

---

## **Particle smoothing via Markov chain Monte Carlo in general state space models**

---

Meng Gao\*

Yantai Institute of Coastal Zone Research,  
Chinese Academy of Sciences,  
Yantai, 264003, China  
Email: gaomeng03@hotmail.com  
\*Corresponding author

Hui Zhang

School of Natural and Applied Sciences,  
Northwestern Polytechnical University,  
Xi'an, 710072, China  
Email: huizhang@nwpu.edu.cn

**Abstract:** Sequential Monte Carlo (SMC) methods (also known as particle filter) provide a way to solve the state estimation problem in nonlinear non-Gaussian state space models (SSM) through numerical approximation. Particle smoothing is one retrospective state estimation method based on particle filtering. In this paper, we propose a new particle smoother. The basic idea is easy and leads to a forward-backward procedure, where the Metropolis-Hastings algorithm is used to resample the filtering particles. The goodness of the new scheme is assessed using a nonlinear SSM. It is concluded that this new particle smoother is suitable for state estimation in complicated dynamical systems.

**Keywords:** Sequential Monte Carlo; SMC; particle filter; forward filtering-backward smoothing; Metropolis-Hastings.

**Reference** to this paper should be made as follows: Gao, M. and Zhang, H. (2018) 'Particle smoothing via Markov chain Monte Carlo in general state space models', *Int. J. Computing Science and Mathematics*, Vol. 9, No. 2, pp.181–188.

**Biographical notes:** Meng Gao is an Associate Professor at the YIC, CAS since 2009. He received his PhD in Applied Mathematics from Lanzhou University in 2008. He has been involved in numerous national and research projects, including statistical data assimilation and applied statistics for big data analysis. His research interests cover the broad scope of information and computational sciences. He is an author of numerous papers at refereed journal articles and reviewer for several international conferences.

Hui Zhang has been working as an Associate Professor at the School of Natural and Applied Sciences, Northwestern Polytechnical University since 2010. She received her PhD in Applied Mathematics from Lanzhou University in 2010. Her current research focus lies on biomathematics and computational biology.

## 1 Introduction

Stochastic state space models (SSMs) are widely used for modelling and predicting dynamic processes across natural sciences, social sciences and engineering (Wikle and Berliner, 2007). SSMs provide a standard framework for combining dynamic processes, system noises and measurement errors (Arulampalam et al., 2002; Kantas et al., 2009; Gao and Zhang, 2012). A generic SSM of discrete time is usually formulated as:

$$x_{t+1} = f_t(x_t, v_t) \quad (1)$$

$$y_t = h_t(x_t, w_t) \quad (2)$$

where  $x_t \in \mathfrak{R}^{n_x}$  denotes the state vector whose transition from  $t$  to  $t + 1$  is assumed to follow the Markovian rule. To be more specific,  $v_t$  and  $w_t$  are independent random vectors representing the system noise and measurement error, respectively. In addition,  $f_t : \mathfrak{R}^{n_x} \times \mathfrak{R}^{n_v} \rightarrow \mathfrak{R}^{n_x}$  is the system equation,  $y_t \in \mathfrak{R}^{n_y}$  is the measurement vector and  $h_t : \mathfrak{R}^{n_x} \times \mathfrak{R}^{n_w} \rightarrow \mathfrak{R}^{n_y}$  is the measurement equation. Typically, equations (1)–(2) are derived from physical, chemical, or biological principles, often turning out nonlinear (Doucet et al., 2001; Wikle and Berliner, 2007; Tulsyan et al., 2013).

Formally, a SSM is a partially observed Markov process and the estimation of unknown model states from incomplete or inaccurate observations represents the key problem (Kantas et al., 2009). Given the observations, state estimation can be categorised into *filtering* and *smoothing* depending if inference is performed in an online or offline manner (Doucet and Johansen, 2009; Douc et al., 2011; Neddermeyer, 2011). Currently, the theory and practice of filtering is quite well established while smoothing aspects, in particular in nonlinear models, have been less investigated (Doucet and Johansen, 2009). Smoothing comprises two related tasks:

- 1 the joint smoothing problem to infer the joint distribution of the entire state sequence,  $p(x_{1:T}|y_{1:T})$ , given all the observations (where  $T$  is the number of time steps)
- 2 the fixed-interval smoothing problem to infer the marginal distribution of each state,  $p(x_t|y_{1:T})$ , given all the observations (Bunch and Godsill, 2012).

In recent years, high performance computing makes it now possible the use of Monte Carlo methods to solve complex state estimation problem in nonlinear or non-Gaussian SSMs, where the tractable closed form solutions are rare. In particle filtering, Monte Carlo techniques are exploited to generate samples (or ‘particles’) to approximate the filtering distribution  $p(x_t|y_{1:t})$  (Gordon et al., 1993). Similar ideas also apply to the problem of smoothing (Kitagawa, 1996; Doucet et al., 2000; Tanizaki, 2001; Godsill

et al., 2004; Briers et al., 2010). It is also possible to develop a Markov Chain Monte Carlo (MCMC) algorithm to sample from the joint density  $p(x_{1:T}|y_{1:T})$  and, hence, from the marginal smoothing densities  $p(x_t|y_{1:T})$ . Pillonetto and Bell (2008) proposed a particle smoothing algorithm via Markov chain Monte Carlo, in which the proposal density draws samples from suitable approximations of the posterior distribution. In this paper, we propose a new particle smoother that also exploits MCMC and is carried out in a forward filtering-backward smoothing procedure. An advantage of our approach is that it is not necessary to compute the weights or to construct complicated proposal distributions. The basic idea is to implement backward smoothing by employing Metropolis-Hastings procedure to resample the filtering particles, so that convergence to the posterior distribution is guaranteed.

## 2 Particle filtering and smoothing

Let  $x_{1:T} = \{x_1, x_2, \dots, x_T\}$  and  $y_{1:T} = \{y_1, y_2, \dots, y_T\}$ , where  $[0, T]$  indicates the interval where smoothing has to be performed. Given the observations  $y_{1:T}$ , adopting the Bayesian perspective our aim is to derive the a posteriori probability density of the state sequence

$$\begin{aligned} p(x_{1:T}|y_{1:T}) &= \frac{p(y_{1:T}|x_{1:T})p(x_{1:T})}{p(y_{1:T})} \\ &\propto p(y_{1:T}|x_{1:T})p(x_{1:T}) \end{aligned} \quad (3)$$

Applying independence and Markovian assumptions to  $p(x_{1:T}|y_{1:T})$  and  $p(x_{1:T})$  yields

$$p(x_{1:T}|y_{1:T}) \propto p(x_1) \prod_{t=2}^T p(x_t|x_{t-1}) \prod_{t=1}^T p(y_t|x_t) \quad (4)$$

where  $p(x_t|x_{t-1})$  and  $p(y_t|x_t)$  can be derived from equations (1) and (2) respectively. Equation (4) suggests that the estimate of  $x_t$  can be updated sequentially when a new observation becomes available. Sequential state estimation is the focus of this section.

*Filtering* is online state estimation, where the posterior distribution  $p(x_t|y_{1:t})$  is updated in a sequential fashion. At each time step, we assume that  $p(x_{t-1}|y_{1:t-1})$  is already known and  $y_t$  is available, then we use  $p(x_{t-1}|y_{1:t-1})$  and  $y_t$  to find:

- 1 the forecast density,  $p(x_t|y_{1:t-1})$
- 2 the filtering density,  $p(x_t|y_{1:t})$ .

The sequential update is guaranteed by the following equations:

$$\begin{aligned} p(x_t|y_{1:t}) &= p(x_t|y_t, y_{1:t-1}) \\ &\propto p(y_t|x_t, y_{1:t-1})p(x_t|y_{t-1}) \\ &= p(y_t|x_t)p(x_t|y_{t-1}) \\ &= p(y_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \end{aligned} \quad (5)$$

*Smoothing* is instead based on a retrospective analysis of filtering. The aim is to obtain the marginal density  $p(x_t|y_{1:T})$ , also referred to as the smoothing density, from the posterior  $p(x_{1:T}|y_{1:T})$  in equation (4). The sequential update of  $p(x_t|y_{1:T})$  can be recursively implemented as follows. Having  $p(x_{t+1}|y_{1:T})$  at time  $t$ , we can derive  $p(x_t|y_{1:T})$  from

$$p(x_t|y_{1:T}) = \int p(x_t|x_{t+1}, y_{1:T})p(x_{t+1}|y_{1:T})dx_{t+1} \quad (6)$$

and

$$\begin{aligned} p(x_t|x_{t+1}, y_{1:T}) &= p(x_t|x_{t+1}, y_{1:t}) \\ &\propto p(x_{t+1}|x_t, y_{1:t})p(x_t|y_{1:t}) \\ &= p(x_{t+1}|x_t)p(x_t|y_{1:t}) \end{aligned} \quad (7)$$

where  $p(x_t|y_{1:t})$  is the filtered density. The procedures described in equations (5)–(7) provide the theoretical basis of sequential state estimation from Bayesian perspective. They are also known as *forward filtering-backward smoothing* (Kitagawa, 1996).

We now present a new particle smoother that exploits MCMC sampling to approximate the marginal smoothing density  $p(x_t|y_{1:T})$ . The MCMC technique generates a Markov chain having as stationary distribution the posterior of interest using the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970; Gilks et al., 1996). The idea we develop here is to use such sampler instead of importance sampling to generate samples from the density  $p(x_t|y_{1:T})$ . In particular, our particle smoothing algorithm is carried out in a forward filtering-backward smoothing procedure, with the MCMC procedure implemented in a sequential fashion starting from  $t = T$  to  $t = 1$ . For  $t = 1, \dots, T$ , let  $\{x_t^{(i)}\}_{i=1}^N$  denote the filtering particles coming from the forward procedure. Then, the new smoothing method is defined by the following two steps:

Step 1 Initialisation,  $t = T$ , set  $\{x_{T|T}^{(i)}\}_{i=1}^N = \{x_T^{(i)}\}_{i=1}^N$ .

Step 2  $t = T - 1, \dots, 1$ ,

a Randomly choose a sample from  $\{x_t^{(j)}\}_{j=1}^N$  as the start of the Markov chain denoted by  $x_{t|T}^{[0]}$ .

b Set  $k = k + 1$ . Then, randomly choose a sample  $x_t^*$  from filtering particle ensemble  $\{x_t^{(j)}\}_{j=1}^N$ , and compute the acceptance probability

$$\alpha = \min \left\{ 1, \frac{\sum_{j=1}^N p(x_{t+1|T}^{(j)}|x_t^*)}{\sum_{j=1}^N p(x_{t+1|T}^{(j)}|x_{t|T}^{[k-1]})} \right\}.$$

c Draw a random number  $z$  from Uniform distribution  $U(0, 1)$ . If  $z \leq \alpha$ , set  $x_{t|T}^{[k]} = x_t^*$ , otherwise set  $x_{t|T}^{[k]} = x_{t|T}^{[k-1]}$ .

d iterate steps b and c  $N - 1$  times, obtaining  $\{x_{t|T}^{(j)}\}_{j=1}^N$  as the output of the algorithm at instant  $t$ .

The above smoothing procedure can be seen as a resampling operation that, differently from other particle smoothing approaches, is based on MCMC. At each time step, the Metropolis-Hastings algorithm is used adopting an approximation of the true acceptance probability. Notice that, since the state estimation in this smoothing algorithm is performed sequentially (rather than in batch fashion) no high-dimensional proposal density needs to be constructed. We call this algorithm ‘M-H particle smoother’ for simplicity.

### 3 Numerical illustrations

In this section, we use one widely used example from literatures to test the proposed particle smoother. The example considers one extensively used nonlinear time series model from literatures (Gordon et al., 1993; Kitagawa, 1996; Godsill et al., 2004; Neddermeyer, 2011). The state-space equations are

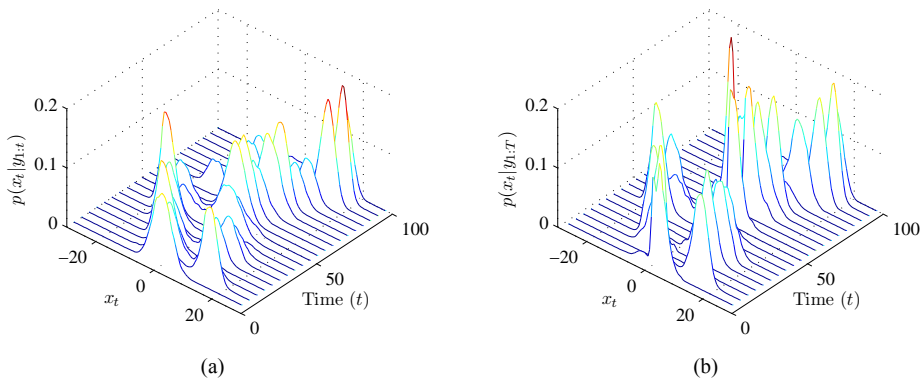
$$x_t = \frac{x_{t-1}}{2} + 25 \frac{x_{t-1}}{1 + x_{t-1}^2} + 8 \cos(1.2t) + v_t$$

and

$$y_t = \frac{(x_t)^2}{20} + w_t$$

where  $x_1 \sim N(0, 10)$ ,  $v_t \sim N(0, \sigma_v^2)$  and  $w_t \sim N(0, \sigma_w^2)$ , and here  $\sigma_v^2 = 10$  and  $\sigma_w^2 = 1$  are considered fixed and known. The length of time series is chosen as  $T = 100$ .

**Figure 1** Posterior probability density functions of  $x_t$ , (a)  $p(x_t|y_{1:t})$  generated by particle filter (SIR) (b)  $p(x_t|y_{1:T})$  generated by the proposed M-H particle smoother in this paper (see online version for colours)



Note: The number of particles is  $10^3$ .

Figure 1 shows the evolution of posterior distributions  $p(x_t|y_{1:t})$  and  $p(x_t|y_{1:T})$ , which are approximated by 1,000 particles. The probability density functions are obtained using kernel density estimation based on the filtering and smoothing particles (Martinez and

Martinez, 2002). It can be seen that the smoothing distribution  $p(x_t|y_{1:T})$  is a little narrower than the filtering distribution  $p(x_t|y_{1:t})$ .

**Table 1** Table of the particle filter and particle smoother algorithms and RMSE for different particle numbers in the nonlinear time series model

<i>Algorithm</i>	$N = 100$	$N = 200$	$N = 500$	$N = 1,000$
Particle filter (SIR)	5.2929	4.9652	4.4797	4.2765
Doucet et al.'s (2000) particle smoother	3.4881	2.8926	2.0625	1.7076
Tanizaki's (2001) particle smoother	3.6701	3.1364	2.1446	2.0374
Godsill et al.'s (2004) particle smoother	3.5906	2.9561	2.0934	1.7146
M-H particle smoother	3.6322	3.0159	2.1200	1.7208

There are many variants of particle smoother in literatures; however, it is not our purpose to compare all of them in this paper. So, we merely select one particle filter (SIR) (Arulampalam et al., 2002) and three particle smoother algorithms (Doucet et al., 2000; Tanizaki, 2001; Godsill et al., 2004). In this study, accuracy is measured in terms of root mean square error (RMSE) based on 100 independent simulations. The number of particles is  $N = 100, 200, 500$  and  $1,000$ . The results of ensemble simulations are summarised in Table 1.

#### 4 Conclusions and discussion

In this paper, we have proposed a new simple scheme for particle smoothing. The proposed M-H particle smoother algorithm is distinct from those based on sequential Monte Carlo sampling techniques (Kitagawa, 1996; Doucet et al., 2000; Tanizaki, 2001; Godsill et al., 2004; Briers et al., 2010; Fearnhead et al., 2010). Also, although the idea and procedure of M-H particle smoother algorithm is similar to that of the classical Metropolis-Hastings algorithm, it is not a 'strict' MCMC method. In our approach, Metropolis-Hastings algorithm is used only to generate smoothing particles by resampling the filtering particles. The resampling operation is easily implemented by point-wise comparison of the relative weight of the particles. The performance of M-H particle smoother algorithm was compared with that relative to other three previous particle smoother algorithms, also carried out in forward filtering-backward smoothing procedure. Numerical simulation showed that the M-H particle smoother algorithm is competitive, resulting in a good state estimator.

All particle smoother mentioned as well as the M-H scheme require the computation of the transition probability  $p(x_{t+1}|x_t)$ . It is crucial to identify the computation of transition probabilities when analysing the computational complexity of M-H particle smoother. The computations of  $p(x_{t+1}|x_t)$  needed for obtaining one particle at time  $t$  when adopting three representative particle smoother reported in references (Doucet et al., 2000; Tanizaki, 2001; Godsill et al., 2004; Briers et al., 2010). When M-H smoothing is adopted, although there are two summations used in evaluating the acceptance probability, the computational cost is not increased with respect to the other approaches since each sum is associated with one particle. Therefore, the computational complexity of M-H particle smoother is  $O(N^2T)$ , the same as the one relative to the other four algorithms.

As the actual computational time is dependent on programming optimisation and parallelisation, we do not compare the computational time explicitly in this paper. Our simulation results based on serial computing verified that the proposed M-H particle smoother algorithm has some advantages in terms of CPU-time too. Here we merely give a brief discussion on a possible optimisation of the algorithm from the perspective of parallelisation. For Doucet et al.'s (2000) particle smoother, the initialisation and re-weighting operations are readily parallelised; therefore, the computational time can be significantly reduced by parallel computing. Godsill et al.'s (2004) particle smoother is most suitable for parallel computing, because the realisations of joint smoothing distribution  $p(x_{1:T}|y_{1:T})$  can be generated independently. However, this parallelisation can only be implemented in distributed computation environment. For Tanizaki's (2001) particle smoother, parallel implementation of initialisation and weighting is also feasible, but resampling is not easy to implement in parallel (Bolić et al., 2005). For M-H particle smoother, it seems that parallelisation is impossible because Metropolis-Hastings algorithm is a sequential algorithm. Actually, M-H particle smoother may also benefit from parallel operations for computing the acceptance probability  $\alpha$ . For example, computing the two summations can be performed by using vectorised operations, a special parallel computing that can be achieved in many programming environment such as MATLAB. In addition, one can see that the expressions of numerator and denominator are similar and many common factors can be cancelled out. Then, the computational time of M-H smoothing algorithm can be effectively reduced.

## References

- Arulampalam, M.S., Maskell, S., Gordon, N. and Clapp, T. (2002) 'A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking', *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp.174–188.
- Bolić, M., Djurić P.M. and Hong, S. (2005) 'Resampling algorithms and architectures for distributed particles', *IEEE Transactions on Signal Processing*, Vol. 53, No. 7, pp.2442–2450.
- Briers, M., Doucet, A. and Maskell S. (2010) 'Smoothing algorithms for state-space models', *Annals of the Institute Statistical Mathematics*, Vol. 62, No. 1, pp.61–89.
- Bunch, P. and Godsill, S. (2012) 'Improved particle approximations to the joint smoothing distribution using Markov chain Monte Carlo', *IEEE Transactions on Signal Processing*, Vol. 61, No. 4, pp.956–963.
- Douc, R., Carivier, A., Moulines, E. and Olsson, J. (2011) 'Sequential Monte Carlo smoothing for general state space hidden Markov models', *Annals of Applied Probability*, Vol. 21, No. 6, pp.2109–2145.
- Doucet, A., Godsill, S.J. and Andrieu, C. (2000) 'On sequential Monte Carlo sampling methods for Bayesian filtering', *Statistics and Computing*, Vol. 10, No. 3, pp.197–208.
- Doucet, A., de Freitas, N. and Gordon, N. (2001) *Sequential Monte Carlo in Practice*, Springer-Verlag, New York.
- Doucet, A. and Johansen, A. (2009) 'A tutorial on particle filtering and smoothing: fifteen years later', *Oxford Handbook of Nonlinear Filtering*, Oxford University Press, UK.
- Fearnhead, P., Wyncoll, D. and Tawn, J. (2010) 'A sequential smoothing algorithm with linear computational cost', *Biometrika*, Vol. 97, No. 2, pp.447–464.
- Gao, M. and Zhang, H. (2012) 'Sequential Monte Carlo methods for parameter estimation in nonlinear state-space models', *Computers & Geosciences*, Vol. 44, pp.70–77.

- Gilks, W.R., Richardson, S. and Spiegelhalter, D.J. (1996) *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London.
- Godsill, S.J., Doucet, A. and West, M. (2004) 'Monte Carlo smoothing for nonlinear time series', *Journal of the American Statistical Association*, Vol. 99, No. 465, pp.156–168.
- Gordon, N., Salmond, D. and Smith, A.F.M. (1993) 'Novel approach to nonlinear and non-Gaussian Bayesian state estimation', *IEE Proc. F. Radar Signal Process*, Vol. 140, No. 2, pp.107–113.
- Hastings, W.K. (1970) 'Monte Carlo sampling methods using Markov chains and their applications', *Biometrika*, Vol. 57, No. 1, pp.97–109.
- Kantas, N., Doucet, A., Singh, S. and Maciejowski, J. (2009) 'An overview of sequential Monte Carlo methods for parameter estimation in general state-space models', *Proc. IFAC Symposium on System Identification (SYSID)*.
- Kitagawa, G. (1996) 'Monte Carlo filter and smoother for nonlinear non-Gaussian state-space models', *Journal of Computational and Graphical Statistics*, Vol. 5, No. 1, pp.1–25.
- Martinez, W.L. and Martinez, A.R. (2002) *Computational Statistics Handbook with MATLAB*, Chapman and Hall, UK.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953) 'Equation of state calculations by fast computing machines', *J. Chem. Phys.*, Vol. 21, No. 6, pp.1087–1091.
- Neddermeyer, J.C. (2011) 'Nonparametric particle filtering and smoothing with quasi-Monte Carlo sampling', *Journal of Statistical Computation and Simulation*, Vol. 81, No. 11, pp.1361–1379.
- Pillonetto, G. and Bell, B.M. (2008) 'Optimal smoothing of nonlinear dynamic systems via Monte Carlo Markov chains', *Automata*, Vol. 44, No. 7, pp.1676–1685.
- Tanizaki, H. (2001) 'Nonlinear and non-Gaussian state space modeling using sampling techniques', *Annals of the Institute of Statistical Mathematics*, Vol. 53, No. 1, pp.63–81.
- Tulsyan, A., Huang, B., Gopaluni, R.B. and Forbes, J.E. (2013) 'On simultaneous on-line state and parameter estimation in non-linear state-space models', *Journal of Process Control*, Vol. 23, No. 4, pp.516–526.
- Wikle, C.K. and Berliner, L.M. (2007) 'A Bayesian tutorial for data assimilation', *Physica D*, Vol. 230, Nos. 1–2, pp.1–16.