcal{B}(\bar{\Xi}))$ as the kernel from $(\Xi, \mathcal{B}(\Xi))$ to the product space $\tilde{\Xi} \times \bar{\Xi}$, equipped with the product σ-algebra $\mathcal{B}(\tilde{\Xi}) \otimes \mathcal{B}(\bar{\Xi})$, given by

$$K \otimes T(\xi, A) \triangleq \iint_{\tilde{\Xi} \times \bar{\Xi}} \mathbb{1}_A(\tilde{\xi}, \bar{\xi})\, K(\xi, \mathrm{d}\tilde{\xi})\, T(\xi, \tilde{\xi}, \mathrm{d}\bar{\xi}) \tag{2.1}$$

for $\xi \in \Xi$, $A \in \mathcal{B}(\tilde{\Xi}) \otimes \mathcal{B}(\bar{\Xi})$.

2.2 Blockwise correlated mutation

In the following we say that a collection of random variables (particles) $\{\xi_{N,i}\}_{i=1}^{M_N}$, taking values in some state space Ξ, and associated nonnegative weights $\{\omega_{N,i}\}_{i=1}^{M_N}$ *targets* a probability measure $\nu \in \mathcal{P}(\Xi)$ if, denoting the weight sum by $\Omega_N \triangleq \sum_{i=1}^{M_N} \omega_{N,i}$,

$$\Omega_N^{-1} \sum_{i=1}^{M_N} \omega_{N,i} f(\xi_{N,i}) \approx \nu(f) \ ,$$

for all functions f in some specified subset of $\mathbb{B}(\Xi)$. Here $\{M_N\}_{N=0}^{\infty}$ is an increasing sequence of integers. The set $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ is referred to as a *weighted sample* on Ξ. In this paper we study the problem of transforming a weighted sample $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ targeting $\nu \in \mathcal{P}(\Xi)$ into a weighted sample $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha M_N}$, $\alpha \in \mathbb{N}^*$, targeting the probability measure

$$\mu(A) = \frac{\nu L(A)}{\nu L(\tilde{\Xi})} = \frac{\int_{\Xi} L(\xi, A)\, \nu(\mathrm{d}\xi)}{\int_{\Xi} L(\xi', \tilde{\Xi})\, \nu(\mathrm{d}\xi')} \ , \quad A \in \mathcal{B}(\tilde{\Xi}) \ , \tag{2.2}$$

where L is a finite transition kernel from $(\Xi, \mathcal{B}(\Xi))$ to $(\tilde{\Xi}, \mathcal{B}(\tilde{\Xi}))$. Feynman-Kac transitions of type (2.2) occur within a variety of fields (see Del Moral, 2004, for examples from, *e.g.*, quantum physics and biology) and in Section 4 we show how

the flow of posterior distributions of the noisily observed Markov chain (state signal) of a state space model can be generated according to (2.2). The transformation is carried out by, firstly, drawing particle positions $\{\tilde{\xi}_{N,i}\}_{i=1}^{\alpha M_N}$ according to, for $j \in \{1,\dots,M_N\}$, $k \in \{1,\dots,\alpha\}$ and $A \in \mathcal{B}(\tilde{\Xi})$,

$$\mathbb{P}\left(\tilde{\xi}_{N,\alpha(j-1)+k} \in A \middle| \mathcal{F}_{N,\alpha(j-1)+k-1}\right) = R_k(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+1}, \dots, \tilde{\xi}_{N,\alpha(j-1)+k-1}, A) ,$$

where we have defined the σ-fields $\mathcal{F}_{N,\ell} \triangleq \sigma(\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}, \{\tilde{\xi}_{N,j}\}_{j=1}^{\ell})$, $\ell \in \{0,\dots,\alpha M_N\}$, and each R_k is a Markovian kernel from $(\Xi \times \tilde{\Xi}^{k-1}, \mathcal{B}(\Xi \times \tilde{\Xi}^{k-1}))$ to $(\tilde{\Xi}, \mathcal{B}(\tilde{\Xi}))$. Hence, using the kernel outer product notation $\otimes$ defined in (2.1), the joint distribution, conditional on $\mathcal{F}_{N,\alpha(j-1)}$, the each block $\{\tilde{\xi}_{N,\alpha(j-1)+k}\}_{k=1}^{\alpha}$ can be expressed as $\bigotimes_{k=1}^{\alpha} R_k(\xi_{N,I_{N,j}}, \cdot)$. Secondly, these particles are associated with the weights

$$\tilde{\omega}_{N,\alpha(j-1)+k} = \omega_{N,j}\, \Phi_k(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+k})$$

with

$$\Phi_k(\xi, \tilde{\xi}) \triangleq \frac{\mathrm{d}L(\xi, \cdot)}{\mathrm{d}\mathcal{R}_{0,k}(\xi, \cdot)}(\tilde{\xi}) , \quad (\xi, \tilde{\xi}) \in \Xi \times \tilde{\Xi} ,$$

and, for integers $0 \le m < k$ and $A \in \mathcal{B}(\tilde{\Xi})$,

$$\begin{aligned}\mathcal{R}_{m,k}(\xi, \tilde{\xi}_{1:m}, A) &\triangleq \bigotimes_{i=m+1}^{k} R_i(\xi, \tilde{\xi}_{1:m}, \tilde{\Xi}^{k-m-1} \times A) \\ &= \int_{\tilde{\Xi}} \cdots \int_{\tilde{\Xi}} R_k(\xi, \tilde{\xi}_{1:k-1}, A) \prod_{\ell=m+1}^{k-1} R_\ell(\xi, \tilde{\xi}_{1:\ell-1}, \mathrm{d}\tilde{\xi}_\ell) ,\end{aligned}$$

where we have introduced vector notation $a_{m:n} \triangleq (a_m, a_{m+1}, \dots, a_n)$ with the convention $a_{m:n} = \varnothing$ if $m > n$. Thus $\mathcal{R}_{m,k}(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+1:\alpha(j-1)+m}, \cdot)$ is the distribution of $\tilde{\xi}_{N,\alpha(j-1)+k}$ conditionally on $\mathcal{F}_{N,\alpha(j-1)+m}$. Finally, we take $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha M_N}$ as an approximation of μ. This *blockwise mutation operation*, which extends, since it allows for statistically dependent particles within each block, the blockwise mutation operation suggested by Douc and Moulines (2005), is summarized in Algorithm 1.

Here the mutation step (2) is expressed using the kernel outer product notation $\otimes$ defined in (2.1).

Algorithm 1 Blockwise correlated mutation

Require: $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ targets ν.

1: **for** $j = 1, \ldots, M_N$ **do**

2: draw $\{\tilde{\xi}_{N,\alpha(j-1)+k}\}_{k=1}^{\alpha} \sim \bigotimes_{k=1}^{\alpha} R_k(\xi_{N,I_{N,j}}, \cdot)$,

3: set, for $k \in \{1, \ldots \alpha\}$,

$$\tilde{\omega}_{N,\alpha(j-1)+k} \leftarrow \Phi_k(\xi_{N,I_{N,j}}, \tilde{\xi}_{N,\alpha(j-1)+k}) \ ,$$

4: **end for**

5: let $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha\tilde{M}_N}$ approximate μ.

2.3 Blockwise correlated mutation with resampling

In the sequential context, where the problem consists in estimating a *sequence* of measures generated according to the mapping (2.2), it is, in order to avoid *weight degeneracy*, essential to combine the correlated blockwise mutation operation described in Algorithm 1 with a prefatory *resampling operation* where particles having small weights are eliminated and those having large ones are duplicated. As observed by Pitt and Shephard (1999) (see also Douc *et al.*, 2008, for a theoretical study), the variance of the produced SMC estimates can be reduced efficiently by introducing, as in the APF, a set $\{\psi_{N,i}\}_{i=1}^{M_N}$ of *adjustment multiplier weights* and selecting the particles with probabilities proportional to $\{\omega_{N,i}\psi_{N,i}\}_{i=1}^{M_N}$. This gives us the scheme described in Algorithm 2.

Algorithm 2 APF with blockwise correlated mutation

Require: $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ targets ν.

1: Draw $\{I_{N,j}\}_{j=1}^{\tilde{M}_N} \sim \mathcal{M}(\tilde{M}_N, \{\omega_{N,i}\psi_{N,i} / \sum_{\ell=1}^{M_N} \omega_{N,\ell}\psi_{N,\ell}\}_{i=1}^{M_N})$,

2: **for** $j = 1, \ldots, \tilde{M}_N$ **do**

3: draw $\{\tilde{\xi}_{N,\alpha(j-1)+k}\}_{k=1}^{\alpha} \sim \bigotimes_{k=1}^{\alpha} R_k(\xi_{N,I_{N,j}}, \cdot)$,

4: set, for $k \in \{1, \ldots \alpha\}$,

$$\tilde{\omega}_{N,\alpha(j-1)+k} \leftarrow \psi_{N,I_{N,j}}^{-1} \Phi_k(\xi_{N,I_{N,j}}, \tilde{\xi}_{N,\alpha(j-1)+k}) \ ,$$

5: **end for**

6: let $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha\tilde{M}_N}$ approximate μ.

2.4 Antithetic blockwise mutation with resampling

The main motivation of Pitt and Shephard (1999) for introducing the adjustment multiplier weights was the possibility of designing these in such a manner that the resulting (second stage) particle weights $\{\tilde{\omega}_{N,i}\}_{i=1}^{\alpha\tilde{M}_N}$ become close to uniform; in this case, in which the APF is referred to as *fully adapted*, the instrumental and target distributions of the APF coincide. Adapting fully the APF involves typically some approximation of the so-called *optimal* proposal kernel $L(\xi,\cdot)/L(\xi,\Xi)$. Indeed, let $\mathcal{L}$ be a kernel from $(\Xi,\mathcal{B}(\Xi))$ to $(\tilde{\Xi},\mathcal{B}(\tilde{\Xi}))$ such that $\mathcal{L}(\xi,A)\approx L(\xi,A)$ for all $\xi\in\Xi$ and $A\in\mathcal{B}(\tilde{\Xi})$; then Algorithm 2 with $\psi_{N,i}=\mathcal{L}(\xi_{N,i},\tilde{\Xi})$ and $\mathcal{R}_{0,k}(\xi,\cdot)=\mathcal{L}(\xi,\cdot)/\mathcal{L}(\xi,\tilde{\Xi})$ for all $i\in\{1,\ldots,M_N\}$ and $k\in\{1,\ldots,\alpha\}$ returns, since then

$$\begin{aligned}\tilde{\omega}_{N,\alpha(j-1)+k} &= \mathcal{L}^{-1}(\xi_{N,I_{N,j}},\tilde{\Xi})\frac{\mathrm{d}L(\xi_{N,I_{N,j}},\cdot)}{\mathrm{d}\mathcal{R}_{0,k}(\xi_{N,I_{N,j}},\cdot)}(\tilde{\xi}_{N,\alpha(j-1)+k})\\ &= \frac{\mathrm{d}L(\xi_{N,I_{N,j}},\cdot)}{\mathrm{d}\mathcal{L}(\xi_{N,I_{N,j}},\cdot)}(\tilde{\xi}_{N,\alpha(j-1)+k})\approx 1\ ,\end{aligned}$$

a close to uniformly weighted particle sample. Thus, methods for approximating the optimal kernel have been proposed by several authors; see *e.g.* Pitt and Shephard (1999) and Doucet *et al.* (2000).

For our purposes, putting the APF in a close to fully adapted mode is attractive from another point of view: the close to uniform weights render efficient antithetic acceleration of the standard APF possible, which might reduce the variance of the produced SMC estimates significantly. Hence, the aim of this paper is to justify, in theory as well as in simulations, the following algorithm in which $\mathcal{L}$ and f denote a given approximation of L and a given target function, respectively.

Step (3) in Algorithm 3 can be carried out in several different ways. The simplest way to introduce negative correlation between two real-valued random variables is to use a pair (U,U') of uniforms, where $U=r$, $U'=1-r$, and $r\sim\mathcal{U}(0,1)$ is uniformly distributed (on $(0,1)$). Such a coupling has the *extreme antithetic* (EA) *property*: if F is an arbitrary distribution function, then the correlation between $\xi=F^{\leftarrow}(U)$ and $\xi'=F^{\leftarrow}(U')$, $F^{\leftarrow}$ denoting the inverse of F, achieves the minimal possible value subject to the constraint that $\xi,\xi'\sim F$. This implies immediately that the strategy also achieves EA for variates $g(\xi)$ and $g(\xi')$, where $g:\mathbb{R}\to\mathbb{R}$ is any monotone function such that $\int g^2(\xi)\,F(\mathrm{d}\xi)<\infty$, since (U,U') achieves EA simultaneously for all F and $g(\xi)$ (and $g(\xi')$) has distribution function $F\circ g^{\leftarrow}$. This remarkable observation is related to the fact that the

Algorithm 3 APF with antithetic blockwise mutation

Require: $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ targets ν.

1: Draw $\{I_{N,j}\}_{j=1}^{\tilde{M}_N} \sim \mathcal{M}(\tilde{M}_N, \{\omega_{N,i}\mathcal{L}(\xi_{N,i}, \tilde{\Xi})/\sum_{\ell=1}^{M_N} \omega_{N,\ell}\mathcal{L}(\xi_{N,\ell}, \tilde{\Xi})\}_{i=1}^{M_N})$,

2: **for** $j = 1, \ldots, \tilde{M}_N$ **do**

3: simulate, using an appropriate family of kernels $\{R_k\}_{k=1}^{\alpha}$, a block $\{\tilde{\xi}_{N,\alpha(j-1)+k}\}_{k=1}^{\alpha} \sim \bigotimes_{k=1}^{\alpha} R_k(\xi_{N,I_{N,j}}, \cdot)$ of particles such that $\mathcal{R}_{0,k}(\xi_{N,I_{N,j}}, \cdot) = \mathcal{L}(\xi_{N,I_{N,j}}, \cdot)/\mathcal{L}(\xi_{N,I_{N,j}}, \tilde{\Xi})$ and the real-valued variables $\{f(\tilde{\xi}_{N,\alpha(j-1)+k})\}_{k=1}^{\alpha}$ are, conditionally on $\xi_{N,I_{N,j}}$, mutually negatively correlated,

4: set, for $k \in \{1, \ldots, \alpha\}$,

$$\tilde{\omega}_{N,\alpha(j-1)+k} \leftarrow \mathcal{L}^{-1}(\xi_{N,I_{N,j}}, \tilde{\Xi})\, \Phi_k(\xi_{N,I_{N,j}}, \tilde{\xi}_{N,\alpha(i-1)+k}) \ ,$$

5: **end for**

6: let $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha\tilde{M}_N}$ approximate μ.

construction (U, U') satisfies the stronger property of *negative association*, which requires that the negative correlation is preserved by monotone transformations. The following definition, adopted form Craiu and Meng (2005), extends this property to an arbitrary number of variates.

Definition 2.1 (Pairwise negative association)**.** *The random variables $\xi_1, \xi_2, \ldots, \xi_n$ are said to be* pairwise negatively associated *(PNA) if, for any nondecreasing (or nonincreasing) functions f_1, f_2 and $(i, j) \in \{1, \ldots, n\}^2$ such that $i \neq j$,*

$$\mathrm{Cov}[f_1(\xi_i), f_2(\xi_j)] \leq 0$$

whenever this covariance is well defined.

In the light of the previous it is appealing to mutate the particles in such a way that the α offspring particles of a certain block are conditionally EA given the common ancestor. A rather generic way to achieve this goes via the *permuted displacement method* (developed by Arvidsen and Johnsson, 1982) presented below, where S_α denotes the set of all possible permutations of the numbers $\{1, \ldots, \alpha\}$.

In this setting, Craiu and Meng (2005, Theorem 3) showed that the uniformly distributed variates $\{U_i\}_{i=1}^{\alpha}$ produced in Algorithm 4 are PNA for $\alpha \leq 3$.

Algorithm 4 Permuted displacement method

1: Draw $r_1 \sim \mathcal{U}(0,1)$,
2: **for** $k = 2, \ldots, \alpha - 1$ **do**
3: set $r_k = \langle 2^{k-2} r_1 + 1/2 \rangle$,
4: **end for**
5: set $r_\alpha = 1 - \langle 2^{\alpha-2} r_1 \rangle$,
6: pick a random $\sigma \in S_\alpha$,
7: **for** $k = 1, \ldots, \alpha$ **do**
8: set $U_k \triangleq r_{\sigma(k)}$.
9: **end for**

For $\alpha \geq 4$ one has not at present been able to neither prove nor refute a similar result. Thus, Step (3) of Algorithm 3 can be carried out by producing, using Algorithm 4, PNA uniforms $\{U_k\}_{k=1}^{\alpha}$ and setting, for $k \in \{1, \ldots, \alpha\}$,

$$\tilde{\xi}_{N,\alpha(j-1)+k} = F^{\leftarrow}_{k,\xi_{N,j}}[f](U_k) \ ,$$

where $F_{k,\xi}[f](x) \triangleq \mathcal{L}(\xi, \{f(\tilde{\xi}) \leq x\})/\mathcal{L}(\xi, \tilde{\Xi})$, $x \in \mathbb{R}$, denotes the conditional distribution function of the $f(\tilde{\xi}_{N,\alpha(j-1)+k})$'s given $\xi_{N,j} = \xi \in \Xi$. Since each function $F^{\leftarrow}_{k,\xi}[f]$ is monotone, it follows that $\{f(\tilde{\xi}_{N,\alpha(j-1)+k})\}_{k=1}^{\alpha}$ are conditionally EA. Of course, this method is applicable only when $F_{k,\xi}[f]$ is easy to invert; this is however not always the case and in Section 4 we present some alternative techniques for introducing negative correlation between the offspring particles.

3 Theoretical results

In this section we justify theoretically Algorithm 3 using novel results on triangular arrays obtained by Douc and Moulines (2005). The arguments rely on results describing the weak convergence of Algorithms 1 and 2 in a rather general setting.

3.1 Notation and definitions

From now on the quality of a weighted sample will be described in terms of the following asymptotic properties, adopted from Douc and Moulines (2005), where a set C of real-valued functions on Ξ is said to be *proper* if the following conditions hold: **i)** C is a linear space; **ii)** if $g \in \mathsf{C}$ and f is measurable with $|f| \leq |g|$, then $|f| \in \mathsf{C}$; **iii)** for all $c \in \mathbb{R}$, the constant function $f \equiv c$ belongs to C.

Definition 3.1 (Consistency)**.** *A weighted sample* $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ *on* Ξ *is said to be* consistent *for the probability measure* μ *and the proper set* C *if, for any* $f \in \mathsf{C}$*, as* $N \to \infty$*,*

$$\Omega_N^{-1} \sum_{i=1}^{M_N} \omega_{N,i} f(\xi_{N,i}) \xrightarrow{\mathbb{P}} \mu(f) \ ,$$
$$\Omega_N^{-1} \max_{1 \le i \le M_N} \omega_{N,i} \xrightarrow{\mathbb{P}} 0 \ .$$

Definition 3.2 (Asymptotic normality)**.** *A weighted sample* $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ *on* Ξ *is called* asymptotically normal *(AN) for* $(\mu, \mathsf{A}, \mathsf{W}, \sigma, \gamma, \{a_N\}_{N=1}^{\infty})$ *if* A *and* W *are proper and, as* $N \to \infty$*,*

$$a_N \Omega_N^{-1} \sum_{i=1}^{M_N} \omega_{N,i} [f(\xi_{N,i}) - \mu(f)] \xrightarrow{\mathcal{D}} \mathcal{N}[0, \sigma^2(f)] \quad \text{for any } f \in \mathsf{A} \ ,$$
$$a_N^2 \Omega_N^{-1} \sum_{i=1}^{M_N} (\omega_{N,i})^2 f(\xi_{N,i}) \xrightarrow{\mathbb{P}} \gamma(f) \quad \text{for any } f \in \mathsf{W} \ ,$$
$$a_N \Omega_N^{-1} \max_{1 \le i \le M_N} \omega_{N,i} \xrightarrow{\mathbb{P}} 0 \ .$$

We impose the following assumptions.

(A1) *The initial sample* $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ *is consistent for* (ν, C)*.*

(A2) *The initial sample* $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ *is AN for* $(\nu, \mathsf{A}, \mathsf{W}, \sigma, \gamma, \{a_N\}_{N=1}^{\infty})$*.*

Under **(A1)** and **(A2)**, we define

$$\begin{aligned} \tilde{\mathsf{C}} &\triangleq \{f \in \mathsf{L}^1(\tilde{\Xi}, \mu) : L(\cdot, |f|) \in \mathsf{C}\} \ , \\ \tilde{\mathsf{A}} &\triangleq \{f : L(\cdot, |f|) \in \mathsf{A}, \mathcal{R}_{0,k}(\cdot, \Phi_k^2 f^2) \in \mathsf{W}; k \in \{1, \dots, \alpha\}\} \ , \\ \tilde{\mathsf{W}} &\triangleq \{f : \mathcal{R}_{0,k}(\cdot, \Phi_k^2 |f|) \in \mathsf{W}; k \in \{1, \dots, \alpha\}\} \ . \end{aligned} \tag{3.1}$$

Moreover, let, for $f \in \tilde{\mathsf{A}}$ and $\xi \in \Xi$, assuming that $m \leq n$,

$$
\begin{aligned}
&\mathbb{M}_{m,n}(\xi, f) \\
&\triangleq \mathbb{E}\left[\Phi_m(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+m})\, \Phi_n(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+n}) f(\tilde{\xi}_{N,\alpha(j-1)+m}) f(\tilde{\xi}_{N,\alpha(j-1)+n}) \right. \\
&\qquad\qquad \left. \mid \xi_{N,j} = \xi \right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\Phi_n(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+n}) f(\tilde{\xi}_{N,\alpha(j-1)+n}) \,\middle|\, \xi_{N,j} = \xi, \tilde{\xi}_{N,\alpha(j-1)+1:\alpha(j-1)+m} \right] \right. \\
&\qquad\qquad \left. \times \Phi_m(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+m}) f(\tilde{\xi}_{N,\alpha(j-1)+m}) \,\middle|\, \xi_{N,j} = \xi \right] \\
&= \int_{\tilde{\Xi}} \cdots \int_{\tilde{\Xi}} \mathcal{R}_{m,n}(\xi, \tilde{\xi}_{1:m}, \Phi_n(\xi, \cdot) f) \\
&\qquad\qquad \times \Phi_m(\xi, \tilde{\xi}_m) f(\tilde{\xi}_m) \bigotimes_{\ell=1}^{m} R_\ell(\xi, \mathrm{d}\tilde{\xi}_1 \times \cdots \times \mathrm{d}\tilde{\xi}_m) \,,
\end{aligned}
$$

and introduce the conditional covariances

$$
\begin{aligned}
\mathbb{C}_{m,n}(\xi, f) &\triangleq \mathrm{Cov}\left[\Phi_m(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+m}) f(\tilde{\xi}_{N,\alpha(j-1)+m}), \right. \\
&\qquad \left. \Phi_n(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+n}) f(\tilde{\xi}_{N,\alpha(j-1)+n}) \,\middle|\, \xi_{N,j} = \xi \right] \qquad (3.2) \\
&= \mathbb{M}_{m,n}(\xi, f) - L^2(\xi, f) \,.
\end{aligned}
$$

3.2 Convergence of Algorithms 1 and 2

Under the assumptions above we have the following convergence results, whose proofs are found in the appendix.

Theorem 3.1. *Assume* **(A1)** *and suppose that* $L(\cdot, \tilde{\Xi}) \in \mathsf{C}$. *Then the set* $\tilde{\mathsf{C}}$ *defined in* (3.1) *is proper and the weighted sample* $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha M_N}$ *produced in Algorithm 1 is consistent for* $(\mu, \tilde{\mathsf{C}})$.

Theorem 3.2. *Let the assumptions of Theorem 3.1 hold. In addition, assume* **(A2)** *and suppose that all functions* $\mathcal{R}_{0,k}(\cdot, \Phi_k^2)$, $k \in \{1, \ldots, \alpha\}$, *belong to* W. *Moreover, assume that* $L(\cdot, \tilde{\Xi})$ *belongs to* A. *Then the sets* $\tilde{\mathsf{A}}$ *and* $\tilde{\mathsf{W}}$ *defined in* (3.1) *are proper and the weighted sample* $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha M_N}$ *produced in Algorithm 1 is AN for* $(\mu, \tilde{\mathsf{A}}, \tilde{\mathsf{W}}, \tilde{\sigma}, \tilde{\gamma}, \{a_N\}_{N=1}^{\infty})$, *where, for* $f \in \tilde{\mathsf{A}}$,

$$
\tilde{\sigma}^2(f) \triangleq \sigma^2\{L[f - \mu(f)]\}/[\nu L(\tilde{\Xi})]^2 + \sum_{(m,n) \in \{1,\ldots,\alpha\}^2} \gamma \mathbb{C}_{m,n}[f - \mu(f)]/[\alpha \nu L(\tilde{\Xi})]^2 \,,
$$

(3.3)

and, for $f \in \tilde{\mathsf{W}}$,

$$\tilde{\gamma}(f) \triangleq \sum_{k=1}^{\alpha} \gamma \mathcal{R}_{0,k}(\Phi_k^2 f)/[\alpha \nu L(\tilde{\Xi})]^2 \ .$$

Remark 3.1. *In the case where* $R_k(\xi, \tilde{\xi}_{i:k-1}, \cdot) = R(\xi, \cdot)$ *and* $\Phi_k = \Phi = \mathrm{d}L/\mathrm{d}R$*, that is, the particles within a block are mutated independently of each other, we have that* $\mathbb{C}_{m,n} = 0$ *for all* $m \neq n$*. This yields an asymptotic variance* (3.3) *of form*

$$\begin{aligned} \tilde{\sigma}^2(f) &= \sigma^2\{L[f - \mu(f)]\}/[\nu L(\tilde{\Xi})]^2 + \sum_{m=1}^{\alpha} \gamma \mathbb{C}_{m,m}[f - \mu(f)]/[\alpha \nu L(\tilde{\Xi})]^2 \\ &= \sigma^2\{L[f - \mu(f)]\}/[\nu L(\tilde{\Xi})]^2 \\ &\quad + \alpha^{-1}\{\gamma R(\Phi^2[f - \mu(f)]^2) - \gamma L^2[f - \mu(f)]\}/[\nu L(\tilde{\Xi})]^2 \ , \end{aligned} \tag{3.4}$$

which is exactly the expression obtained by Douc and Moulines (2005, Theorem 2).

We move on to the convergence of Algorithm 2. Throughout the rest of this paper assume, entirely in line with Algorithm 3, that the adjustment multiplier weights satisfy the following assumption.

(A3) *There exists a function* $\Psi : \Xi \to \mathbb{R}^+$ *such that* $\psi_{N,i} = \Psi(\xi_{N,i})$ *and* $\Psi \in \mathsf{C} \cap \mathsf{L}^1(\Xi, \nu)$.

Define

$$\begin{aligned} \bar{\mathsf{C}} &\triangleq \{f \in \mathsf{L}^1(\mu, \tilde{\Xi}) : L(\cdot, |f|) \in \mathsf{C} \cap \mathsf{L}^1(\nu, \tilde{\Xi})\} \ , \\ \bar{\mathsf{A}} &\triangleq \{\Psi^{-1}L^2(\cdot, |f|) \in \mathsf{C} \cap \mathsf{L}^1(\nu, \Xi), L(\cdot, |f|) \in \mathsf{A}, L^2(\cdot, |f|) \in \mathsf{W}, \\ &\qquad \Psi^{-1}\mathcal{R}_{0,k}(\cdot, \Phi_k^2 f^2) \in \mathsf{C} \cap \mathsf{L}^1(\nu, \Xi); k \in \{1, \ldots, \alpha\}\} \ , \\ \bar{\mathsf{W}} &\triangleq \{\Psi^{-1}\mathcal{R}_{0,k}(\cdot, \Phi_k^2 |f|) \in \mathsf{C} \cap \mathsf{L}^1(\nu, \Xi); k \in \{1, \ldots, \alpha\}\} \ ; \end{aligned} \tag{3.5}$$

now, by combining Theorem 3.2 with results obtained by Douc *et al.* (2008) we establish the convergence of Algorithm 2. This is the contents of the following corollaries whose proofs are omitted for brevity.

Corollary 3.1. *Let the assumptions of Theorem 3.1 hold and assume* **(A3)**. *Then the set* $\bar{\mathsf{C}}$ *defined in* (3.5) *is proper and the weighted sample* $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha\tilde{M}_N}$ *obtained in Algorithm 2 is consistent for* $(\mu, \bar{\mathsf{C}})$.

Corollary 3.2. *Let the assumptions of Theorem 3.1 hold and assume* **(A2)** *with* $a_N^2/M_N \to \beta$, $\beta \in [0, \infty)$. *In addition, suppose that* $\Psi \in \mathsf{A}$, $\Psi^2 \in \mathsf{W}$ *and that all functions* $\Psi^{-1}\mathcal{R}_{0,k}(\cdot, \Phi_k^2)$, $k \in \{1, \ldots, \alpha\}$, *belong to* $\mathsf{C} \cap \mathsf{L}^1(\nu, \tilde{\Xi})$. *Moreover, assume that* $\Psi^{-1}L^2(\cdot, \tilde{\Xi}) \in \mathsf{C} \cap \mathsf{L}^1(\nu, \tilde{\Xi})$, $L(\cdot, \tilde{\Xi}) \in \mathsf{A}$, *and* $L^2(\cdot, \tilde{\Xi}) \in \mathsf{W}$. *Then the sets* $\bar{\mathsf{A}}$ *and* $\bar{\mathsf{W}}$ *defined in* (3.5) *are proper and the weighted sample* $\{(\tilde{\xi}_{N,i}, \tilde{\omega}_{N,i})\}_{i=1}^{\alpha\tilde{M}_N}$ *obtained in Algorithm 2 with* $\tilde{M}_N/M_N \to \ell$, $\ell \in [0, \infty]$, *is AN for* $(\mu, \bar{\mathsf{A}}, \bar{\mathsf{W}}, \bar{\sigma}, \bar{\gamma}, \{a_N\}_{N=1}^{\infty})$, *where, for* $f \in \bar{\mathsf{A}}$,

$$\begin{aligned}\bar{\sigma}^2[\Psi](f) \triangleq{}& \sigma^2\{L[\cdot, f - \mu(f)]\}/[\nu L(\tilde{\Xi})]^2 \\ &+ \beta\ell^{-1}\nu(\Psi) \sum_{(m,n)\in\{1,\ldots,\alpha\}^2} \nu(\Psi\mathbb{M}_{m,n}\{\cdot, \Psi^{-1}[f - \mu(f)]\})/[\alpha\nu L(\tilde{\Xi})]^2\end{aligned} \tag{3.6}$$

and, for $f \in \bar{\mathsf{W}}$,

$$\bar{\gamma}[\Psi](f) \triangleq \beta\ell^{-1}\nu(\Psi) \sum_{k=1}^{\alpha} \nu[\Psi^{-1}\mathcal{R}_{0,k}(\cdot, \Phi_k^2 f)]/[\alpha\nu L(\tilde{\Xi})]^2 \, .$$

Remark 3.2. *The resampling step (1) in Algorithm 2 can, of course, be based on resampling techniques different from multinomial resampling,* e.g., residual resampling *or* Bernoulli branching. *However, we believe that the convergence results stated in Theorems 3.1 and 3.2 as well as the methodology developed above can be extended straightforwardly to these selection schemes, since their asymptotic behavior is well investigated (see Chopin, 2004, Douc and Moulines, 2005).*

3.3 Theoretical justification of Algorithm 3

In order to justify the use of antithetic variables in Algorithm 3, we examine the asymptotic variance given in (3.6). Since the first term is not at all effected by the way the particles are mutated, we direct focus to the second term and write, using

(3.2),

$$\begin{aligned}
&\beta\ell^{-1}\nu(\Psi)\sum_{(m,n)\in\{1,\ldots,\alpha\}^2}\nu(\Psi\mathbb{M}_{m,n}\{\cdot,\,\Psi^{-1}[f-\mu(f)]\})/[\alpha\nu L(\tilde{\Xi})]^2\\
&=\beta\ell^{-1}\nu(\Psi)\,\nu(\Psi L^2\{\cdot,\,\Psi^{-1}[f-\mu(f)]\})/[\nu L(\tilde{\Xi})]^2\\
&\quad+\beta\ell^{-1}\nu(\Psi)\sum_{(m,n)\in\{1,\ldots,\alpha\}^2}\nu(\Psi\mathbb{C}_{m,n}\{\cdot,\,\Psi^{-1}[f-\mu(f)]\})/[\alpha\nu L(\tilde{\Xi})]^2\,,
\end{aligned}$$

where the first term on the RHS is again independent of the correlation structure of the mutation step. The second term will be smaller than in the case where all particles within each block are mutated independently if the covariances $\mathbb{C}_{m,n}\{\cdot,\,\Psi^{-1}[f-\mu(f)]\}$ are negative for all $m\neq n$; however, since

$$\Psi^{-1}(\xi)\,\Phi(\xi,\tilde{\xi})\approx 1,\quad \text{for all } (\xi,\tilde{\xi})\in\Xi\times\tilde{\Xi}$$

in the close to fully adapted case, it holds that

$$\mathbb{C}_{m,n}\{\xi,\,\Psi^{-1}[f-\mu(f)]\}\approx\mathrm{Cov}\left[f(\tilde{\xi}_{N,\alpha(j-1)+m}),f(\tilde{\xi}_{N,\alpha(j-1)+n})\middle|\,\xi_{N,j}=\xi\right]\,, \tag{3.7}$$

which is negative when the functions $f(\tilde{\xi}_{N,\alpha(j-1)+k})$ are negatively correlated.

In addition, it is possible to relate the performance of the antithetic SMC scheme in Algorithm 3 to that of the standard APF (for which $\alpha=1$). More specifically, we establish a criterion (depending on the model and target function under consideration) which guarantees that introducing antithetic variates yields a strictly more accurate (in terms of variance) *and* computationally more efficient algorithm than the standard APF. In order to keep the particle population size constant, *i.e.* having $\tilde{M}_N=M_N$, through a run of Algorithm 3 for a given block size α, only a fraction $\tilde{M}_N=\lceil M_N/\alpha\rceil$ (yielding $\ell=1/\alpha$ in Corollary 3.2) of the original particle population should be selected at the resampling operation. In this case Corollary 3.2 provides, using (3.2), the asymptotic variance

$$\begin{aligned}
\bar{\sigma}^2[\Psi](f)&\triangleq\sigma^2\{L[\cdot,f-\mu(f)]\}/[\nu L(\tilde{\Xi})]^2\\
&\quad+\beta\alpha\nu(\Psi)\sum_{(m,n)\in\{1,\ldots,\alpha\}^2}\nu(\Psi\mathbb{M}_{m,n}\{\cdot,\,\Psi^{-1}[f-\mu(f)]\})/[\alpha\nu L(\tilde{\Xi})]^2\,.
\end{aligned} \tag{3.8}$$

On the other hand, letting $\alpha = 1$ and $\ell = 1$, corresponding to the uncorrelated standard APF, in Corollary 3.2 yields the asymptotic variance

$$\bar{\sigma}^2_{\alpha=\ell=1}[\Psi](f) \triangleq \sigma^2\{L[\cdot, f - \mu(f)]\}/[\nu L(\tilde{\Xi})]^2 + \beta\nu(\Psi)\nu(\Psi\mathbb{M}_{1,1}\{\cdot, \Psi^{-1}[f - \mu(f)]\})/[\nu L(\tilde{\Xi})]^2 ,$$

and, under the assumption that the inherent covariance structure of each block is *uniform* with $\mathbb{M}_{m,n} = \mathbb{M}^*$ for all $(m, n) \in \{1, \ldots, \alpha\}^2$ such that $m \neq n$, the criterion

$$\begin{aligned} \bar{\sigma}^2[\Psi](f) &\leq \bar{\sigma}^2_{\alpha=\ell=1}[\Psi](f) \\ &\Leftrightarrow \\ -\nu(\Psi\mathbb{C}^*\{\cdot, \Psi^{-1}[f - \mu(f)]\}) &\geq \nu(\Psi L^2\{\cdot, \Psi^{-1}[f - \mu(f)]\}) . \end{aligned} \tag{3.9}$$

Remark 3.3. *From the criterion* (3.9) *it is evident that mutating the particles in blocks without any (or positive) inherent correlation structure (that is, letting* $\mathbb{C}^* \geq 0$*) will, not surprisingly, increase the asymptotic variance vis-à-vis the standard APF. Moreover, since the correlation* $\mathbb{C}^* \geq 0$ *is a decreasing function of* α*, we conclude that there is a critical block size above which* (3.9) *will not hold even if the offspring particles of a block have the EA property conditionally on their ancestor.*

4 Application to state space models

In *state space models* a time series $Y \triangleq \{Y_n\}_{n=0}^{\infty}$, taking values in some state space $(\mathsf{Y}, \mathcal{B}(\mathsf{Y}))$, is modeled as noisy *observation* of an unobservable (possibly time-inhomogeneous) Markov chain $X \triangleq \{X_n\}_{n=0}^{\infty}$. The Markov chain, also referred to as the *state sequence*, is assumed to take values in some state space $(\mathsf{X}, \mathcal{B}(\mathsf{X}))$. In the examples discussed below we will exclusively let $\mathsf{X} \equiv \mathbb{R}$. The observed values are assumed to be conditionally independent given the latent process X in such a way that the distribution of Y_n depends on X_n only. For a model of this type, all inference about the hidden states has to be made through the observations only.

Denote by $\{Q_n\}_{n=0}^{\infty}$ and ν_0 the Markov transition kernel and initial distribution of the hidden chain, respectively. In addition, suppose that the conditional distribution of Y_n given X_n admits a density g_n on Y with respect to some reference measure η, that is,

$$\mathbb{P}(Y_n \in A | X_n) = \int_A g_n(X_n, y)\, \eta(\mathrm{d}y) , \quad A \in \mathcal{B}(\mathsf{Y}) .$$

This gives us a the following complete description of a state space model:

$$\begin{aligned} X_0 &\sim \nu_0 \ , \\ X_{n+1}|X_n &\sim Q_n(X_n, \cdot) \ , \\ Y_n|X_n &\sim g_n(X_n, \cdot) \ . \end{aligned}$$

In this setting, the *optimal filtering problem* consists in computing, recursively in time as new observations become available, the *filter* posterior distributions

$$\varphi_n(A) \triangleq \mathbb{P}(X_n \in A | Y_{0:n}) \ , \quad A \in \mathcal{B}(\mathsf{X}), \ n \geq 0 \ .$$

A straightforward application of Bayes' rule yields, for $A \in \mathcal{B}(\mathsf{X})$, the recursion

$$\begin{aligned} \varphi_0(A) &= \frac{\int_A g_0(x, Y_0) \ \nu_0(\mathrm{d}x)}{\int_{\mathsf{X}} g_0(x, Y_0) \ \nu_0(\mathrm{d}x)} \ , \\ \varphi_{n+1}(A) &= \frac{\int_{\mathsf{X}} \int_A g_{n+1}(x', Y_{n+1}) \ Q_n(x, \mathrm{d}x') \ \varphi_n(\mathrm{d}x)}{\int\int_{\mathsf{X}^2} g_{n+1}(x', Y_{n+1}) \ Q_n(x, \mathrm{d}x') \ \varphi_n(\mathrm{d}x)} \ , \end{aligned} \tag{4.1}$$

referred to as the *filtering recursion*. Since closed form solutions to the filtering recursion are obtainable only in the case of a linear/Gaussian model or when the state space X is finite, we apply the SMC methodology described in the previous; indeed, having defined, for $A \in \mathcal{B}(\mathsf{X})$ and $x \in \mathsf{X}$, the unnormalized transition kernels

$$L_n(x, A) = \int_A g_{n+1}(x', Y_{n+1}) \ Q_n(x, \mathrm{d}x') \ , \tag{4.2}$$

yielding the equivalent Feynman-Kac representation

$$\varphi_{n+1}(A) = \frac{\varphi_n L_n(A)}{\varphi_n L_n(\mathsf{X})} \ , \quad A \in \mathcal{B}(\mathsf{X}) \ ,$$

of (4.1), we conclude that the optimal filtering problem can be perfectly cast into the framework of Section 2 with $\Xi = \tilde{\Xi} = \mathsf{X}$, $\nu = \varphi_n$, $L = L_n$, and $\mu = \varphi_{n+1}$.

4.1 ARCH model

As a first example we consider the classical Gaussian *autoregressive conditional heteroscedasticity* (ARCH) *model* observed in noise (Bollerslev *et al.*, 1994) given by

$$\begin{aligned} X_{n+1} &= W_{n+1} \sqrt{\beta_0 + \beta_1 X_n^2} \ , \\ Y_n &= X_n + \sigma V_k \ . \end{aligned}$$

where $\{W_n\}_{n=1}^{\infty}$ and $\{V_n\}_{n=0}^{\infty}$ are mutually independent sequences of standard normal distributed variables such that W_n is independent of $\{(X_i, Y_i)\}_{i=0}^{n}$ and V_n is independent of $\{(X_i, Y_i)\}_{i=0}^{n-1}$ and X_n. In this case the optimal kernel $L_n(x, \cdot)/L_n(x, \mathsf{X})$, $x \in \mathbb{R}$, being the conditional distribution of the state X_{n+1} given $X_n = x$ and the observation Y_{n+1}, is Gaussian with mean $m_n(x)$ and variance $\hat{\sigma}_n^2(x)$, where

$$m_n(x) = \frac{\beta_0 + \beta_1 x^2}{\beta_0 + \beta_1 x^2 + \sigma^2} Y_{n+1} \ , \quad \hat{\sigma}_n^2(x) = \frac{\beta_0 + \beta_1 x^2}{\beta_0 + \beta_1 x^2 + \sigma^2} \sigma^2 \ .$$

Thus, the optimal adjustment multiplier weight function $\Psi_n(x) = L_n(x, \mathsf{X})$ can be expressed on closed form as

$$\Psi_n(x) = \mathcal{N}(Y_{n+1}; 0, \beta_0 + \beta_1 x^2 + \sigma^2) \tag{4.3}$$

where $\mathcal{N}(x; \mu, \sigma^2) \triangleq \exp[-(x-\mu)^2/(2\sigma^2)]/\sqrt{2\pi\sigma^2}$ denotes the univariate Gaussian density function, yielding exactly uniform importance weights $\tilde{\omega}_{N,i} \equiv 1$, $i \in \{1, \ldots, \alpha M_N\}$.

In this setting we used SMC to estimate posterior filter means $\varphi_n(\mathrm{I}_{\mathsf{X}})$, where I_{X} denotes the identity mapping $\mathrm{I}_{\mathsf{X}}(x) = x$ on X. Initially, to form an idea of the effect of the antithetic coupling we compared the auxiliary particle filter in Algorithm 2, using $\alpha \in \{2, 3\}$ conditionally independent offspring of each particle $\xi_{N,i}$, $i \in \{1, \ldots, M_N\}$, in the mutation step, to the filter in Algorithm 3 using equally many antithetically coupled offspring. In the case $\alpha = 2$ we used the standard coupling

$$\begin{aligned} \tilde{\xi}_{N,\alpha(i-1)+1}^{(n+1)} &= m_n(\xi_{N,i}^{(n)}) + \hat{\sigma}_n(\xi_{N,i}^{(n)}) \varepsilon_i^{(n)} \ , \\ \tilde{\xi}_{N,\alpha(i-1)+2}^{(n+1)} &= 2m_n(\xi_{N,i}^{(n)}) - \tilde{\xi}_{N,\alpha(i-1)+1}^{(n+1)} \ , \end{aligned} \tag{4.4}$$

where $\{\varepsilon_i^{(n)}\}_{i=1}^{M_N}$ is a sequence of mutually independent standard normal distributed random variables being independent of everything else. This coupling yields largest possible negative correlation (that is, is EA) conditionally on $\xi_{N,i}^{(n)}$, *i.e.* $\mathrm{Corr}(\tilde{\xi}_{N,\alpha(i-1)+1}^{(n+1)}, \tilde{\xi}_{N,\alpha(i-1)+2}^{(n+1)} | \xi_{N,i}^{(n)}) = -1$, and in the kernel language of Section 2 it holds that $R_1(\xi, A) = \int_A \mathcal{N}(\tilde{\xi}; m_n(\xi), \hat{\sigma}_n^2(\xi)) \, \mathrm{d}\tilde{\xi}$ and $R_2(\xi, \tilde{\xi}_1, A) = \delta_{2m_n(\xi) - \tilde{\xi}_1}(A)$ for Borel sets A. A similar technique was used in the case $\alpha = 3$;

here we set

$$\begin{aligned}
\widetilde{\xi}^{(n+1)}_{N,\alpha(i-1)+1} &= m_n(\xi^{(n)}_{N,i}) + \hat{\sigma}_n(\xi^{(n)}_{N,i})\varepsilon^{(n)}_{i,1} \,, \\
\widetilde{\xi}^{(n+1)}_{N,\alpha(i-1)+2} &= \frac{1}{2}\left(3m_n(\xi^{(n)}_{N,i}) - \widetilde{\xi}^{(n+1)}_{N,\alpha(i-1)+1} + \sqrt{3}\hat{\sigma}_n(\xi^{(n)}_{N,i})\varepsilon^{(n)}_{i,2}\right) \,, \\
\widetilde{\xi}^{(n+1)}_{N,\alpha(i-1)+3} &= 3m_n(\xi^{(n)}_{N,i}) - \widetilde{\xi}^{(n+1)}_{N,\alpha(i-1)+1} - \widetilde{\xi}^{(n+1)}_{N,\alpha(i-1)+2} \,,
\end{aligned} \tag{4.5}$$

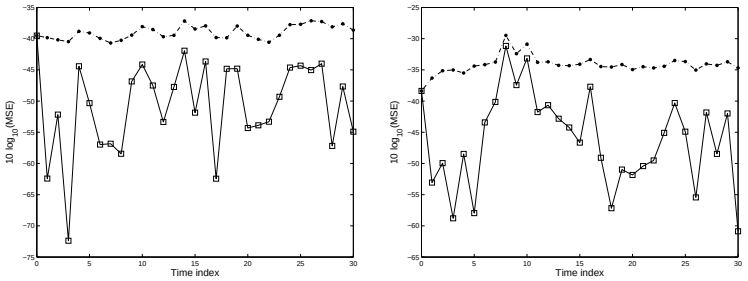
where the independent sequences $\{\varepsilon^{(n)}_{i,1}\}_{i=1}^{M_N}$ and $\{\varepsilon^{(n)}_{i,2}\}_{i=1}^{M_N}$ are as above. The coupling (4.5) yields the conditional correlation

$$\mathrm{Corr}(\widetilde{\xi}^{(n+1)}_{N,\alpha(i-1)+m}, \widetilde{\xi}^{(n+1)}_{N,\alpha(i-1)+m'} | \xi^{(n)}_{N,i}) = -1/2,$$

for $(m, m') \in \{1, 2, 3\}$ and $m \neq m'$.

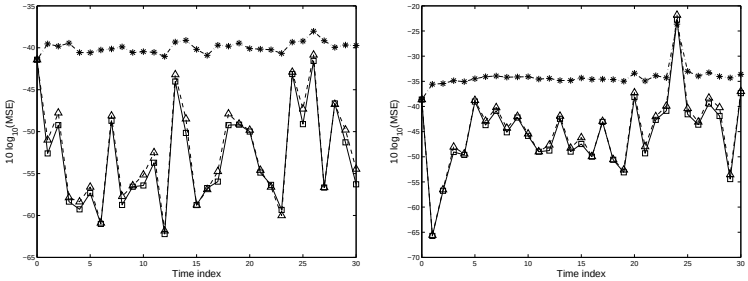
The comparison was done for two different data sets obtained by simulation of ARCH models parameterized by $(\beta_0, \beta_1, \sigma) = (0.9, 0.6, 1)$ and $(\beta_0, \beta_1, \sigma) = (0.9, 0.6, 10)$, corresponding to informative and non-informative observations, respectively. The mean squared errors (MSEs) for 400 runs of each filter with $M_N = 6{,}000/\alpha$ are, for the different values of α, displayed in Figure 1(a) (the informative case) and Figure 1(b) (the non-informative case). The MSEs are based on reference posterior filter mean values obtained by means of the standard APF (for which $\alpha = \ell = 1$) using as many as 500,000 particles. From both figures it is evident that letting the particles of a block be antithetically coupled instead of conditionally independent decreases the variance significantly. Moreover, the improvement is especially noticeable in the informative case.

More relevant is to compare the performance of Algorithm 3, again with $\alpha \in \{2, 3\}$ and $M_N = 6{,}000/\alpha$, to that of the standard fully adapted APF using 6,000 particles without any block structure. In this setting, both antithetic filters are clearly more computationally efficient since, firstly, only a half and a third of the particles are selected at each resampling operation, and, secondly, a half and a third of the random moves at each mutation step are replaced by simple assignments (matrix manipulations) in the two cases $\alpha = 2$ and $\alpha = 3$, respectively. The outcome is displayed in Figure 2(a) (the informative case) and Figure 2(b) (the non-informative case), from which it is clear that performances of the antithetic filters are, despite being less costly, superior, especially in the case of informative observations; indeed, the improvement is over 20 Decibel at some time steps. Moreover, it is evident that the computational gain of using g $\alpha = 3$ instead of $\alpha = 2$ offspring in each block is at the expense of a slight decrease of precision.



(a) (b)

Figure 1: Plot of MSEs (in Decibel) of filters being implementations of Algorithm 3 with $\alpha = 2$ antithetically coupled ($\square$) and conditionally independent ($\bullet$) offspring for the ARCH model with informative (a) and non-informative (b) observations. The MSE values are based on 400 runs of each algorithm with $M_N = 3{,}000$.



(a) (b)

Figure 2: Plot of MSEs (in Decibel) of the standard optimal APF ($*$) with 6,000 particles and antithetic filters with $\alpha = 2$ ($\square$) and $\alpha = 3$ ($\triangle$) for the ARCH model with informative (a) and non-informative (b) observations. $\alpha M_N = 6,000$ for both antithetic filters and the MSE values are based on 400 runs of each algorithm.

4.2 Growth model

The univariate *growth model* given by, for $n \geq 0$,

$$\begin{aligned} X_{n+1} &= a_n(X_n) + \sigma_w W_{n+1} \,, \\ Y_n &= bX_n^2 + \sigma_v V_n \,, \end{aligned} \tag{4.6}$$

where

$$a_n(x) = \alpha_0 x + \alpha_1 \frac{x}{1+x^2} + \alpha_2 \cos(1.2n) \,, \quad x \in \mathbb{R} \,,$$

and the sequences $\{W_n\}_{n=1}^{\infty}$ and $\{V_n\}_{n=0}^{\infty}$ are as in the previous example, was discussed by Kitagawa (1987) (see also Polson *et al.*, 2002) and has served as a benchmark for state space filtering techniques during the last decades. We will follow the lines of Cappé *et al.* (2005) and consider the parameter vector $(\alpha_0, \alpha_1, \alpha_2, b, \sigma_v^2) = (0.5, 25, 8, 0.05, 1)$ and $\sigma_w^2 \in \{1, 10\}$, the values of the latter parameter corresponding to non-informative and informative observations, respectively. The initial state is set deterministically to $X_0 = 0.1$. For a given observation Y_n in $\mathbb{R}$, the local likelihood for the state at time n is given by the function

$$x \in \mathbb{R} \mapsto g(x, Y_n) = \mathcal{N}(Y_n; bx_n^2, \sigma_v^2) \in \mathbb{R}^+ \,. \tag{4.7}$$

which is symmetric about zero for any observation Y_n. Interestingly, functions (4.7) associated with negative observations $Y_n \leq 0$ are *unimodal*, while those associated with positive observations $Y_n > 0$ are *bimodal* with modes located at $\pm\sqrt{Y_n/b}$. This bimodality is challenging from a filtering point of view and puts heavy demands on the applied SMC method.

Unlike the ARCH model in the previous section, direct simulation from the optimal kernel is infeasible in this case since the measurement equation (4.6) is nonlinear in the state. Thus, in order to mimic efficiently the optimal kernel and adjustment multiplier weights we approximate the local likelihood (4.7) by a mixture

$$\mathcal{G}(x, Y_n) \triangleq \mathcal{N}(x; \mu_1(Y_n), \varsigma^2(Y_n))/2 + \mathcal{N}(x; \mu_2(Y_n), \varsigma^2(Y_n))/2$$

of two Gaussian densities, where

$$(\mu_1(Y_n), \mu_2(Y_n), \varsigma^2(Y_n)) \triangleq \begin{cases} (0, 0, -\sigma_v^2/(2bY_n)) & \text{for } Y_n \leq 0 \,, \\ (-\sqrt{Y_n/b}, \sqrt{Y_n/b}, \sigma_v^2/(4bY_n)) & \text{for } Y_n > 0 \,. \end{cases}$$

Consequently, we let the means and standard deviations of the two strata be the locations (which coincide when $Y_n \leq 0$) and (common) inverted negated log curvature of the modes of the local likelihood, respectively; more specifically, $\varsigma^2(Y_n) = -1/(\mathrm{d}^2 \log g(x, Y_n)/\mathrm{d}x^2)|_{x=\mu_1(Y_n)}$. From now on we omit for brevity the dependence on the observation from the notation of the quantities above and write $(\mu_1, \mu_2, \varsigma^2)$ instead of $(\mu_1(Y_n), \mu_2(Y_n), \varsigma^2(Y_n))$. Plugging the approximation $\mathcal{G}$ into the expression (4.2) of the unnormalized optimal kernel yields straightforwardly the mixture

$$\mathcal{L}_n(x, A) \triangleq \int_A \mathcal{G}(x', Y_{n+1})\, Q_n(x, \mathrm{d}x') = \beta_n^{(1)}(x) G_n^{(1)}(x, A) + \beta_n^{(2)}(x) G_n^{(2)}(x, A) \ ,$$

for $x \in \mathsf{X}$, $A \in \mathcal{B}(\mathsf{X})$, where each Gaussian stratum

$$G_n^{(d)}(x, A) \triangleq \int_A \mathcal{N}(x'; \tau_n^{(d)}(x), \eta_n^2)\, \mathrm{d}x' \ , \quad d \in \{1, 2\} \ ,$$

with means and variance (recall that μ_d, $d \in \{1, 2\}$, and ς^2 depend on Y_{n+1})

$$\tau_n^{(d)}(x) \triangleq \frac{\sigma_w^2 \mu_d + \varsigma^2 a_n(x)}{\sigma_w^2 + \varsigma^2} \ ,$$
$$\eta_n^2 \triangleq \frac{\sigma_w^2 \varsigma^2}{\sigma_w^2 + \varsigma^2} \ ,$$

is weighted by

$$\beta_n^{(d)}(x) \triangleq \mathcal{N}(\mu_d; a_n(x), \sigma_w^2 + \varsigma^2) \ , \quad d \in \{1, 2\} \ .$$

By normalizing we obtain the approximation

$$\mathcal{L}_n(x, A)/\mathcal{L}_n(x, \mathsf{X}) = \bar{\beta}_n(x) G_n^{(1)}(x, A) + (1 - \bar{\beta}_n(x)) G_n^{(2)}(x, A) \ , \tag{4.8}$$

for $x \in \mathsf{X}$, $A \in \mathcal{B}(\mathsf{X})$, of the optimal kernel, where we have defined the normalized weight

$$\bar{\beta}_n(x) \triangleq \frac{\beta_n^{(1)}(x)}{\beta_n^{(1)}(x) + \beta_n^{(2)}(x)} \ , \quad x \in \mathsf{X} \ .$$

Moreover, in this setting the approximate optimal adjustment multiplier weights are given by

$$\Psi_n(x) = \mathcal{L}_n(x, \mathsf{X}) = \beta_n^{(1)}(x) + \beta_n^{(2)}(x) \ , \quad x \in \mathsf{X} \ .$$

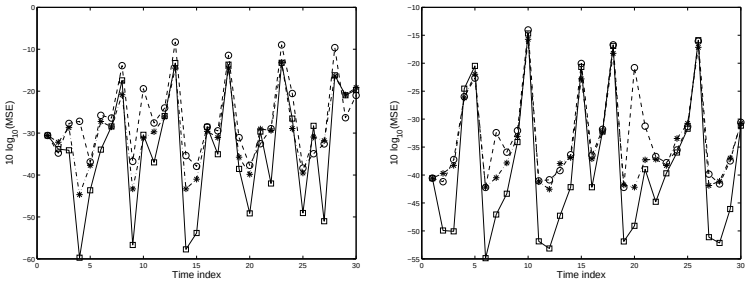
Using (4.8) as proposal, the experiment of the previous example was repeated with focus set on the case $\alpha = 2$. In order to impose a conditionally negative correlation structure we let each pair of offspring particles evolve according to

$$\begin{aligned}\widetilde{\xi}_{N,\alpha(i-1)+1}^{(n+1)} &= \tau_n^{(1)}(\xi_{N,i}^{(n)})\mathbb{1}_{\{U_i^{(n)}<\bar{\beta}_n(\xi_{N,i}^{(n)})\}} + \tau_n^{(2)}(\xi_{N,i}^{(n)})\mathbb{1}_{\{U_i^{(n)}\geq\bar{\beta}_n(\xi_{N,i}^{(n)})\}} + \eta_n\varepsilon_i^{(n)}\,,\\ \widetilde{\xi}_{N,\alpha(i-1)+2}^{(n+1)} &= \tau_n^{(1)}(\xi_{N,i}^{(n)})\mathbb{1}_{\{1-U_i^{(n)}<\bar{\beta}_n(\xi_{N,i}^{(n)})\}} + \tau_n^{(2)}(\xi_{N,i}^{(n)})\mathbb{1}_{\{1-U_i^{(n)}\geq\bar{\beta}_n(\xi_{N,i}^{(n)})\}} - \eta_n\varepsilon_i^{(n)},\end{aligned} \tag{4.9}$$

where $\{U_i^{(n)}\}_{i=1}^{M_N}$ and $\{\varepsilon_i^{(n)}\}_{i=1}^{M_N}$ are independent sequences of mutually independent uniformly (on $[0,1]$) and standard normal distributed random variables, respectively, such that each pair $(U_i^{(n)}, \varepsilon_i^{(n)})$ is independent of everything else. It is easily established that each of the offspring particles $\widetilde{\xi}_{N,\alpha(i-1)+1}^{(n+1)}$ and $\widetilde{\xi}_{N,\alpha(i-1)+2}^{(n+1)}$ of the coupling (4.9) is marginally distributed according to the approximate optimal kernel (4.8). In addition, one can show that (see Section A.3 for details) the correlation between the offspring of a block is given by, for $\xi \in \mathsf{X}$,

$$\begin{aligned}&\mathrm{Corr}\left[\widetilde{\xi}_{N,\alpha(i-1)+1}^{(n+1)}, \widetilde{\xi}_{N,\alpha(i-1)+2}^{(n+1)} \,\middle|\, \xi_{N,i}^{(n)} = \xi\right]\\ &\quad= -\left[\tau_n^{(1)}(\xi) - \tau_n^{(2)}(\xi))^2\bar{\beta}_n(\xi)(1-\bar{\beta}_n(\xi)) + \eta_n^2\right]^{(-1)}\\ &\qquad\times\left(\left[\tau_n^{(1)}(\xi) - \tau_n^{(2)}(\xi)\right]^2\left[\bar{\beta}_n^2(\xi)\mathbb{1}_{\{\bar{\beta}_n(\xi)\leq 1/2\}}\right.\right.\\ &\qquad\qquad\left.\left.+\left(\bar{\beta}_n^2(\xi)-1\right)^2\mathbb{1}_{\{\bar{\beta}_n(\xi)>1/2\}}\right] + \eta_n^2\right)\end{aligned} \tag{4.10}$$

which is *always* negative and simplifies to -1 in the unimodal case (as $\tau_n^{(1)}(\xi) = \tau_n^{(2)}(\xi)$ for all $\xi \in \mathsf{X}$ when $Y_{n+1} < 0$). Figure 3 displays MSE (in Decibel) comparisons between the antithetic APF with $\alpha = 2$ and $\alpha M_N = 5{,}000$, a (close to) fully adapted APF, based on the proposal kernel (4.8) and 5,000 particles, and the plain bootstrap filter using 5,000 particles. Like in the ARCH example, we let the filters approximate filter posterior means $\varphi_n(\mathrm{I}_{\mathsf{X}})$ for observation records of length 30, and since the initial value is known deterministically the log MSE is null at time zero. The comparison was made for informative ($\sigma_w^2 = 10$, Figure 3(a)) as well as non-informative ($\sigma_w^2 = 1$, Figure 3(b)) observations and the MSEs, measured with respect to reference values obtained with the fully adapted APF using 500,000 particles, were based on 400 runs of each algorithm. Also for



(a) (b)

Figure 3: Plot of MSEs (in Decibel) of the plain bootstrap filter (○) using 5,000 particles, the standard optimal APF (∗) using 5,000 particles, and the antithetic filter (□) with $\alpha = 2$ and $\alpha M_N = 5{,}000$ for the growth model with informative (a) and non-informative (b) observations. The MSE values are based on 400 runs of each algorithm.

this demanding model the variance reduction introduced by the antithetic coupling is significant; indeed, despite being clearly less computationally costly (see the discussion in the previous example), the antithetic filter improves the MSE performances of the APF and the bootstrap filter by more than 10 Decibels at several time points for both observation records. Moreover, from the figures it is evident that proposing particles according to the approximate optimal kernel (4.8) instead of the prior kernel yields, as we may expect, generally more precise posterior filter mean estimates, since the APF outperforms the bootstrap particle filter at most time steps.

A Proofs

A.1 Proof of Theorem 3.1

The result follows straightforwardly from Slutsky's theorem and results obtained by Douc and Moulines (2005) in the case of independently mutated particles. Indeed, by (Douc and Moulines, 2005, Equation (36)) we have, for any $1 \leq k \leq \alpha$,

$$\Omega_N^{-1} \sum_{j=1}^{M_N} \tilde{\omega}_{N,\alpha(j-1)+k} f(\tilde{\xi}_{N,\alpha(j-1)+k}) \xrightarrow{\mathbb{P}} \nu L(f) \ ,$$

yielding immediately

$$(\alpha\Omega_N)^{-1}\sum_{i=1}^{\alpha M_N}\tilde{\omega}_{N,i}f(\tilde{\xi}_{N,i}) = \alpha^{-1}\sum_{k=1}^{\alpha}\Omega_N^{-1}\sum_{j=1}^{M_N}\tilde{\omega}_{N,\alpha(j-1)+k}f(\tilde{\xi}_{N,\alpha(j-1)+k}) \xrightarrow{\mathbb{P}} \nu L(f)\ . \tag{A.1}$$

By applying (A.1) for this limit for $f \equiv 1$ (recall that $L(\cdot,\tilde{\Xi}) \in \mathsf{C}$ by assumption, implying that the constant function belongs to $\tilde{\mathsf{C}}$) we obtain, using again Slutsky's theorem,

$$\tilde{\Omega}_N^{-1}\sum_{i=1}^{\alpha M_N}\tilde{\omega}_{N,i}f(\tilde{\xi}_{N,i}) \xrightarrow{\mathbb{P}} \nu L(f)/\nu L(\tilde{\Xi}) = \mu(f)\ .$$

To prove the second property in Definition 3.1, write

$$(\alpha\Omega_N)^{-1}\max_{1\le i\le \alpha M_N}\tilde{\omega}_{N,i} \le \alpha^{-1}\sum_{k=1}^{\alpha}\tilde{\Omega}_N^{-1}\max_{1\le j\le M_N}\tilde{\omega}_{N,\alpha(j-1)+k}\ ; \tag{A.2}$$

however, by inspecting the proof of (Douc and Moulines, 2005, Theorem 1) we conclude that each term on the RHS of (A.2) tends to zero in probability, which in combination with (A.1) implies that

$$\tilde{\Omega}_N^{-1}\max_{1\le i\le \alpha M_N}\tilde{\omega}_{N,i} = (\alpha\Omega_N/\tilde{\Omega}_N)(\alpha\Omega_N)^{-1}\max_{1\le i\le \alpha M_N}\tilde{\omega}_{N,i} \xrightarrow{\mathbb{P}} 0\ .$$

This completes the proof.

A.2 Proof of Theorem 3.2

Let $f \in \tilde{\mathsf{A}}$ and assume without loss of generality that $\mu(f) = 0$. Then write, following the lines of the proof of (Douc and Moulines, 2005, Theorem 2),

$$a_N\tilde{\Omega}_N^{-1}\sum_{i=1}^{\alpha M_N}\tilde{\omega}_{N,i}f(\tilde{\xi}_{N,i}) = \alpha\Omega_N\tilde{\Omega}_N^{-1}(A_N + B_N)\ , \tag{A.3}$$

where

$$A_N \triangleq \sum_{j=1}^{M_N}\mathbb{E}\left[U_{N,j}\middle|\mathcal{F}_{N,\alpha(j-1)}\right]\ , \quad B_N \triangleq \sum_{j=1}^{M_N}\left\{U_{N,j} - \mathbb{E}\left[U_{N,j}\middle|\mathcal{F}_{N,\alpha(j-1)}\right]\right\}\ ,$$

and $U_{N,j} \triangleq a_N(\alpha\Omega_N)^{-1}\sum_{k=1}^{\alpha}\tilde{\omega}_{N,\alpha(j-1)+k}f(\tilde{\xi}_{N,\alpha(j-1)+k})$.
Since, by (A.1), $\tilde{\Omega}_N/(\alpha\Omega_N) \xrightarrow{\mathbb{P}} \nu L(\tilde{\Xi})$, as $N \to \infty$, it is enough to prove that

$$A_N + B_N \xrightarrow{\mathcal{D}} \mathcal{N}\{0, \sigma^2[L(\cdot,f)] + \eta^2(f)\} ,$$

where

$$\eta^2(f) \triangleq \alpha^{-2} \sum_{(m,n)\in\{1,\ldots,\alpha\}^2} \gamma\mathbb{C}_{m,n}(f) .$$

For A_N it holds, since the weighted sample $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ is AN for $(\mu, \mathsf{A}, \mathsf{W}, \sigma, \gamma, \{a_N\}_{N=1}^{\infty})$ by assumption and $L(\cdot,f) \in \mathsf{A}$, that

$$\begin{aligned} A_N &= a_N(\alpha\Omega_N)^{-1}\sum_{j=1}^{M_N}\sum_{k=1}^{\alpha}\mathbb{E}\left[\tilde{\omega}_{N,\alpha(j-1)+k}f(\tilde{\xi}_{N,\alpha(j-1)+k})\middle|\mathcal{F}_{N,\alpha(j-1)}\right] \\ &= a_N\Omega_N^{-1}\sum_{j=1}^{M_N}\omega_{N,j}L(\xi_{N,j},f) \xrightarrow{\mathcal{D}} \mathcal{N}\{0,\sigma^2[L(\cdot,f)]\} . \end{aligned}$$

We now consider B_N and establish that, for any $u \in \mathbb{R}$,

$$\mathbb{E}\left[\exp(\mathrm{i}uB_N)|\,\mathcal{F}_{N,0}\right] \xrightarrow{\mathbb{P}} \exp(-u^2\eta^2(f)/2) \tag{A.4}$$

from which the result of the theorem follows. The proof of (A.4) consists in showing that the two conditions of Theorem 13 in (Douc and Moulines, 2005) are satisfied for the triangular array $\{(U_{N,j}, \mathcal{F}_{N,\alpha j})\}_{j=1}^{M_N}$.

For establishing condition *i)* of the theorem in question, write

$$\begin{aligned} &\mathbb{E}\left[U_{N,j}^2\middle|\,\mathcal{F}_{N,\alpha(j-1)}\right] = a_N^2(\alpha\Omega_N)^{-2} \\ &\times \sum_{(k,m)\in\{1,\ldots,\alpha\}^2}\mathbb{E}\left[\tilde{\omega}_{N,\alpha(j-1)+k}f(\tilde{\xi}_{N,\alpha(j-1)+k})\tilde{\omega}_{N,\alpha(j-1)+m}f(\tilde{\xi}_{N,\alpha(j-1)+m})\,\middle|\mathcal{F}_{N,\alpha(j-1)}\right] \\ &= a_N^2(\alpha\Omega_N)^{-2}\sum_{j=1}^{M_N}\omega_{N,j}^2\sum_{(k,m)\in\{1,\ldots,\alpha\}^2}\mathbb{M}_{k,m}(\xi_{N,j},f) . \end{aligned} \tag{A.5}$$

However, for all $(k, m) \in \{1, \ldots, \alpha\}^2$,

$$\mathbb{M}_{k,m}(\cdot, f) \leq \mathcal{R}_{0,k}(\cdot, \Phi_k^2 f^2) + \mathcal{R}_{0,m}(\cdot, \Phi_m^2 f^2) \in \mathsf{W};$$

since W is proper, this implies (under **(A2)**) the limit

$$\begin{aligned} a_N^2 (\alpha \Omega_N)^{-2} \sum_{j=1}^{M_N} \omega_{N,j}^2 \sum_{(k,m) \in \{1,\ldots,\alpha\}^2} & \mathbb{M}_{k,m}(\xi_{N,j}, f) \\ & \xrightarrow{\mathbb{P}} \alpha^{-2} \sum_{(k,m) \in \{1,\ldots,\alpha\}^2} \gamma \mathbb{M}_{k,m}(f) \ . \end{aligned} \tag{A.6}$$

Now consider

$$\begin{aligned} & \sum_{j=1}^{M_N} \mathbb{E}^2 \left[U_{N,j} \middle| \mathcal{F}_{N,\alpha(j-1)} \right] \\ & \quad = a_N^2 (\alpha \Omega_N)^{-2} \\ & \quad \times \sum_{j=1}^{M_N} \omega_{N,j}^2 \mathbb{E}^2 \left[\sum_{k=1}^{\alpha} \Phi_k(\xi_{N,j}, \tilde{\xi}_{N,\alpha(j-1)+k}) f(\tilde{\xi}_{N,\alpha(j-1)+k}) \middle| \mathcal{F}_{N,\alpha(j-1)} \right] \\ & \quad = a_N^2 \Omega_N^{-2} \sum_{j=1}^{M_N} \omega_{N,j}^2 L^2(\xi_{N,j}, f) \end{aligned} \tag{A.7}$$

Here, for any $k \in \{1, \ldots, \alpha\}$,

$$L^2(\cdot, f) = \mathcal{R}_{0,k}^2(\cdot, \Phi_k f) \leq \mathcal{R}_{0,k}(\cdot, \Phi_k^2 f^2) \in \mathsf{W},$$

and reusing the asymptotic normality of $\{(\xi_{N,i}, \omega_{N,i})\}_{i=1}^{M_N}$ yields

$$a_N^2 \Omega_N^{-2} \sum_{j=1}^{M_N} \omega_{N,j}^2 L^2(\xi_{N,j}, f) \xrightarrow{\mathbb{P}} \gamma L^2(f) \ . \tag{A.8}$$

Finally, by combining Equations (A.5)–(A.8) we conclude that

$$\begin{aligned} \sum_{j=1}^{M_N} \left\{ \mathbb{E} \left[U_{N,j}^2 \middle| \mathcal{F}_{N,\alpha(j-1)} \right] - \mathbb{E}^2 \left[U_{N,j} \middle| \mathcal{F}_{N,\alpha(j-1)} \right] \right\} & \\ \xrightarrow{\mathbb{P}} \alpha^{-2} \sum_{(k,m) \in \{1,\ldots,\alpha\}^2} \gamma \mathbb{M}_{k,m}(f) - \gamma L^2(f) = \eta^2(f) \ , & \end{aligned}$$

which establishes condition *i)*.

It remains to check condition *ii)*, that is, for any $\varepsilon > 0$,

$$C_N \triangleq \sum_{j=1}^{M_N} \mathbb{E}\left[U_{N,j}\mathbb{1}_{\{|U_{N,j}|\geq\varepsilon\}}\left|\mathcal{F}_{N,\alpha(j-1)}\right.\right] \xrightarrow{\mathbb{P}} 0\ .$$

Thus, argue along the lines of the proof of (Douc and Moulines, 2005, Theorem 2) and write, for any $C > 0$,

$$\begin{aligned} C_N \leq a_N^2(\alpha\Omega_N)^{-2}\sum_{j=1}^{M_N}\omega_{N,j}^2 \sum_{(k,m)\in\{1,\ldots,\alpha\}^2} \mathbb{M}_{k,m}\left(\xi_{N,j}, f\mathbb{1}_{\{|\sum_{k=1}^{\alpha}\Phi_k f|\geq C\}}\right) \\ + \mathbb{1}_{\{a_N(\alpha\Omega_N)^{-1}\max_i\omega_{N,i}\geq\varepsilon C^{-1}\}}\sum_{j=1}^{M_N}\mathbb{E}\left[U_{N,j}^2\left|\mathcal{F}_{N,\alpha(j-1)}\right.\right]. \quad \text{(A.9)} \end{aligned}$$

Under **(A2)** the indicator function of the second term on the RHS of (A.9) tends to zero in probability and since, for all $(k,m)\in\{1,\ldots,\alpha\}^2$,

$$\mathbb{M}_{k,m}(\cdot, f\mathbb{1}_{\{|\sum_{k=1}^{\alpha}\Phi_k f|\geq C\}}) \leq \mathcal{R}_{0,k}(\cdot,\Phi_k^2 f^2) + \mathcal{R}_{0,m}(\cdot,\Phi_m^2 f^2) \in \mathsf{W}$$

we obtain

$$\begin{aligned} a_N^2(\alpha\Omega_N)^{-2}\sum_{j=1}^{M_N}\omega_{N,j}^2 \sum_{(k,m)\in\{1,\ldots,\alpha\}^2} \mathbb{M}_{k,m}\left(\xi_{N,j}, f\mathbb{1}_{\{|\sum_{k=1}^{\alpha}\Phi_k f|\geq C\}}\right) \\ \xrightarrow{\mathbb{P}} \alpha^{-2}\sum_{(k,m)\in\{1,\ldots,\alpha\}^2}\gamma\mathbb{M}_{k,m}\left(f\mathbb{1}_{\{|\sum_{k=1}^{\alpha}\Phi_k f|\geq C\}}\right). \quad \text{(A.10)} \end{aligned}$$

By dominated convergence, the RHS of (A.10) can be made arbitrarily small by taking C sufficiently large. Therefore, also condition *ii)* of (Douc and Moulines, 2005, Theorem 13) is satisfied, implying the convergence (A.4). This establishes (A.3).

We turn to the second property of Definition 3.2 and show that, for any $f\in\tilde{\mathsf{W}}$,

$$a_N^2\tilde{\Omega}_N^{-2}\sum_{i=1}^{\alpha M_N}\tilde{\omega}_{N,i}^2 f(\tilde{\xi}_{N,i}) \xrightarrow{\mathbb{P}} \tilde{\gamma}(f)\ . \quad \text{(A.11)}$$

However, since

$$\mathcal{R}_{0,k}(\cdot, \Phi_k^2 f) \leq \mathbb{1}_{\{\cdot : |f(\cdot)|>1\}} \mathcal{R}_{0,k}(\cdot, \Phi_k^2 f^2) + \mathbb{1}_{\{\cdot : |f(\cdot)|\leq 1\}} \mathcal{R}_{0,k}(\cdot, \Phi_k^2) \in \mathsf{W},$$

a direct application of (Douc and Moulines, 2005, Equation 39) yields that, for any $k \in \{1, \ldots, \alpha\}$,

$$a_N^2 \Omega_N^{-2} \sum_{j=1}^{M_N} \tilde{\omega}_{N,\alpha(j-1)+k}^2 f(\tilde{\xi}_{N,\alpha(j-1)+k}) \xrightarrow{\mathbb{P}} \gamma \mathcal{R}_{0,k}(\Phi_k^2 f) \ .$$

Combining (A.11) with the limit $\tilde{\Omega}_N/(\alpha\Omega_N) \xrightarrow{\mathbb{P}} \nu L(\tilde{\Xi})$ (see (A.1)) we obtain, using Slutsky's theorem,

$$\begin{aligned}
&a_N^2 \tilde{\Omega}_N^{-2} \sum_{i=1}^{\alpha M_N} \tilde{\omega}_{N,i}^2 f(\tilde{\xi}_{N,i}) \\
&= (\alpha\Omega_N/\tilde{\Omega}_N)^2 \alpha^{-2} \sum_{k=1}^{\alpha} a_N^2 \Omega_N^{-2} \sum_{j=1}^{M_N} \tilde{\omega}_{N,\alpha(j-1)+k}^2 f(\tilde{\xi}_{N,\alpha(j-1)+k}) \\
&\xrightarrow{\mathbb{P}} \alpha^{-2} \sum_{k=1}^{\alpha} \gamma \mathcal{R}_{0,k}(\Phi_k^2 f)/[\nu L(\tilde{\Xi})]^2 = \tilde{\gamma}(f) \ .
\end{aligned}$$

Finally, we establish the last property of Definition 3.2, that is,

$$a_N \tilde{\Omega}_N^{-1} \max_{1\leq i\leq \alpha M_N} \tilde{\omega}_{N,i} \xrightarrow{\mathbb{P}} 0 \ . \tag{A.12}$$

However, since, as shown by Douc and Moulines (2005, p. 30), for any $k \in \{1, \ldots, \alpha\}$,

$$a_N^2 (\alpha\Omega_N)^{-2} \max_{1\leq j\leq M_N} \tilde{\omega}_{N,\alpha(j-1)+k}^2 \xrightarrow{\mathbb{P}} 0 \ ,$$

we immediately obtain

$$a_N^2 \tilde{\Omega}_N^{-2} \max_{1\leq i\leq \alpha M_N} \tilde{\omega}_{N,i}^2 \leq (\alpha\Omega_N/\tilde{\Omega}_N)^2 \sum_{k=1}^{\alpha} a_N^2 (\alpha\Omega_N)^{-2} \max_{1\leq j\leq M_N} \tilde{\omega}_{N,\alpha(j-1)+k}^2 \xrightarrow{\mathbb{P}} 0 \ ,$$

from which (A.12) follows.

It remains to show that the sets $\tilde{\mathsf{A}}$ and $\tilde{\mathsf{W}}$ are proper. Since, by assumption, $L(\cdot, \tilde{\Xi}) \in \mathsf{A}$ and $\mathcal{R}_{0,k}(\cdot, \Phi_k^2) \in \mathsf{W}$, $k \in \{1, \dots, \alpha\}$, we conclude immediately that all constant functions $f \equiv c$ belong to $\tilde{\mathsf{A}}$. Now, let $|f| \leq |g|$, where g belongs to $\tilde{\mathsf{A}}$. Then $L(\cdot, |f|) \leq L(\cdot, |g|) \in \mathsf{A}$ and $\mathcal{R}_{0,k}(\cdot, \Phi_k^2 f^2) \leq \mathcal{R}_{0,k}(\cdot, \Phi_k^2 g^2) \in \mathsf{W}$, $k \in \{1, \dots, \alpha\}$, implying, by property **ii)** in the definition of a proper set, that $f \in \tilde{\mathsf{A}}$. Finally, let f and g be any two functions in $\tilde{\mathsf{A}}$. Then, for any constants $(a, b) \in \mathbb{R}^2$, $L(\cdot, |af + bg|) \leq |a| L(\cdot, |f|) + |b| L(\cdot, |g|) \in \mathsf{A}$; moreover, for all $k \in \{1, \dots, \alpha\}$,

$$\mathcal{R}_{0,k}(\cdot, \Phi_k^2 [af + bg]^2) \leq (a^2 + |a|) \mathcal{R}_{0,k}(\cdot, \Phi_k^2 f^2) + (b^2 + |b|) \mathcal{R}_{0,k}(\cdot, \Phi_k^2 g^2) \in \mathsf{W} \,,$$

implying that $af + bg \in \tilde{\mathsf{A}}$. The properness of $\tilde{\mathsf{W}}$ is established in a similar manner. This completes the proof.

A.3 Proof of (4.10)

Since $U_i^{(n)}$ and $\varepsilon_i^{(n)}$ are independent, it holds that

$$\begin{aligned}
&\operatorname{Cov}\left(\tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+1}, \tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+2} \,\middle|\, \xi^{(n)}_{N,i} = \xi \right) \\
&= [(\tau_n^{(1)}(\xi))^2 + (\tau_n^{(2)}(\xi))^2] \operatorname{Cov}\left(\mathbb{1}_{\{U_i^{(n)} < \bar{\beta}_n(\xi)\}}, \mathbb{1}_{\{1 - U_i^{(n)} < \bar{\beta}_n(\xi)\}} \,\middle|\, \xi^{(n)}_{N,i} = \xi \right) \\
&\quad + 2\tau_n^{(1)}(\xi)\tau_n^{(2)}(\xi) \operatorname{Cov}\left(\mathbb{1}_{\{U_i^{(n)} < \bar{\beta}_n(\xi)\}}, \mathbb{1}_{\{1 - U_i^{(n)} \geq \bar{\beta}_n(\xi)\}} \,\middle|\, \xi^{(n)}_{N,i} = \xi \right) - \eta_n^2 \,.
\end{aligned} \tag{A.13}$$

In addition, as $U_i^{(n)}$ is independent of $\xi^{(n)}_{N,i}$ we obtain

$$\begin{aligned}
&\operatorname{Cov}\left(\mathbb{1}_{\{U_i^{(n)} < \bar{\beta}_n(\xi)\}}, \mathbb{1}_{\{1 - U_i^{(n)} < \bar{\beta}_n(\xi)\}} \,\middle|\, \xi^{(n)}_{N,i} = \xi \right) \\
&\qquad = \mathbb{P}\left(1 - \bar{\beta}_n(\xi) < U_i^{(n)} < \bar{\beta}_n(\xi) \,\middle|\, \xi^{(n)}_{N,i} = \xi \right) - \bar{\beta}_n^2(\xi) \\
&\qquad = \mathbb{1}_{\{\bar{\beta}_n(\xi) > 1/2\}} (2\bar{\beta}_n(\xi) - 1) - \bar{\beta}_n^2(\xi) \,,
\end{aligned} \tag{A.14}$$

and, analogously,

$$\begin{aligned}
&\operatorname{Cov}\left(\mathbb{1}_{\{U_i^{(n)} < \bar{\beta}_n(\xi)\}}, \mathbb{1}_{\{1 - U_i^{(n)} \geq \bar{\beta}_n(\xi)\}} \,\middle|\, \xi^{(n)}_{N,i} = \xi \right) \\
&\qquad = \mathbb{P}\left(U_i^{(n)} \leq \min\{\bar{\beta}_n(\xi), 1 - \bar{\beta}_n(\xi)\} \,\middle|\, \xi^{(n)}_{N,i} = \xi \right) - \bar{\beta}_n(\xi)(1 - \bar{\beta}_n(\xi)) \\
&\qquad = \mathbb{1}_{\{\bar{\beta}_n(\xi) \leq 1/2\}} \bar{\beta}_n(\xi) + \mathbb{1}_{\{\bar{\beta}_n(\xi) > 1/2\}} (1 - \bar{\beta}_n(\xi)) - \bar{\beta}_n(\xi)(1 - \bar{\beta}_n(\xi)) \,.
\end{aligned}$$

(A.15)

Now, assume that $\bar{\beta}_n(\xi) > 1/2$; then, using (A.13)–(A.15),

$$\begin{aligned}
&\mathrm{Cov}\left(\tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+1}, \tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+2} \,\middle|\, \xi^{(n)}_{N,i} = \xi\right) \\
&= -[(\tau^{(1)}_n(\xi))^2 + (\tau^{(2)}_n(\xi))^2](1-\bar{\beta}_n(\xi))^2 + 2\tau^{(1)}_n(\xi)\tau^{(2)}_n(\xi)(1-\bar{\beta}_n(\xi))^2 - \eta_n^2 \\
&= -(\tau^{(1)}_n(\xi) - \tau^{(2)}_n(\xi))^2(1-\bar{\beta}_n(\xi))^2 - \eta_n^2 \,.
\end{aligned}$$

Moreover, that assuming $\bar{\beta}_n(\xi) \leq 1/2$ yields similarly

$$\mathrm{Cov}\left(\tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+1}, \tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+2} \,\middle|\, \xi^{(n)}_{N,i} = \xi\right) = -(\tau^{(1)}_n(\xi) - \tau^{(2)}_n(\xi))^2\bar{\beta}_n^2(\xi) - \eta_n^2 \,.$$

Finally, since $\tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+1}$ and $\tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+2}$ have, conditionally on $\xi^{(n)}_{N,i}$, the same marginal distributions, and

$$\begin{aligned}
&\mathrm{Var}\left(\tilde{\xi}^{(n+1)}_{N,\alpha(i-1)+1} \,\middle|\, \xi^{(n)}_{N,i} = \xi\right) \\
&= (\tau^{(1)}_n(\xi) - \tau^{(2)}_n(\xi))^2\,\mathrm{Var}\left(\mathbb{1}_{\{U_i^{(n)} < \bar{\beta}_n(\xi)\}} \,\middle|\, \xi^{(n)}_{N,i} = \xi\right) + \eta_n^2 \\
&= (\tau^{(1)}_n(\xi) - \tau^{(2)}_n(\xi))^2\bar{\beta}_n(\xi)(1-\bar{\beta}_n(\xi)) + \eta_n^2 \,,
\end{aligned}$$

the identity (4.10) follows.

References

Arvidsen, N. I., and Johnsson, T. (1982) Variance reduction through negative correlation—a simulation study. *J. Statist. Comput. Simulation* **15**, pp. 119–127.

Bollerslev, T., Engle, R. F., and Nelson, D. B. (1994) ARCH Models. In *The Handbook of Econometrics* (eds. Engle, R. F., and McFadden, D.), **4**, pp. 2959–3038. Amsterdam: North-Holland.

Cappé, O., Moulines, É., and Rydén, T. (2005) *Inference in Hidden Markov Models*. New York: Springer

Chopin, N. (2004) Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Ann. Statist.*, **32**, pp. 2385–2411.

Craiu, V. R., and Meng, X.-L. (2005) Multiprocess parallel antithetic coupling for backward and forward Markov chain Monte Carlo. *Ann. Stat.*, **33**, pp. 661–697.

Del Moral, P. (2004) *Feyman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. New York: Springer.

Douc, R., and Moulines, É. (2005) Limit theorems for weighted samples with applications to sequential Monte Carlo methods. To appear in *Ann. Stat.*

Douc, R., Moulines, É., and Olsson, J. (2008) Optimality of the auxiliary particle filter. To appear in *Prob. Math. Statist.*, **28**.

Doucet, A., de Freitas, N., and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice*. New York: Springer.

Doucet, A., Godsill, S., and Andrieu, C. (2000) On sequential Monte-Carlo sampling methods for Bayesian filtering. *Stat. Comput.*, **10**, pp. 197–208.

Hull, J., and White, A. (1987) The pricing of options on assets with stochastic volatilities. *J. Finance*, **42**, pp. 281–300.

Kitagawa, G. (1987) Non-Gaussian state space modeling of non-stationary time series. *J. Am. Statist. Assoc.*, **82**, pp. 1023–1063.

Pitt, M. K., and Shephard, N. (1999) Filtering via simulation: Auxiliary particle filters. *J. Am. Statist. Assoc.*, **87**, pp. 493–499.

Polson, N. G., Stroud, J. R., and Müller, P. (2002) Practical filtering with sequential parameter learning. Technical report, University of Chicago.