

# Static-parameter estimation in piecewise deterministic processes using particle Gibbs samplers\*

Axel Finke<sup>†‡</sup>     Adam M. Johansen<sup>‡</sup>     Dario Spanò<sup>‡</sup>

7th February 2014

Department of Statistics  
University of Warwick  
Coventry CV4 7AL, UK

## Abstract

We develop particle Gibbs samplers for static-parameter estimation in discretely-observed piecewise deterministic processes (PDPS). PDPS are stochastic processes that jump randomly at a countable number of stopping times but otherwise evolve deterministically in continuous time. A sequential Monte Carlo (SMC) sampler for filtering in PDPS has recently been proposed. We first provide new insight into the consequences of an approximation inherent within that algorithm. We then derive a new representation of the algorithm. It simplifies ensuring that the importance weights exist and also allows the use of variance-reduction techniques known as backward and ancestor sampling. Finally, we propose a novel Gibbs step that improves mixing in particle Gibbs samplers whose SMC algorithms make use of large collections of auxiliary variables, such as many instances of SMC samplers. We provide a comparison between the two particle Gibbs samplers for PDPS developed in this paper. Simulation results indicate that they can outperform reversible-jump MCMC approaches.

*Keywords:* Particle MCMC · Particle Gibbs sampler · Piecewise deterministic processes · Sequential Monte Carlo

---

\*This text corresponds to the accepted version of a manuscript which is in press for *The Annals of the Institute of Statistical Mathematics (Tokyo)*; <http://www.ism.ac.jp/editsec/aism/>.

<sup>†</sup>Corresponding author: [a.finke@warwick.ac.uk](mailto:a.finke@warwick.ac.uk)

<sup>‡</sup>AF was supported by Engineering and Physical Sciences Research Council (EPSRC) doctoral training grant EP/J500586/1. AMJ was partially supported by EPSRC grant EP/I017984/1 and is grateful to the Isaac Newton Institute for a visiting fellowship which allowed him time to work on this manuscript during the programme: Inference for Change-Point and Related Processes. DS' research was partially funded by CRISM, an EPSRC/HEFCE-funded grant.

# 1 Introduction

Piecewise deterministic processes (PDPS) are stochastic processes that jump randomly at a countable number of stopping times but otherwise evolve deterministically in continuous time. In this article, we employ sequential Monte Carlo (SMC)-based methods to conduct inference on the static parameters in PDPS which are observed only partially, noisily and in discrete time. Such models are more general than state-space models and inference for them is often more difficult.

Simple particle filters for PDPS, based around the bootstrap approach and termed variable rate particle filters (VRPFs), were introduced by Godsill and Vermaak (2004) and a corresponding smoothing algorithm for non-degenerate models was suggested by Bunch and Godsill (2013). In order to apply more sophisticated particle filtering techniques to these models, an SMC filter for PDPS, based on the SMC-sampler framework by Del Moral et al (2006), was introduced in Whiteley et al (2011).

However, methods for efficiently estimating the static parameters in such models still need to be developed. A few approaches have been proposed in the literature. A stochastic expectation-maximisation algorithm based on a reversible-jump MCMC (RJMCMC) sampler (Green, 1995) was introduced by Centanni and Minozzo (2006a,b). A simple SMC sampler was attempted in Del Moral et al (2007) to which some improvements were made in Martin et al (2012). In addition, Rao and Teh (2012) developed a Gibbs sampler for the special case in which the state space is discrete.

Our approach is to employ a particle Gibbs sampler as introduced by Andrieu et al (2010), based around the SMC filter for PDPS from Whiteley et al (2011), to estimate the static parameters. Our methodological contributions are as follows.

- (1) We provide new insight into the approximation induced by the SMC filter for PDPS and by related algorithms used in Del Moral et al (2006, 2007); Martin et al (2012). We also suggest a way of ensuring the existence of the importance weights.
- (2) We derive a new representation of the algorithm that – for non-degenerate models – permits the use of backward and ancestor sampling, essential variance-reduction techniques for particle Gibbs samplers that were suggested by Whiteley (2010); Whiteley et al (2010) and Lindsten et al (2012).
- (3) We propose a novel way of rejuvenating the potentially large number of auxiliary variables used in the SMC filter for PDPS (and more generally). This reduces their impact on the overall mixing of the algorithm at virtually no extra computational cost, even resulting in computational savings in many situations.

We demonstrate our method on two challenging examples. Our simulations indicate that it can compete with both a VRPF-based particle Gibbs sampler and a RJMCMC sampler, at a potentially lower computational cost. We also investigate the impact of the approximation inherent within the algorithm.

The structure of this paper is as follows. Section 2 defines PDPS and provides motivating examples. Section 3 recapitulates SMC methods. Section 4 describes the SMC filter for PDPS, investigates some of its properties, and proposes a novel representation for it. Section 5 describes static-parameter estimation via particle Gibbs samplers and introduces a novel Gibbs step. Section 6 provides simulation results.

## 2 Piecewise deterministic processes

### 2.1 Definition

In this section, we introduce discretely-observed piecewise deterministic processes, the class of models with which the remainder of this article is concerned. They are stochastic processes that jump randomly at an almost surely countable number of random times but otherwise evolve deterministically in continuous time. Their description here follows Whiteley et al (2011). We also provide motivating examples.

First, we clarify some notational conventions used throughout this work. We denote by  $\mathcal{L}^\theta(\cdot)$  (or sometimes  $\mathcal{L}^\theta(dx)$ ) the distribution of a random variable  $X$  indexed by some parameter  $\theta$  while  $\mathcal{L}^\theta(x)$  denotes its density with respect to some  $\sigma$ -finite measure,  $dx$ , e.g. with respect to the Lebesgue or counting measure, evaluated at some point  $x$ . In particular,  $\delta_z$  denotes the Dirac measure or point mass at  $z$ . In addition,  $\#A$  denotes the cardinality of some set  $A$ . For  $m, n \in \mathbb{N}$  with  $m \leq n$ , we use the notation  $m:n := (m, m+1, \dots, n)$  and  $\llbracket n \rrbracket := \{k \in \mathbb{N} \mid k \leq n\}$ . For vectors  $x = (x_1, \dots, x_n)$  and  $a = (a_1, \dots, a_k)$ , where  $\{a_1, \dots, a_k\} \subseteq \llbracket n \rrbracket$ , we write  $x_a := (x_{a_1}, \dots, x_{a_k})$ . Finally,  $x^{-a} = x \setminus x_a$  denotes the vector that is identical to  $x$ , except that the components  $x_{a_1}, \dots, x_{a_k}$  have been removed.

Let  $(\tau_j, \phi_j)_{j \in \mathbb{N} \cup \{0\}}$  be a stochastic process such that  $0 = \tau_0 < \tau_1 < \tau_2 < \dots$  and such that each  $\phi_j$  takes a value in some non-empty set  $\Phi$ . In addition, define a deterministic function  $F^\theta: [0, \infty)^2 \times \Phi \rightarrow \Phi$  that satisfies  $F^\theta(t, t, \cdot) = \text{id}$  for any  $t \geq 0$ . A *piecewise deterministic process (PDP)* is then a continuous-time stochastic process  $\zeta := (\zeta_t)_{t \geq 0}$  with initial condition  $\zeta_0 = \phi_0$  and such that

$$\zeta_t = F^\theta(t, \tau_{\nu_t}, \phi_{\nu_t}),$$

for  $t > 0$ . Here,  $\nu_t := \sup\{j \in \mathbb{N} \cup \{0\} \mid \tau_j \leq t\}$ , so that  $\tau_{\nu_t}$  represents the time of the last jump before time  $t$ . In other words, after time  $\tau_{j-1}$ , the PDP evolves deterministically in continuous time according to  $F^\theta$  until it reaches the next *jump time*  $\tau_j$ , at which point the process randomly jumps to a new value given by the *jump size*  $\phi_j$ . Here and throughout,  $\theta$  denotes the ordered set of all static parameters present in the model, that is,  $\theta$  contains all the parameters that do not change over time and which therefore cannot be estimated via standard (particle) filtering methods.

Let  $0 = t_0 < t_1 < t_2 < \dots$  be known (non-random) times and let  $K_n := \nu_{t_n}$  – with realisations  $k_n$  and convention  $k_0 = 0$  – denote the number of jumps before time  $t_n$ , then  $\zeta_{[0, t_n]} := (\zeta_t)_{t \in [0, t_n]}$  is completely determined by  $(K_n, \tau_{1:K_n}, \phi_{1:K_n}, \phi_0)$ . For simplicity, as in Whiteley et al (2011), we assume the following Markovian prior on the number, times and sizes of jumps in the interval  $[0, t_n]$  for any  $n \in \mathbb{N}$ ,

$$\begin{aligned} p_n^\theta(k_n, \tau_{1:k_n}, \phi_{0:k_n}) &= S^\theta(t_n, \tau_{k_n}) q_0^\theta(\phi_0) \mathbb{1}_{(0, t_n]}(\tau_{k_n}) \\ &\quad \times \prod_{j=1}^{k_n} q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1}) f^\theta(\tau_j | \tau_{j-1}), \end{aligned}$$

where  $q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1}) f^\theta(\tau_j | \tau_{j-1})$  is the step- $j$  transition kernel of  $(\tau_j, \phi_j)_{j \in \mathbb{N} \cup \{0\}}$  with the support of  $f^\theta(\tau_j | \tau_{j-1})$  being  $(\tau_{j-1}, \infty)$ ,  $q_0^\theta(\phi_0)$  is the distribution of the initial

jump size, and finally,  $S^\theta(t, \tau) := 1 - \int_\tau^t f^\theta(ds|\tau)$  denotes the probability of no jump occurring in the interval  $(\tau, t]$  (for  $\tau \leq t$ ).

Inference for such models becomes necessary if we assume that  $\zeta$  can be observed only partially, at discrete times, and subject to some measurement error. Observations may be recorded at fixed or random times. Let  $y_{(s,t]}$  denote the vector of all observations in the interval  $(s, t]$  for some  $0 < s < t < \infty$ , the density of which is represented by  $g^\theta(y_{(s,t]}|\zeta_{(s,t]})$ . Observations in disjoint time intervals are assumed to be conditionally independent given the PDP. This implies that

$$g^\theta(y_{(0,t_n]}|\zeta_{(0,t_n]}) = g^\theta(y_{[\tau_{k_n}, t_n]}|\tau_{k_n}, \phi_{k_n}) \prod_{j=1}^{k_n} g^\theta(y_{[\tau_{j-1}, \tau_j]}|\tau_{j-1}, \phi_{j-1}),$$

where we sometimes use the notation  $g^\theta(y_{[\tau_{j-1}, \tau_j]}|\tau_{j-1}, \phi_{j-1}) = g^\theta(y_{[\tau_{j-1}, \tau_j]}|\zeta_{[\tau_{j-1}, \tau_j]})$  to stress that  $\zeta_{[\tau_{j-1}, \tau_j]}$  is conditionally independent of all the other jump times, jump sizes, and the total number of jumps, given  $(\tau_j, \tau_{j-1}, \phi_{j-1})$  (and given  $\theta$ ).

The conditional independence of observations is reminiscent of state-space models. However, as mentioned earlier, PDPs are more general than state-space models. Indeed, state-space models may be viewed as PDPs in which  $f^\theta(\tau_j|\tau_{j-1})$  is degenerate, i.e. in which the number of jumps and the jump times are known. Hence, for the remainder of this work, we assume that  $f^\theta(\tau_j|\tau_{j-1})$  is non-degenerate.

The conditional posterior distribution of the jumps up to time  $t_n$  (as well as their number) may then be written as

$$\begin{aligned} \tilde{\pi}_n^\theta(k_n, \tau_{1:k_n}, \phi_{0:k_n}) &= \tilde{\gamma}_n^\theta(k_n, \tau_{1:k_n}, \phi_{0:k_n}) / \mathcal{Z}_n^\theta \\ &:= p_n^\theta(k_n, \tau_{1:k_n}, \phi_{0:k_n}) g^\theta(y_{(0,t_n]}|\zeta_{(0,t_n]}) / \mathcal{Z}_n^\theta, \end{aligned} \quad (1)$$

where the normalising constant  $\mathcal{Z}_n^\theta > 0$  is typically unknown.

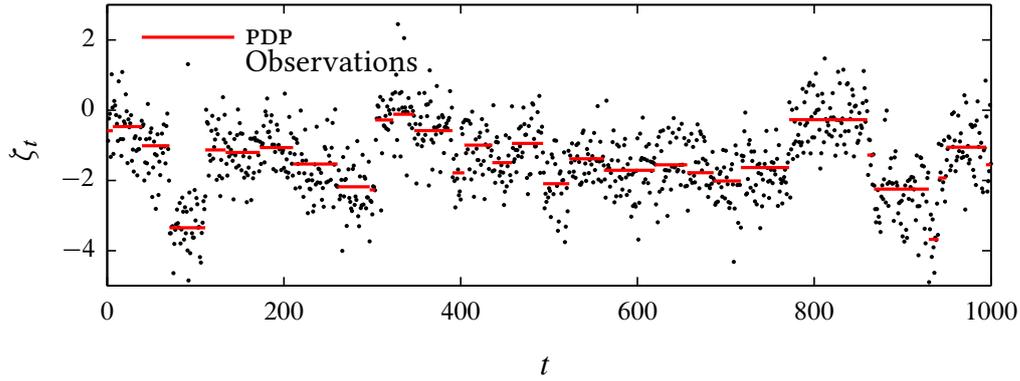
Explicitly including the dimensionality parameter  $K_n$  within the state space ensures that for all  $n \in \mathbb{N}$ , the step- $n$  posterior distributions are defined on increasing subsets of the same space, i.e. the support of  $\tilde{\pi}_n^\theta$  is a subset of

$$\tilde{\mathbb{E}}_n := \bigcup_{k=0}^{\infty} (\{k\} \times \mathbb{T}_{(0,t_n],k} \times \Phi^{k+1}),$$

where  $\mathbb{T}_{(s,t],k} := \{(\tau_1, \dots, \tau_k) \in (0, \infty)^k \mid s < \tau_1 < \dots < \tau_k \leq t\}$ . This representation makes explicit the unknown number of jumps in any interval of time,  $(0, t_n]$ .

## 2.2 Example I: elementary change-point model

This subsection introduces an elementary change-point model as a first example of a PDP. We assume that the interjump times are distributed according to some parametric family indexed by a parameter vector  $\theta_\tau$ . Conditional on the jump times, the jump sizes follow a Gaussian AR(1)-process, i.e.  $q^\theta(\phi_j|\phi_{j-1}, \tau_j, \tau_{j-1}) = \mathcal{N}(\phi_j; \rho\phi_{j-1}, \sigma_\phi^2)$ , for  $\rho \in \mathbb{R}$ . The deterministic function is taken to be piecewise constant, i.e. given by  $F^\theta(t, \tau, \phi) := \phi$ . Observations are recorded at regular intervals of length  $\Delta$  and are formed by adding Gaussian noise with mean 0 and variance  $\sigma_y^2$  to the PDP.



**Figure 1:** PDP and observations simulated from the elementary change-point model.

Figure 1 shows data simulated from the model over a horizon of  $T = 1000$  time units with  $\Delta = 1$ ,  $\rho = 0.9$ ,  $\sigma_\phi^2 = 1$  and  $\sigma_y^2 = 0.5$  using gamma-distributed interjump times with shape and scale parameters  $\theta_\tau := (\alpha, \beta) = (4, 10)$ .

As  $\zeta$  is only discretely and noisily observed, (particle) filtering methods are generally needed to conduct inference about the jump times and jump sizes. In addition, the static parameters  $\theta := (\rho, \sigma_\phi^2, \sigma_y^2, \theta_\tau) \in \mathbb{R} \times (0, \infty)^4$  have to be estimated.

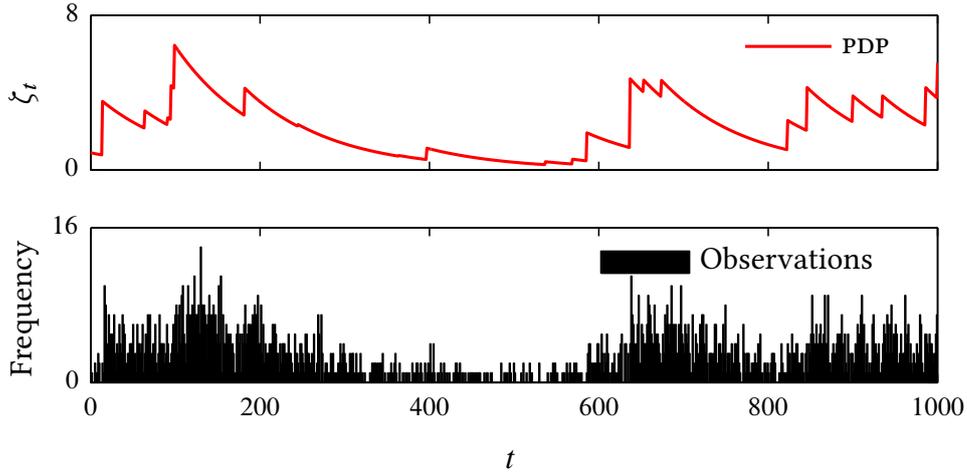
### 2.3 Example II: shot-noise Cox process

A second example of a PDP is a shot-noise-Cox-process model described in Whiteley et al (2011). The model assumes that observations are taken on a Cox process with piecewise deterministic shot-noise intensity,  $\zeta = (\zeta_t)_{t \geq 0}$ .

Such models have applications in finance, as described in Centanni and Minozzo (2006a,b): in the modelling of ultra-high-frequency financial data, observations are two-dimensional, comprising the time and size of the price movements of a stock. That is, the stock price process (which can be fully observed) is piecewise constant since the quoted price is only updated at a countable collection of random times. The times at which the stock price changes are realisations of a Cox process with unobserved shot-noise intensity  $\zeta$ .

The latent intensity process  $\zeta$  has the following interpretation. The  $j$ th stopping time,  $\tau_j$ , corresponds to the arrival of the  $j$ th news item at the market. This causes a positive jump in the intensity process, whose size,  $\phi_j > 0$ , depends on the ‘importance’ of the news item. Between  $\tau_j$  and  $\tau_{j+1}$  the intensity gradually decays as the news is absorbed by the market. The intensity process thus governs the amount of activity in the market: each jump leads to an increase in the trading activity as measured by the number of subsequent change points in the (observed) price process.

Such models are also used in insurance to price catastrophe insurance derivatives as described in Dassios and Jang (2003). In this context, the observations are only one-dimensional and represent the times at which claims are being recorded. In other words, the claim arrival process is a Cox process with intensity process  $\zeta$ . The  $j$ th jump in the intensity process, at time  $\tau_j$ , thus corresponds to a catastrophic event. The associated jump size  $\phi_j$  characterises the event’s severity.



**Figure 2:** Data simulated from the shot-noise Cox process model. *Top:* intensity process. *Bottom:* histogram of the observations using a bin width of 2.5.

More precisely, we have  $q^\theta(\phi_0) = \lambda_\phi \exp(-\lambda_\phi \phi_0) \mathbb{1}_{[0, \infty)}(\phi_0)$ , as well as

$$f^\theta(\tau_j | \tau_{j-1}) := \lambda_\tau \exp(-\lambda_\tau (\tau_j - \tau_{j-1})) \mathbb{1}_{(\tau_{j-1}, \infty)}(\tau_j),$$

$$q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1}) := \lambda_\phi \exp(-\lambda_\phi (\phi_j - \zeta_{\tau_j}^-)) \mathbb{1}_{(\zeta_{\tau_j}^-, \infty)}(\phi_j),$$

where  $\zeta_{\tau_j}^- := \phi_{j-1} \exp(-\kappa(\tau_j - \tau_{j-1}))$  is the intensity immediately before the  $j$ th jump. Furthermore at any time  $t$ , the intensity is a deterministic function of  $t$  as well as of the most recent jump time and jump size as follows,

$$\zeta_t = F^\theta(t, \tau_{v_t}, \phi_{v_t}) := \phi_{v_t} \exp(-\kappa(t - \tau_{v_t})).$$

In addition, the likelihood of the observations in the interval  $(t_{n-1}, t_n]$  (the times at which claims are recorded in this interval), denoted  $y_{(t_{n-1}, t_n]}$ , is given by

$$g^\theta(y_{(t_{n-1}, t_n]} | \zeta_{(t_{n-1}, t_n]}) \propto \exp\left(-\int_{t_{n-1}}^{t_n} \zeta_s ds\right) \prod_{i: y_i \in (t_{n-1}, t_n]} \zeta_{y_i}.$$

Figure 2 shows an example trajectory and observations simulated from the model with  $\kappa = 1/100$ ,  $\lambda_\tau = 1/40$ ,  $\lambda_\phi = 2/3$ .

As the process  $\zeta$  is not directly observed, (particle) filtering methods are needed to conduct inference about the jump times and jump sizes in the intensity process. In addition, the static parameters  $\theta := (\kappa, \lambda_\tau, \lambda_\phi) \in (0, \infty)^3$  must be estimated.

## 2.4 Example III: object tracking

This subsection briefly mentions, as a third example of a PDP, a model for tracking fighter aircraft from Whiteley et al (2011).

In this model, the PDP represents the evolution of position, speed, and velocity of the aircraft. The assumption is that the pilot accelerates or decelerates at a countable

collection of random times which correspond to the jumps in the PDP. Between jumps, the aircraft's location and speed are deterministic functions – given by the standard equations of motion – of the location, speed, and acceleration at the most recent jump time as well as of the time elapsed since the most recent jump. However, only countably many noisy observations on the location of the aircraft are available.

While filtering for this model was shown to be feasible in Whiteley et al (2011), it exhibits a characteristic that makes static-parameter estimation difficult: the transitions  $q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1})$  have degenerate components because the location and speed components of the PDP evolve continuously, i.e. they only have trivial jumps.

In anticipation of Subsection 5.2 we note here that the variance-reduction techniques mentioned therein cannot be applied to such degenerate problems which makes particle-Gibbs-sampler-based inference impractical. However, RJMCMC-based algorithms, such as those from Centanni and Minozzo (2006a,b); Del Moral et al (2007); Martin et al (2012) will not be practical for such models either and the problem remains generally unsolved.

### 3 Sequential Monte Carlo methods

#### 3.1 Generic smc algorithm

In this section, we recapitulate sequential Monte Carlo methods which can be used, among other things, for filtering in PDPs. They are also at the heart of particle Markov chain Monte Carlo methods which are discussed in Section 5.

*Sequential Monte Carlo (SMC)* methods are Monte Carlo algorithms for approximating a sequence of related distributions,  $(\pi_n^\theta)_{n \in \mathbb{N}}$ , which are defined on spaces of increasing dimension,  $(E^{(n)})_{n \in \mathbb{N}}$ . They may be viewed as sequential importance sampling with added resampling steps (e.g. Doucet et al, 2001; Doucet and Johansen, 2011), as importance sampling on a suitably extended space (Andrieu et al, 2010) or as interacting-particle approximations of Feynman–Kac flows (Del Moral, 2004).

Here and throughout, we assume that only an unnormalised version of (the measure or density)  $\pi_n^\theta, \gamma_n^\theta := \pi_n^\theta \mathcal{Z}_n^\theta$  for some unknown normalising constant  $\mathcal{Z}_n^\theta > 0$ , can be evaluated. SMC algorithms approximate  $\pi_{n-1}^\theta(dx_{1:n-1})$  at step  $n - 1$  with

$$\hat{\pi}_{n-1}^\theta(dx_{1:n-1}) := \sum_{i=1}^N W_{n-1}^i \delta_{X_{1:n-1}^i}(dx_{1:n-1}), \quad (2)$$

which is the weighted empirical measure corresponding to a collection of  $N$  samples  $\{X_{1:n-1}^i : i \in \llbracket N \rrbracket\}$ , often referred to as ‘particle’ trajectories, and corresponding weights,  $\{W_{n-1}^i : i \in \llbracket N \rrbracket\}$ , which are normalised to sum to 1. This approximation of  $\pi_{n-1}^\theta$  at step  $n - 1$  is propagated to approximate  $\pi_n^\theta$  at step  $n$ . For this, the trajectories are first extended by sampling additional particle locations  $\{X_n^i : i \in \llbracket N \rrbracket\}$  from some stochastic kernel  $K_n^\theta(\cdot | X_{1:n-1}^i)$ . The extended trajectories  $X_{1:n}^1, \dots, X_{1:n}^N$  are then reweighted to ensure that their weighted empirical measure approximates  $\pi_n^\theta$ .

The computational complexity of these algorithms is typically  $O(N)$  at each iteration. This important feature of SMC methods is one reason for their successful application to the problem of (on-line) filtering in non-Gaussian or non-linear state-space models

(Gordon et al, 1993; Del Moral, 1995; Kitagawa, 1996). In this particular setting, SMC methods are often called ‘particle filters’.

In the following, we write  $\mathbf{X}_n := (X_n^1, \dots, X_n^N)$  and  $\mathbf{W}_n := (W_n^1, \dots, W_n^N)$  for all the particles and importance weights generated at the  $n$ th step of the SMC algorithm. Realisations of these random vectors are denoted  $\mathbf{x}_n$  and  $\mathbf{w}_n$ . To ease the notational burden, we adopt the convention that whenever actions or definitions are discussed for the  $i$ th particle, it is intended that they should be applied for all  $i \in \llbracket N \rrbracket$ .

Occasionally, the algorithm discards trajectories with small weights and multiplies trajectories with large weights. This is known as *resampling*. It ensures that subsequent iterations of the algorithm focus on propagating only those particle trajectories which provide a good characterisation of the current target distribution. For this work, we use the interpretation of resampling given by Andrieu et al (2010): the  $i$ th particle at step  $n$ ,  $X_n^i$ , is viewed as an offspring of particle  $A_{n-1}^i$  at step  $n-1$ . To keep track of the ancestral lineages of the particles, define  $B_{n|n}^i := i$ , and, for  $p \in \llbracket n-1 \rrbracket$ ,

$$B_{p|n}^i := A_p^{B_{p+1|n}^i}$$

so that  $B_{p|n}^i$  is the index of the step- $p$  ancestor of the  $i$ th particle at step  $n$ . We also use the convention that  $X_{1:n}^i$  refers to the  $i$ th particle trajectory at step  $n$ . Thus,

$$X_{1:n}^i := (X_{1:n-1}^{A_{n-1}^i}, X_n^i) = (X_1^{B_{1|n}^i}, \dots, X_n^{B_{n|n}^i}).$$

The  $N$  particle trajectories  $X_{1:n}^1, \dots, X_{1:n}^N$  obtained after resampling, are associated with an evenly-distributed set of (normalised) importance weights, i.e.  $W_n^i = 1/N$ . For this, it is required that the resampling scheme – the conditional distribution  $r^\theta(\cdot | \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1})$  generating the parent indices  $\mathbf{A}_n := (A_n^1, \dots, A_n^N)$  – is *unbiased* in the sense that for each  $i \in \llbracket N \rrbracket$ ,  $\mathbb{E}[\sum_{j=1}^N \mathbb{1}_{\{A_n^j=i\}} | \mathbf{X}_{1:n}, \mathbf{A}_{1:n-1}] = N W_n^i$ . Widely used unbiased resampling schemes are *multinomial* (Gordon et al, 1993), *residual* (Liu and Chen, 1998), *stratified* (Kitagawa, 1996), and *systematic* resampling (Carpenter et al, 1999). An extensive comparison can be found in Douc et al (2005).

Due to resampling, all particles will eventually share a common ancestor, i.e. for all  $p \in \mathbb{N}$ ,  $\#\{B_{p|n}^i : i \in \llbracket N \rrbracket\} \downarrow 1$  as  $n \rightarrow \infty$ . This is commonly referred to as *sample impoverishment*. As a result, for some small integer  $m$ , the approximation of marginals  $\pi_n^\theta(x_{1:m}) = \int \pi_n^\theta(x_{1:n}) dx_{m+1:n}$  will be based on few distinct particles and will not be reliable. The following steps alleviate (but usually do not prevent) this impoverishment.

- (a) Employing low-variance resampling schemes, such as residual, stratified or systematic resampling, and, in particular, avoiding multinomial sampling.
- (b) Resampling only if the variation in the weights becomes too severe. A commonly used approach, theoretically justified by Del Moral et al (2012), is to resample only if the estimated *effective sample size*,  $\widehat{ESS}_n := 1/[\sum_{i=1}^N (W_n^i)^2]$  (Kong et al, 1994), has dropped below some threshold  $N_{\text{ESS}}$ . More formally, the decision of whether or not to resample at step  $n$  may depend on  $n$  and the realisation  $(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1})$ . Not resampling at step  $n$  can then simply be treated as if  $r^\theta(\cdot | \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}) = \delta_{1:N}$  (but without resetting the weights).
- (c) Devising more efficient proposal kernels  $K_n^\theta$ .

For the remainder of this subsection, we provide details on the calculation of the importance weights in Equation 2. Assume that the particle trajectories  $X_{1:n-1}^i$  are weighted to target  $\pi_{n-1}^\theta$ , and that additional variables  $X_n^i \sim K_n^\theta(\cdot | X_{1:n-1}^i)$  have been sampled. Since  $Z_n^\theta$  is unknown, only the *unnormalised*  $i$ th incremental importance weight  $G_n(X_{1:n}^i)$  can be evaluated. Here,  $G_1^\theta(x_1) := \gamma_1^\theta(x_1)/K_1^\theta(x_1)$  and, for  $n > 1$ ,

$$G_n^\theta(x_{1:n}) := \frac{\gamma_n^\theta(x_{1:n})}{\gamma_{n-1}^\theta(x_{1:n-1})K_n^\theta(x_n|x_{1:n-1})}.$$

The *normalised*  $i$ th weight at step  $n$  – needed to construct the weighted empirical measure defined in Equation 2 – is given by  $W_1^i := G_1^\theta(X_1^i)/\sum_{j=1}^N G_1^\theta(X_1^j)$  and

$$W_n^i := \frac{G_n^\theta(X_{1:n}^i)W_{n-1}^{A_{n-1}^i}}{\sum_{j=1}^N G_n^\theta(X_{1:n}^j)W_{n-1}^{A_{n-1}^j}},$$

if  $n > 1$ . In particular, if the algorithm has resampled at step  $n - 1$ ,

$$W_n^i = \frac{G_n^\theta(X_{1:n}^i)}{\sum_{j=1}^N G_n^\theta(X_{1:n}^j)}. \quad (3)$$

One of the strengths of SMC methods is that

$$\hat{Z}_n^\theta(\mathbf{X}_{1:n}, \mathbf{A}_{1:n-1}) := \prod_{p=1}^n \left[ \frac{1}{N} \sum_{i=1}^N G_p^\theta(X_{1:p}^i) \right],$$

assuming for simplicity that resampling occurs at every iteration, is an unbiased estimator for the normalising constant,  $Z_n^\theta$ . This was first shown in Del Moral (1996).

A generic SMC algorithm is summarised in Algorithm 1. It admits almost all other SMC algorithms, e.g. SMC samplers (Del Moral et al, 2006), auxiliary particle filters (Pitt and Shephard, 1999; Johansen and Doucet, 2008) as well as the algorithms presented in Section 4, as special cases.

### Algorithm 1 (SMC)

- (1) Sample  $X_1^i \sim K_1^\theta$ , set  $W_1^i = G_1^\theta(X_1^i)/\sum_{j=1}^N G_1^\theta(X_1^j)$  and set  $n := 2$ .
- (2) If  $r^\theta(\cdot | \mathbf{X}_{1:n-1}, \mathbf{A}_{1:n-2}) = \delta_{1:N}$ , set  $A_{n-1}^i := i$ .  
Otherwise, sample  $\mathbf{A}_{n-1} \sim r^\theta(\cdot | \mathbf{X}_{1:n-1}, \mathbf{A}_{1:n-2})$  and set  $W_{n-1}^i := 1/N$ .
- (3) Sample  $X_{1:n}^i \sim K_n^\theta(\cdot | X_{1:n-1}^{A_{n-1}^i})$  and set  $X_{1:n}^i := (X_{1:n-1}^{A_{n-1}^i}, X_n^i)$ .
- (4) Set  $W_n^i := G_n^\theta(X_{1:n}^i)W_{n-1}^{A_{n-1}^i}/\sum_{j=1}^N G_n^\theta(X_{1:n}^j)W_{n-1}^{A_{n-1}^j}$ .
- (5) Set  $n \leftarrow n + 1$  and go back to Step 2.

So far, the SMC methods outlined above only generate samples weighted to target the distribution of time-varying parameters  $X_{1:n}$ , conditional on knowing any static parameters  $\theta$ , i.e. parameters that do not change over time. Various methods have been proposed to also estimate  $\theta$  by SMC methods, dating back to at least Kitagawa (1998). Several likelihood-based approaches for estimating  $\theta$  in the context of state-space

models have been developed (e.g. Poyiadjis et al, 2011); see also Kantas et al (2009) for a recent overview of parameter-estimation approaches in such models. Off-line Bayesian methods for quite general settings are summarised in Section 5.

The following subsection describes the SMC-sampler framework of Del Moral et al (2006) around which Whiteley et al (2011) and this paper develop algorithms for inference in PDPS.

### 3.2 SMC samplers

Recall that at the beginning of the previous subsection, it was assumed that the target distributions  $(\pi_n^\theta)_{n \in \mathbb{N}}$  were defined on spaces of *increasing* dimension. However, SMC methods can actually be used to target distributions  $\tilde{\pi}_n^\theta = \tilde{\gamma}_n^\theta / \mathcal{Z}_n^\theta$  defined on spaces  $\tilde{E}_n$  of *arbitrary* dimension. To circumvent the calculation of marginal proposal densities – these would be required to evaluate the importance weights – SMC methods target ‘extended’ distributions,  $\pi_n^\theta = \gamma_n^\theta / \mathcal{Z}_n^\theta$ , which

- (a) are defined on spaces  $(E^{(n)})_{n \in \mathbb{N}}$  whose dimensions grow appropriately with  $n$ ,
- (b) have the property that for each  $n \in \mathbb{N}$ ,  $\pi_n^\theta$  admits  $\tilde{\pi}_n^\theta$  as a marginal.

The SMC-sampler framework developed by Del Moral et al (2006) provides a recipe for constructing these extended target distributions. The idea is to define

$$\pi_n^\theta(x_{1:n}) \propto \gamma_n^\theta(x_{1:n}) := \tilde{\gamma}_n^\theta(x_n) \prod_{p=1}^{n-1} L_p^\theta(x_p | x_{p+1}),$$

on  $E^{(n)} := \prod_{p=1}^n \tilde{E}_p$ , where  $L_p^\theta(x_p | x_{p+1})$  is a ‘backward’ Markov kernel from  $\tilde{E}_{p+1}$  to  $\tilde{E}_p$ . The  $i$ th unnormalised incremental weight is then given by  $G_n^\theta(X_{1:n}^i)$ , where  $G_1^\theta(x_1) = \gamma_1^\theta(x_1) / K_1^\theta(x_1)$  and, for  $n > 1$ ,

$$G_n^\theta(x_{1:n}) = \frac{\gamma_n^\theta(x_{1:n})}{\gamma_{n-1}^\theta(x_{1:n-1}) K_n^\theta(x_n | x_{1:n-1})} = \frac{\tilde{\gamma}_n^\theta(x_n) L_{n-1}^\theta(x_{n-1} | x_n)}{\tilde{\gamma}_{n-1}^\theta(x_{n-1}) K_n^\theta(x_n | x_{1:n-1})}.$$

Del Moral et al (2006) provide guidelines for the choice of efficient backward kernels.

Their approach also allows the use of a mixture of forward kernels by including the index of the step- $n$  mixture component,  $M_n$ , into  $X_n$  and targeting the further extended distribution  $\pi_n^\theta(x_{1:n}) \propto \gamma_n^\theta(x_{1:n})$  on  $E^{(n)} := \prod_{p=1}^n E_p$ . Here,  $E_p := (M \times \tilde{E}_p)$ , where  $M$  is the countable set of all mixture component indices. Writing  $X_n = (M_n, Z_n)$ , the unnormalised version of this further extended distribution is

$$\gamma_n^\theta(x_{1:n}) := \tilde{\gamma}_n^\theta(z_n) \beta_0^\theta(m_1 | z_1) \prod_{p=1}^{n-1} L_p^\theta(z_p, m_{p+1} | z_{p+1}). \quad (4)$$

Backward mixture kernels

$$L_p^\theta(z_p, m_{p+1} | z_{p+1}) = \beta_p^\theta(m_{p+1} | z_{p+1}) L_{p, m_{p+1}}^\theta(z_p | z_{p+1}),$$

need to be employed if the (forward) proposal kernels are mixture kernels of the form

$$K_p^\theta(x_p|x_{1:p-1}) = \alpha_p^\theta(m_p|z_{p-1})K_{p,m_p}^\theta(z_p|z_{p-1}).$$

Here,  $\alpha_p^\theta(\cdot|z_{p-1})$  and  $\beta_{p-1}^\theta(\cdot|z_p)$  are distributions on  $M$  which determine the forward and backward kernel mixture weights at step  $n$ . At step 1, it is common to sample from a single distribution (not from a mixture), so that  $\beta_0^\theta(dm_1|z_1) = \alpha_1^\theta(dm_1) = \delta_m(dm_1)$  for some  $m \in M$ .

We note that it is not usual to view these mixture component indices as forming a permanent part of the state or to retain them through subsequent steps; in the present context, we do so for reasons which will become apparent in the sequel.

## 4 SMC filter for piecewise deterministic processes

### 4.1 Variable rate particle filter

In this section, we describe filtering for PDPS via SMC methods. All three algorithms presented in this section may be viewed as special cases the generic SMC algorithm from Section 3. Hence, we always use the same symbols  $X_{1:n}$ ,  $\pi_n^\theta$ ,  $\gamma_n^\theta$ ,  $K_n^\theta$  and  $G_n^\theta$  to refer to the ‘states’, normalised and unnormalised target distributions, proposal kernels and unnormalised incremental weights even though the particular form of these quantities may change between the next three subsections. Throughout the entire section,  $\tilde{\gamma}_n^\theta$ ,  $\tilde{E}_p$  and  $\mathcal{Z}_n^\theta$  are defined as in Subsection 2.1.

The first particle filter for PDPS, termed *variable rate particle filter (VRPF)*, was proposed by Godsill and Vermaak (2004). The VRPF is simply an application of the generic SMC algorithm to a slightly reparametrised model described in the following.

Let  $0 = t_0 < t_1 < t_2 < \dots$ , where  $t_p$ , for  $p > 1$ , represents the time of the  $p$ th SMC step. Moreover, let  $(\tau_{p,k}, \phi_{p,k})$  denote the  $k$ th jump time in the interval  $(t_{p-1}, t_p]$  and its associated jump size. Let  $k_p \geq 0$  be the total number of jumps in this interval and define the ‘states’ to be  $X_1 := (k_1, \tau_{1,1:k_1}, \phi_{1,1:k_1}, \phi_0)$  and  $X_n := (k_n, \tau_{n,1:k_n}, \phi_{n,1:k_n})$ , for  $n > 1$ . These take values in subsets of  $E_1 := \bigcup_{k=0}^{\infty} (\{k\} \times \mathbb{T}_{(0,t_1],k} \times \Phi^{k+1})$  and  $E_n := \bigcup_{k=0}^{\infty} (\{k\} \times \mathbb{T}_{(t_{n-1},t_n],k} \times \Phi^k)$ , respectively.

Let  $\nu(n) := \sup\{m \in \llbracket n-1 \rrbracket \mid k_m > 0\}$  with the convention that  $\tau_{\nu(n),k_{\nu(n)}} = \tau_0 = 0$  and  $\phi_{\nu(n),k_{\nu(n)}} = \phi_0$  if  $\nu(n) = -\infty$ . In other words,  $\nu(n)$  represents the index of the last interval of the form  $(t_{p-1}, t_p]$  before  $(t_{n-1}, t_n]$  in which the PDPS has had a jump. The target distribution is then given by  $\pi_n^\theta := \gamma_n^\theta / \mathcal{Z}_n^\theta$  on  $E^{(n)} := \prod_{p=1}^n E_p$ , where

$$\begin{aligned} \gamma_n^\theta(x_{1:n}) &:= S^\theta(t_n, \tau_{\nu(n),k_{\nu(n)}})q_0^\theta(\phi_0)g^\theta(y_{(0,t_n]}|\zeta_{(0,t_n]}) \\ &\times \prod_{p \in \tilde{D}_n} \left[ f^\theta(\tau_{p,1}|\tau_{\nu(p),k_{\nu(p)}})q^\theta(\phi_{p,1}|\phi_{\nu(p),k_{\nu(p)}}) \right. \\ &\quad \left. \times \mathbb{1}_{\mathbb{T}_{(t_{p-1},t_p],k_p}}(\tau_{p,1:k_p}) \prod_{j=2}^{k_p} q^\theta(\phi_{p,j}|\phi_{p,j-1}, \tau_{p,j}, \tau_{p,j-1}) f^\theta(\tau_{p,j}|\tau_{p,j-1}) \right]. \end{aligned}$$

Here,  $\tilde{D}_n := \{p \in \llbracket n \rrbracket \mid k_p > 0\}$  is the collection of indices of intervals of the form  $(t_{p-1}, t_p]$  that contain at least one jump. The PDPS is  $\zeta_t := F^\theta(t, \tau_{\nu(n),k_{\nu(n)}}, \phi_{\nu(n),k_{\nu(n)}})$ , for  $t \in (t_{n-1}, t_n]$  if  $k_n = 0$ , and for  $t \in (t_{n-1}, \tau_{n,1})$  if  $k_n > 0$ . In the latter case, we also

have  $\zeta_t := F^\theta(t, \tau_{n,j}, \phi_{n,j})$ , for  $t \in [\tau_{n,j}, \tau_{n,j+1})$ , with the convention that  $\tau_{n,k_n+1} = t_n$ . The distribution  $\pi_n^\theta$  admits  $\tilde{\pi}_n^\theta$  as a marginal.

The algorithm proceeds by sampling values  $X_n \sim K_n^\theta(\cdot | x_{1:n-1})$  at step  $n$ , where

$$K_n^\theta(x_n | x_{1:n-1}) := K_{n,1}^\theta(k_n | x_{1:n-1}) K_{n,2}^\theta(\tau_{n,1:k_n}, \phi_{n,1:k_n} | k_n, x_{1:n-1}).$$

In the above equation, the kernels on the right hand side are selected in such a way that the usual absolute-continuity conditions are satisfied. At step 1, the kernel  $K_{1,2}^\theta$  also samples a value for  $\phi_0$ .

An unnormalised incremental weight at step  $n$  is then given by the following expressions. If  $k_n = 0$ , then

$$G_n^\theta(x_{1:n}) = \frac{S^\theta(t_n, \tau_{v(n), k_{v(n)}})}{S^\theta(t_{n-1}, \tau_{v(n), k_{v(n)}})} \frac{g^\theta(y_{(t_{n-1}, t_n]} | \phi_{v(n), k_{v(n)}}, \tau_{v(n), k_{v(n)}})}{K_{n,1}^\theta(k_n | x_{1:n-1})}.$$

If  $k_n \geq 1$ , then

$$\begin{aligned} G_n^\theta(x_{1:n}) &= \frac{S^\theta(t_n, \tau_{n, k_n})}{S^\theta(t_{n-1}, \tau_{v(n), k_{v(n)}})} \frac{g^\theta(y_{(t_{n-1}, t_n]} | \phi_{v(n), k_{v(n)}}, \tau_{v(n), k_{v(n)}})}{K_n^\theta(x_n | x_{1:n-1})} \\ &\times \left[ \prod_{j=1}^{k_n-1} g^\theta(y_{(\tau_{n,j}, \tau_{n,j+1})} | \phi_{n,j}, \tau_{n,j}) \right] g^\theta(y_{(\tau_{n, k_n}, t_n]} | \phi_{n, k_n}, \tau_{n, k_n}) \\ &\times q^\theta(\phi_{n,1} | \phi_{v(n), k_{v(n)}}, \tau_{n,1}, \tau_{v(n), k_{v(n)}}) f^\theta(\tau_{n,1} | \tau_{v(n), k_{v(n)}}) \\ &\times \prod_{j=2}^{k_n} q^\theta(\phi_{n,j} | \phi_{n,j-1}, \tau_{n,j}, \tau_{n,j-1}) f^\theta(\tau_{n,j} | \tau_{n,j-1}). \end{aligned}$$

As shown in Whiteley et al (2011), the VRPF can suffer from severe sample impoverishment. This is because at step  $n$ , jumps are only proposed in the interval  $(t_{n-1}, t_n]$  and only based on information available up to time  $t_n$ . If subsequent observations are highly informative about jumps in  $(t_{n-1}, t_n]$ , as they usually are in PDPS, then this information can only be incorporated through reweighting existing jumps.

The SMC filter from Whiteley et al (2011), outlined below, can reduce sample impoverishment because it allows new jumps to be sampled anywhere after the most recent jump and also allows previously generated jumps to be adjusted.

## 4.2 SMC filter from Whiteley et al (2011)

The SMC filter for PDPS from Whiteley et al (2011) is based on the SMC-sampler framework described in Subsection 3.2. That is, it is a ‘standard’ SMC algorithm that targets a sequence of artificial extended distributions  $\pi_n^\theta := \gamma_n^\theta / \mathcal{Z}_n^\theta$  (as in Equation 4) defined on  $E^{(n)} := \prod_{p=1}^n E_p$  with  $E_p := (M \times \tilde{E}_p)$ , by means of mixture proposal kernels.

We now add an additional subscript to the model parameters to account for the fact that for any particle, the  $j$ th jump time or jump size at the  $n$ th step of the algorithm may be different from the  $j$ th jump time or jump size at step  $n - 1$ . Thus, we hereafter write  $X_n := (M_n, K_n, \tau_{n,1:k_n}, \phi_{n,0:k_n})$  for a particle at step  $n$ . To ease the notational burden, we often write  $Z_n := X_n \setminus M_n = (K_n, \tau_{n,1:k_n}, \phi_{n,0:k_n})$ .

In the most basic form presented in this work, there are just two mixture components,  $M = \{a, b\}$ . At step  $n$ , an *adjustment* move ( $M_n = a$ ),

$$K_{n,a}^\theta(dz_n|z_{n-1}) = \delta_{k_{n-1}}(dk_n) \delta_{\tau_{n-1,1:k_{n-1}-1}}(d\tau_{n,1:k_{n-1}}) \delta_{\phi_{n-1,0:k_{n-1}-1}}(d\phi_{n,0:k_{n-1}}) \\ \times \rho_{n,a}^\theta(d\tau_{n,k_n}|z_{n-1}) \eta_{n,a}^\theta(d\phi_{n,k_n}|\tau_{n,k_n}, z_{n-1}),$$

moves the most recent stopping time to a new location according to a distribution  $\rho_{n,a}^\theta(\cdot|z_{n-1})$  with support  $(\tau_{n-1,k_{n-1}-1}, t_n]$  and samples a new value for the corresponding jump size from a distribution  $\eta_{n,a}^\theta(\cdot|\tau_{n,k_n}, z_{n-1})$  on  $\Phi$ . A *birth* move ( $M_n = b$ ),

$$K_{n,b}^\theta(dz_n|z_{n-1}) = \delta_{k_{n-1}+1}(dk_n) \delta_{\tau_{n-1,1:k_{n-1}}}(\tau_{n,1:k_{n-1}}) \delta_{\phi_{n-1,0:k_{n-1}}}(\phi_{n,0:k_{n-1}}) \\ \times \rho_{n,b}^\theta(d\tau_{n,k_n}|z_{n-1}) \eta_{n,b}^\theta(d\phi_{n,k_n}|\tau_{n,k_n}, z_{n-1}),$$

adds a new stopping time by sampling it from a distribution  $\rho_{n,b}^\theta(\cdot|z_{n-1})$  with support  $(\tau_{n-1,k_{n-1}}, t_n]$ . Additionally, a new jump-size parameter is sampled from a distribution  $\eta_{n,b}^\theta(\cdot|\tau_{n,k_n}, z_{n-1})$  on  $\Phi$ .

As proposed in Whiteley et al (2011), the forward mixture weights may be set to  $\alpha_n^\theta(a|z_{n-1}) := S^\theta(t_n, \tau_{n-1,k_{n-1}})$  as well as  $\alpha_n^\theta(b|z_{n-1}) := 1 - \alpha_n^\theta(a|z_{n-1})$ , i.e. the probability of a birth move grows as the time to the last jump increases. At step 1, a birth move is enforced for each particle so that  $\alpha_1^\theta(dm_1) = \delta_b(dm_1)$ .

The corresponding backward kernel component for an adjustment move is

$$L_{n-1,a}^\theta(dz_{n-1}|z_n) := \delta_{k_n}(dk_{n-1}) \delta_{\tau_{n,1:k_n-1}}(d\tau_{n-1,1:k_{n-1}-1}) \delta_{\phi_{n,0:k_n-1}}(d\phi_{n-1,0:k_{n-1}-1}) \\ \times Q_{n-1,a}^\theta(d[\tau_{n-1,k_{n-1}}, \phi_{n-1,k_{n-1}}]|z_n),$$

where  $Q_{n-1,a}^\theta(\cdot|z_n)$  is a distribution whose support is a subset of  $(\tau_{n-1,k_{n-1}-1}, t_{n-1}] \times \Phi$ . For a birth move, the corresponding backward kernel component is

$$L_{n-1,b}^\theta(dz_{n-1}|z_n) = \delta_{k_{n-1}}(dk_{n-1}) \delta_{\tau_{n,1:k_n-1}}(d\tau_{n-1,1:k_{n-1}}) \delta_{\phi_{n,0:k_n-1}}(d\phi_{n-1,0:k_{n-1}}).$$

The adjustment and birth move kernels only affect the most recent jump time and jump size. This is a reasonable approach as smc filters can only be expected to work for ergodic models and for these, this strategy should be adequate. Nonetheless, other moves could easily be incorporated. For instance, the second, say, most recent jump time or jump size may also be modified.

Indeed, as noted by Whiteley et al (2011), a kernel for multiple-birth moves should be included because otherwise, the above choice of forward/backward kernels induces an approximation. However, the probability of such moves is typically so small that this leads to computationally the same algorithm. To keep the presentation simple, we refrain from including such moves here (as was done in Del Moral et al (2006, 2007)), although there is no technical difficulty with so doing.

In the following, we characterise the approximation induced by the above choice of forward and backward kernels. Let  $b_j := \inf\{q \in \mathbb{N} \mid \sum_{l=1}^q \mathbb{1}_{\{b\}}(m_l) = j\}$  denote the index of the smc step at which the  $j$ th birth move occurs and let

$$\tilde{s}(\tau_{1:j}) := \sup\{s(\tau_{j-l+1}) + l - 1 \mid l \in \llbracket j \rrbracket\},$$

where  $s(\tau) := \inf\{q \in \mathbb{N} \mid t_q \geq \tau\}$ . Throughout, we assume that the number of jumps up to time  $t_0 = 0$  is set to zero. The proposal distribution  $K_1^\theta(x_1) \prod_{p=2}^n K_p^\theta(x_p | x_{1:p-1})$  then has support

$$E^{r,(n)} := \{x_{1:n} \in E^{(n)} \mid b_1 = 1 \text{ and } \forall j \in \{2, \dots, k_n\} : \tilde{s}(\tau_{n,1:k_j-1}) < b_j \leq n\}.$$

In particular, the marginal distribution of  $X_n$  under the proposal distribution has support

$$\tilde{E}_n^r := \{(k_n, \tau_{1:k_n}, \phi_{0:k_n}) \in \tilde{E}_n \mid \forall j \in \llbracket k_n \rrbracket : s(\tau_j) \leq n - k_n + j\}.$$

Recall that we write  $z_n = x_n \setminus m_n = (k_n, \tau_{1:k_n}, \phi_{0:k_n})$ . To ensure that the importance weights exist, the algorithm can therefore only target, as a marginal, the distribution

$$\tilde{\pi}_n^{r,\theta}(z_n) \propto \tilde{\gamma}_n^{r,\theta}(z_n) \propto \tilde{\gamma}_n^\theta(z_n) \mathbb{1}_{\tilde{E}_n}(z_n).$$

If the time between successive SMC steps,  $t_n - t_{n-1}$ , is short compared to the average time between jumps, the difference between the distribution in Equation 1 and the ‘actual’ target distribution  $\tilde{\pi}_n^{r,\theta}$  should be negligible. We will consider the influence of this approximation in Subsection 6.2

The extended target distribution of the algorithm is given by

$$\pi_n^{r,\theta}(x_{1:n}) \propto \tilde{\gamma}_n^{r,\theta}(z_n) \beta_0^\theta(m_1 | z_1) \prod_{p=1}^{n-1} \beta_p^\theta(m_{p+1} | z_{p+1}) L_{p,m_{p+1}}^\theta(z_p | z_{p+1}),$$

where we assume, for the moment, that the backward mixture weights  $\beta_p^\theta(\cdot | z_{p+1})$  can be chosen such that this extended target distribution does not have probability mass outside of  $E^{r,(n)}$  to ensure that the importance weights exist. A detailed discussion of the choice of backward mixture weights is given below.

We conclude this subsection by describing some implementation issues regarding the above-mentioned SMC algorithm. To our knowledge, they have not been pointed out in the literature. The point we wish to stress here is that backward and proposal kernels need to be chosen carefully and in a manner that is consistent with each other in order to avoid introducing biases resulting from a loss of absolute continuity, for instance. Such biases may be small in the case of filtering (i.e. if the static parameters are known). However, if the static parameters are to be estimated alongside the jump times and jump sizes, even small biases in the filter can induce large biases in the estimates of the static parameters.

*Jump-size proposal kernels.* It was advocated in Whiteley et al (2011) to sample the jump sizes from their full conditional posterior distribution. However, given the structure of the algorithm, this posterior distribution will often be based on observations ‘too far’ into the future, i.e. after another jump which will be added in subsequent SMC steps with high probability. We therefore recommend to only take observations from the interval  $(0, t_n \wedge (\tau_{n,k_n} + \lambda \mu_\tau])$  into account when sampling  $\phi_{n,k_n}$ . Here,  $\lambda \in (0, 1)$  and  $\mu_\tau$  may be the mean interjump time or a quantile of the interjump-time distribution.

*Backward mixture weights.* As previously mentioned, the backward mixture weights must be chosen such that the extended target distribution does not have probability mass outside of  $E^{r,(n)}$ . The most obvious problem with a poor choice of backward mixture weights is that the extended target distribution does not actually admit the right marginal (in addition to having ill-defined importance weights).

There is a one-to-one correspondence between  $\sum_{p=1}^n \mathbb{1}_{\{b\}}(m_p)$ , the number of birth moves, and  $k_n$ , the number of jumps in the proposal distribution and hence in the support of the truncated target distribution  $\tilde{\pi}_n^{r,\theta}$ . Therefore we cannot specify their distributions independently. The target already specifies a distribution over  $k_n$ ; if the backward mixture weights  $\beta_p^\theta(\cdot | z_{p+1})$  do not depend on  $k_{p+1}$ , then they implicitly specify a second distribution over  $k_n$  and the marginal distribution of this quantity under the target distribution will not be what is intended.

For instance, consider setting the backward mixture kernel weights to a uniform distribution over  $M = \{a, b\}$  – a popular choice. Write  $A_n := \{p \in \llbracket n-1 \rrbracket \mid m_{p+1} = a\}$  and  $B_n := \llbracket n-1 \rrbracket \setminus A_n$ , then the algorithm targets as a marginal

$$\begin{aligned}
& \int \pi_n^{r,\theta}(x_{1:n}) d[x_{1:n} \setminus z_n] \\
& \propto \tilde{\pi}_n^{r,\theta}(z_n) \int \sum_{m_{1:n} \in \{b\} \times M^{n-1}} \left[ \prod_{p \in A_n} L_{p,a}^\theta(dz_p | z_{p+1}) \right] \left[ \prod_{p \in B_n} L_{p,b}^\theta(dz_p | z_{p+1}) \right] \\
& = \tilde{\pi}_n^{r,\theta}(z_n) \int \sum_{m_{1:n} \in \{b\} \times M^{n-1}} \left[ \prod_{p \in A_n} \delta_{k_{p+1}}(dk_p) \delta_{\tau_{p+1,1:k_{p+1}-1}}(d\tau_{p,1:k_p-1}) \right. \\
& \quad \left. \times \delta_{\phi_{p+1,0:k_{p+1}-1}}(d\phi_{p,0:k_p-1}) Q_{p,a}^\theta(d[\tau_{p,k_p}, \phi_{p,k_p}] | z_{p+1}) \right] \\
& \quad \times \left[ \prod_{p \in B_n} \delta_{k_{p+1}-1}(dk_p) \delta_{\tau_{p+1,1:k_{p+1}-1}}(d\tau_{p,1:k_p}) \delta_{\phi_{p+1,0:k_{p+1}-1}}(d\phi_{p,0:k_p}) \right] \\
& = \tilde{\pi}_n^{r,\theta}(z_n) \# \left\{ m_{1:n} \in \{b\} \times M^{n-1} \mid \sum_{p=1}^n \mathbb{1}_{\{b\}}(m_p) = k_n \right\} \\
& = \tilde{\pi}_n^{r,\theta}(z_n) \binom{n-1}{k_n-1}.
\end{aligned}$$

For regular proposal kernels, a possible choice of backward kernels restricting the support of the extended target distribution to  $E^{r,(n)}$  is given by

$$\beta_{p-1}^\theta(b | z_p) = \begin{cases} 0, & \text{if } k_p = 1 \text{ and } p > 1, \\ 1, & \text{if } k_p = p \text{ or } p = 1 \text{ or } p = \tilde{s}(\tau_{p,1:k_p-1}) + 1, \\ q_p(z_p), & \text{otherwise,} \end{cases} \quad (5)$$

for some probability  $q_p(z_p) \in (0, 1)$  which may depend on  $z_p$ .

*Local adjustment moves.* Ideally, adjustment moves should direct jumps towards regions of higher posterior probability. If such moves cannot be devised, it is preferable to use *local* adjustment moves, e.g. small-scale Gaussian kernels centred around the current location of the jump. This reduces the risk of moving jumps away from regions of high

posterior probability, which would add to sample impoverishment. However, such local adjustment moves are unlikely to move a jump currently contained in  $(t_{p-1}, t_p]$  out of such an interval. Therefore, even using Equation 5 could result in importance weights with infinite variance.

A simple remedy is to employ *restricted* adjustment moves, i.e. local moves that are limited to the particular interval  $(t_{p-1}, t_p]$  currently containing the jump. More formally, recall that  $s(\tau) = \inf\{q \in \mathbb{N} \mid t_q \geq \tau\}$ . For restricted adjustment moves,  $\rho_{n,a}^\theta(\cdot | z_{n-1})$  has support  $((\tau_{n-1, k_{n-1}-1} \vee t_{s_{n-1}-1}), t_{s_{n-1}}]$ , where  $s_{n-1} := s(\tau_{n-1, k_{n-1}})$ , rather than having support  $(\tau_{n-1, k_{n-1}-1}, t_n]$ .

Also recalling that  $\tilde{s}(\tau_{1:j}) = \sup\{s(\tau_{j-l+1}) + l - 1 \mid l \in \llbracket j \rrbracket\}$ , the support of the joint proposal distribution is then given by

$$E^{r,(n)} = \{x_{1:n} \in E^{(n)} \mid b_1 = \tilde{s}(\tau_{n,1}) = 1 \text{ and } \forall j \in \{2, \dots, k_n\} : \tilde{s}(\tau_{n,1:k_j}) \leq b_j \leq n\}.$$

To ensure that the target distribution does not have probability mass outside of  $E^{r,(n)}$ , the distribution of  $\tau_{n-1, k_{n-1}}$  under  $Q_{n-1,a}^\theta(\cdot | z_n)$  must have support  $((\tau_{n, k_n-1} \vee t_{s_n-1}), t_{s_n}]$ , where  $s_n := s(\tau_{n, k_n})$ . In addition, the backward mixture weights might take the form presented in Equation 5 but with  $\tilde{s}(\tau_{p,1:k_p-1})$  replaced by  $\tilde{s}(\tau_{p,1:k_p}) - 1$ .

### 4.3 A novel reformulation of the SMC filter

One problem with the SMC filter for PDPS from the previous subsection – henceforth referred to as the ‘original’ SMC filter – is that it induces unnecessary degeneracy in the transitions at step  $n$  because most jump times and jump sizes in  $X_{n-1}$  coincide with jump times and jump sizes in  $X_n$ . In other words, the algorithm works explicitly on the path space by embedding a ‘standard’ SMC filter within the SMC-sampler framework using (mostly) trivial degenerate backward transitions.

Unfortunately, these degenerate backward transitions prevent the use of many backward-simulation methods and thus prevent the use of the essential variance-reduction techniques described in Subsection 5.2. At the same time, the algorithm does not gain any benefit from the path-space representation.

In the following, we present a novel representation of the algorithm whose alternative step- $n$  extended target distribution also admits  $\tilde{\pi}_n^{r,\theta}$  as a marginal but whose ‘states’ do not induce degenerate transitions, as long as  $q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1})$  is non-degenerate. In addition, this algorithm makes it easier to ensure the existence of the importance weights as it circumvents the problem of choosing sensible backward mixture weights. Our representation may be viewed as a way of extracting the ‘standard’ SMC filter contained within the original SMC filter for PDPS. This yields an extended target distribution whose structure is reminiscent of the *product-space* formulation from Carlin and Chib (1995) (see also Godsill, 2001).

The algorithm presented in this subsection targets an extended distribution (defined further below) that contains all the ‘states’  $X_{1:n}$ , where  $X_n := (M_n, \tau_n, \phi_n)$ , for  $n > 1$ , takes values in a subset of  $E_n := M \times (0, t_n] \times \Phi$  and  $X_1 := (M_1, \tau_1, \phi_1, \phi_0)$  takes values in a subset of  $E_1 := M \times (0, t_n] \times \Phi^2$ . In this subsection,  $\tau_n$  and  $\phi_n$  are the jump time and associated jump size sampled at the  $n$ th step of the SMC algorithm as part of a birth move or as part of an adjustment move. As before,  $M_n$  indicates an adjustment move

$(M_n = a)$  or birth move ( $M_n = b$ ) at step  $n$ . Again, the mixture kernel indices are added to the state space as auxiliary variables.

The main idea is to use the mixture component indices  $M_{1:n}$  to keep track of which jumps (i.e. jump times and sizes) affect the ‘actual’ target distribution  $\tilde{\pi}_n^{r,\theta}$ . These are the jumps sampled in SMC steps  $p \in H_n$ , where

$$H_n := \{j \in \llbracket n-1 \rrbracket \mid m_{j+1} = b\} \cup \{n\}.$$

That is,  $H_n$  contains the indices of all jumps which have been sampled immediately before a birth move in steps  $1, \dots, n$ . We also define  $V_n := \llbracket n \rrbracket \setminus H_n$  to be the set of indices of the remaining jumps. For easier reference, we collect all elements of  $H_n$  in the vector  $h_n = (h_n(1) \dots h_n(\#H_n))$  and all the elements of  $V_n$  in the vector  $v_n = (v_n(1) \dots v_n(n - \#H_n))$ , each in ascending order.

In the following, we present the extended target distribution of the algorithm. To show that it admits the right marginal, some reparametrisation is required: write  $k_n := \#H_n$  for the total number of birth moves in the first  $n$  steps and let  $i_{1:k_n}$  denote the SMC steps at which these birth moves occur, i.e.  $i_1 := 1$  and  $i_j := h_n(j-1) + 1$  for  $j \in \{2, \dots, k_n\}$ . This allows the one-to-one transformation

$$[m_{1:n}, (\tau_{h_n}, \phi_{h_n}), (\tau_{v_n}, \phi_{v_n})] \longleftrightarrow [(n, k_n, i_{1:k_n}), (\tau'_{1:k_n}, \phi'_{1:k_n}), (\tau_{1:n-k_n}^*, \phi_{1:n-k_n}^*)], \quad (6)$$

where we have implicitly used that  $H_n, V_n, h_n$  and  $v_n$  can be equivalently defined in terms of  $m_{1:n}$  or  $(n, k_n, i_{1:k_n})$ . The above one-to-one correspondence also allows us to use the same symbol for the conditional distributions  $\mu_n^\theta(m_{1:n} | \#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)$  and  $\mu_n^\theta(i_{1:k_n} | k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)$ . More details on the choice of  $\mu_n^\theta$  are given below.

The alternative extended target distribution – the target distribution of the SMC filter introduced in this subsection – is defined as  $\pi_n^\theta = \gamma_n^\theta / Z_n^\theta$ , where

$$\begin{aligned} \gamma_n^\theta(x_{1:n}) &:= \tilde{\gamma}_n^{r,\theta}(\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0) \mu_n^\theta(m_{1:n} | \#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0) \\ &\quad \times \prod_{j \in V_n} Q_{j,a}^\theta(\tau_j, \phi_j | \#H_{j+1}, \tau_{h_{j+1}}, \phi_{h_{j+1}}, \phi_0) \\ &= \tilde{\gamma}_n^{r,\theta}(k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0) \mu_n^\theta(i_{1:k_n} | k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0) \\ &\quad \times \left[ \prod_{j \in D_n} Q_{v_n(j),a}^\theta(\tau_j^*, \phi_j^* | \bar{k}_n(j), (\tau'_{1:\bar{k}_n(j)-1}, \tau_{j+1}^*), (\phi'_{1:\bar{k}_n(j)-1}, \phi_{j+1}^*), \phi_0) \right] \\ &\quad \times \prod_{j \in \llbracket n-k_n \rrbracket \setminus D_n} Q_{v_n(j),a}^\theta(\tau_j^*, \phi_j^* | \bar{k}_n(j), \tau'_{1:\bar{k}_n(j)}, \phi'_{1:\bar{k}_n(j)}, \phi_0). \end{aligned} \quad (7)$$

Here,  $D_n := \{j \in \llbracket n-k_n \rrbracket \mid v_n(j) < n-1 \text{ and } \forall l \in \llbracket k_n \rrbracket : v_n(j) + 2 \neq i_l\}$  whereas  $\bar{k}_n(j) := \sup\{p \in \llbracket n \rrbracket \mid i_p \leq v_n(j)\}$  is the number of birth moves up to step  $v_n(j)$ . In other words,  $j \in D_n$  if and only if the jumps sampled at steps  $v_n(j)$  and  $v_n(j) + 1$  are not kept in the ‘actual’ target distribution. Similarly,  $j \in \llbracket n-k_n \rrbracket \setminus D_n$  if and only if the jump sampled at step  $v_n(j)$  is not kept in the ‘actual’ target but the jump sampled at step  $v_n(j) + 1$  is kept and its jump time and size are given by  $(\tau'_{\bar{k}_n(j)}, \phi'_{\bar{k}_n(j)})$ . Finally,  $Q_{p,a}^\theta$  is defined as in the previous subsection.

The second line in the above definition shows that this extended target distribution admits  $\tilde{\pi}_n^{r,\theta}$  as a marginal. In addition, under this extended target distribution, the

transitions from  $X_{1:n-1}$  to  $X_n$  will be free of degenerate components assuming that  $q^\theta(\phi_j|\phi_{j-1}, \tau_j, \tau_{j-1})$  is non-degenerate.

As proposal kernels we use  $K_n^\theta(x_n|x_{1:n-1}) = \alpha_n(m_n|x_{n-1})K_{n,m_n}^\theta(x_n \setminus m_n|x_{1:n-1})$ , with birth and adjustment moves that are similar to the ones used in the original formulation of the SMC filter for PDPS, except that they do not share the degenerate components. A birth move,

$$K_{n,b}^\theta(\tau_n, \phi_n|x_{1:n-1}) := \rho_{n,b}^\theta(\tau_n|\#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0) \\ \times \eta_{n,b}^\theta(\phi_n|\tau_n, \#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0),$$

adds a new stopping time in  $(\tau_{n-1}, t_n]$  and samples a new jump size. Similarly,

$$K_{n,a}^\theta(\tau_n, \phi_n|x_{1:n-1}) := \rho_{n,a}^\theta(\tau_n|\#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0) \\ \times \eta_{n,a}^\theta(\phi_n|\tau_n, \#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0),$$

is an adjustment move which shifts the most recent stopping time to a new location in  $(\tau_{h_n(\#H_{n-1})}, t_n]$  and also samples a new value for the corresponding jump size. The kernels  $\rho_{n,m_n}^\theta$  and  $\eta_{n,m_n}^\theta$  are defined as in the previous subsection and we again define the forward ‘mixture weights’ by  $\alpha_n^\theta(a|x_{n-1}) := S^\theta(t_n, \tau_{n-1})$ .

The support of the target distribution in Equation 7 must be included in the support of the proposal distribution. To ensure this, we propose to set

$$\mu_n^\theta(di_{1:k_n}|k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0) := \nu_{k_n}(di_{k_n}|n+1, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0) \\ \times \delta_1(di_1) \prod_{j=2}^{k_n-1} \nu_j(di_j|i_{j+1}, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0).$$

Here,  $\nu_j(i_j|l, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)$  has support  $\{\tilde{s}(\tau'_{1:j-1}) + 1, \dots, l-1\}$ , where we recall that  $s(\tau) = \inf\{q \in \mathbb{N} \mid t_q \geq \tau\}$  and that  $\tilde{s}(\tau_{1:j}) = \sup\{s(\tau_{j-l+1}) + l - 1 \mid l \in \llbracket j \rrbracket\}$ .

If only restricted adjustment moves are used (see Subsection 4.2) then the support of  $\nu_j(i_j|l, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)$  must be limited to  $\{\tilde{s}(\tau'_{1:j}), \dots, l-1\}$ .

In the applications presented in Section 6 we employ such restricted adjustment moves. Consequently, we may take  $\nu_j(i_j|l, k_n, \tau'_{1:k_n}, \phi'_{1:k_n}, \phi_0)$  to be a geometric distribution truncated to  $\{\tilde{s}(\tau'_{1:j}), \dots, l-1\}$  or simply a uniform distribution on this set. Such a choice also ensures that the computational cost per SMC step of computing the importance weights remains constant.

The incremental weights,  $G_n^\theta(x_{1:n}) = \gamma_n^\theta(x_{1:n})/[\gamma_{n-1}^\theta(x_{1:n-1})K_n^\theta(x_n|x_{1:n-1})]$ , are computed as follows. For a birth move, i.e.  $m_n = b$ ,

$$G_n^\theta(x_{1:n}) = \frac{S^\theta(t_n, \tau_n)}{S^\theta(t_{n-1}, \tau_{n-1})} \frac{f^\theta(\tau_n|\tau_{n-1})q^\theta(\phi_n|\phi_{n-1}, \tau_n, \tau_{n-1})}{\alpha_n^\theta(b|x_{n-1})K_{n,b}^\theta(\tau_n, \phi_n|x_{1:n-1})} \\ \times \frac{\mu_n^\theta(m_{1:n}|\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)}{\mu_{n-1}^\theta(m_{1:n-1}|\#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0)} \frac{g^\theta(y_{[\tau_n, t_n]}|\tau_n, \phi_n)}{g^\theta(y_{[\tau_{n-1}, t_{n-1}]}|\tau_{n-1}, \phi_{n-1})}.$$

For an adjustment move, i.e.  $m_n = a$ ,

$$\begin{aligned}
G_n^\theta(x_{1:n}) &= \frac{S^\theta(t_n, \tau_n)}{S^\theta(t_{n-1}, \tau_{n-1})} \frac{Q_{n-1,a}^\theta(\tau_{n-1}, \phi_{n-1} | \#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)}{\alpha_n^\theta(a | x_{n-1}) K_{n,a}^\theta(\tau_n, \phi_n | x_{1:n-1})} \\
&\times \frac{\mu_n^\theta(m_{1:n} | \#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)}{\mu_{n-1}^\theta(m_{1:n-1} | \#H_{n-1}, \tau_{h_{n-1}}, \phi_{h_{n-1}}, \phi_0)} \\
&\times \frac{f^\theta(\tau_n | \tau_{h_n(\#H_{n-1})}) q^\theta(\phi_n | \phi_{h_n(\#H_{n-1})}, \tau_n, \tau_{h_n(\#H_{n-1})})}{f^\theta(\tau_{n-1} | \tau_{h_n(\#H_{n-1})}) q^\theta(\phi_{n-1} | \phi_{h_n(\#H_{n-1})}, \tau_{n-1}, \tau_{h_n(\#H_{n-1})})} \\
&\times \frac{g^\theta(y_{[\tau_{n-1} \wedge \tau_n, \tau_n]} | \tau_{h_n(\#H_{n-1})}, \phi_{h_n(\#H_{n-1})}) g^\theta(y_{[\tau_n, t_n]} | \tau_n, \phi_n)}{g^\theta(y_{[\tau_{n-1} \wedge \tau_{n-1}, \tau_{n-1}]} | \tau_{h_n(\#H_{n-1})}, \phi_{h_n(\#H_{n-1})}) g^\theta(y_{[\tau_{n-1}, t_{n-1}]} | \tau_{n-1}, \phi_{n-1})}.
\end{aligned}$$

Here, we use the convention  $g^\theta(y_{[s,t]} | \tau_j, \phi_j) := 1$  if  $s \geq t$ . To compute these weights, it is preferable to switch to the parametrisation from the right hand side in Equation 6.

For easier reference, we will hereafter refer to the algorithm presented in this subsection as the *reformulated sequential Monte Carlo (RSMC)* filter.

## 5 Static-parameter estimation using the particle Gibbs sampler

### 5.1 Particle Gibbs sampler

Bayesian approaches to static-parameter estimation based around the SMC approach have been developed by many authors (e.g. Neal, 2001; Chopin, 2002). More recently, Andrieu et al (2010); Chopin et al (2013) demonstrated that SMC methods can be incorporated within Markov chain Monte Carlo (MCMC) and SMC algorithms, respectively, to provide Bayesian estimates of static parameters and latent states in state-space models (and more generally).

In this section, we to conduct inference about the static parameters  $\Theta$  in PDPS, given observations  $y := y_{[0,t_P]}$ . To that end, we employ the particle Gibbs sampler which is one of the particle Markov chain Monte Carlo (PMCMC) methods for Bayesian parameter estimation that were developed by Andrieu et al (2010). These methods can be applied to a broad class of problems and hence we keep the notation in this section rather general.

Usually, the posterior distribution  $\pi_P(\theta) := p(\theta | y)$  is intractable. However, introducing latent variables  $X_{1:P}$  (i.e. the ‘states’ from Subsection 4.2 or 4.3, in the case of PDPS), we can at least evaluate  $\pi_P(\theta, x_{1:P}) = p(\theta) \gamma_P^\theta(x_{1:P}) / \mathcal{Z} = p(\theta, x_{1:P} | y)$  (the joint posterior distribution) up to its unknown normalising constant  $p(y) = \mathcal{Z} > 0$ . Here,  $p(\theta)$  denotes a prior density of  $\Theta$ . Recall that  $\gamma_P^\theta(x) := \pi_P^\theta(x_{1:P}) \mathcal{Z}_P^\theta = p(x_{1:P}, y | \theta)$ , so that  $\mathcal{Z}_P^\theta = p(y | \theta)$  is the unknown normalising constant of  $\pi_P^\theta(x_{1:P})$ .

A popular approach is to employ MCMC algorithms to approximate  $\pi_P(\theta, x_{1:P})$  which then yields an approximation of the marginal  $\pi_P(\theta)$ . Knowledge of the normalising constant  $\mathcal{Z}$  is not required. However, efficient proposal distributions for the latent variables  $X_{1:P}$  are needed within the MCMC scheme. The use of SMC algorithms to sample such latent variables is the motivation for PMCMC methods.

PMCMC methods can be seen as an extension of the pseudo-marginal approach by Beaumont (2003); Andrieu and Roberts (2009) which allows for the use of importance sampling within MCMC algorithms. Indeed, the justification of PMCMC methods in Andrieu

et al (2010) is based around a reinterpretation of SMC algorithms as importance sampling on a suitably extended space.

More precisely, PMCMC methods are MCMC methods that target a distribution which includes all the random variables generated by an SMC algorithm approximating  $\pi_P^\theta(x_{1:P})$ . These are all the particles  $\mathbf{X}_{1:P}$  and all the parent indices  $\mathbf{A}_{1:P-1}$  sampled over the course of the algorithm of which  $\mathbf{X}_n = (X_n^1, \dots, X_n^N)$  and  $\mathbf{A}_{n-1} = (A_{n-1}^1, \dots, A_{n-1}^N)$  are generated at the  $n$ th step.

For simplicity and clarity, we assume in this section that resampling takes place at every iteration in the sense that  $r^\theta(\cdot | \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}) \neq \delta_{1:N}$  for any  $(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1})$  and any  $n \in \llbracket P-1 \rrbracket$  and that the resampling scheme is *exchangeable* in the sense that any sample  $\mathbf{A}_n \sim r^\theta(\cdot | \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1})$  is exchangeable. As noted by Andrieu et al (2010), for resampling schemes that do not originally have this property, it can easily be ensured by permuting  $\mathbf{A}_n$  uniformly at random. However, we stress that we make this assumption only to simplify the presentation: essentially any unbiased resampling scheme can actually be used within the methods presented here (Lee et al, in prep.).

Assuming that  $P > 1$ , the distribution of  $(\mathbf{X}_{1:P}, \mathbf{A}_{1:P-1})$  is given by

$$\psi_P^\theta(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}) := \left[ \prod_{i=1}^N K_1^\theta(x_1^i) \right] \left[ \prod_{n=2}^P r^\theta(\mathbf{a}_{n-1} | \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}) \prod_{i=1}^N K_n^\theta(x_n^i | x_{1:n-1}^{a_{n-1}^i}) \right].$$

Given a ‘distinguished’ particle path  $X_{1:P}^* := (X_1^{B_1^*}, \dots, X_P^{B_P^*})$  with particle indices  $B_{1:P|P}^* := B_{1:P}^*$ , define

$$\mathbf{X}_{1:P}^{-*} := \mathbf{X}_{1:P} \setminus (X_1^{B_1^*}, \dots, X_P^{B_P^*}) \quad \text{and} \quad \mathbf{A}_{1:P-1}^{-*} := \mathbf{A}_{1:P-1} \setminus (A_1^{B_2^*}, \dots, A_{P-1}^{B_P^*}),$$

as well as

$$\begin{aligned} & \psi_P^\theta(\mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*} \| x_{1:P}^*, b_{1:P}^*) \\ & := \frac{\psi_P^\theta(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})}{K_1^\theta(x_1^*) \prod_{n=2}^P r^\theta(b_{n-1}^* | \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}) K_n^\theta(x_n^* | x_{1:n-1}^*)} \\ & = \left[ \prod_{i \in \llbracket N \rrbracket \setminus \{b_1^*\}} K_1^\theta(x_1^i) \right] \left[ \prod_{n=2}^P r^\theta(\mathbf{a}_{n-1}^{-*} | \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, a_{n-1}^{b_n^*}) \prod_{i \in \llbracket N \rrbracket \setminus \{b_n^*\}} K_n^\theta(x_n^i | x_{1:n-1}^{a_{n-1}^i}) \right], \end{aligned} \quad (8)$$

where  $r^\theta(\mathbf{a}_{n-1}^{-*} | \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}, a_{n-1}^{b_n^*}) = r^\theta(\mathbf{a}_{n-1} | \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}) / r^\theta(b_{n-1}^* | \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2})$  represents the *conditional* resampling distribution which at step  $n$ , ensures that particle  $b_n^*$  has particle  $b_{n-1}^*$  as its parent. Note that the denominator in Equation 8 is *not* the marginal density of  $(X_{1:P}^*, B_{1:P}^*)$  under  $\psi_P^\theta$ . Hence, the notation ‘ $\|$ ’ is used to distinguish the density in Equation 8 from the conditional density  $\psi_P^\theta(\mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*} | x_{1:P}^*, b_{1:P}^*)$ .

Particle MCMC methods target an extended distribution which is given by

$$\begin{aligned} & \bar{\pi}_P(\theta, \mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*}, x_{1:P}^*, b_{1:P}^*) \\ & := \frac{\pi_P(\theta, x_{1:P}^*)}{N^P} \psi_P^\theta(\mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*} \| x_{1:P}^*, b_{1:P}^*) \end{aligned} \quad (9)$$

$$= \frac{p(\theta)}{\mathcal{Z}} w_P^{b_P^*} \hat{\mathcal{Z}}_P^\theta(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}) \psi_P^\theta(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}). \quad (10)$$

Here, we recall that  $\pi_P(\theta, x_{1:P}) = p(\theta)\gamma_P^\theta(x_{1:P})/\mathcal{Z}$ . The factor  $N^P$  is a result of taking the marginal distribution of  $B_{1:P}^*$  to be uniform on  $\llbracket N \rrbracket^P$ . Equation 10 uses the reparametrisation  $(\mathbf{x}_{1:P}^*, \mathbf{a}_{1:P-1}^*, x_{1:P}^*, b_{1:P}^*) \longleftrightarrow (\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}, b_P^*)$ , where

$$B_n^* = A_n^{B_n^*+1} \quad (11)$$

for  $n \in \llbracket P-1 \rrbracket$ . The identity in Equation 10 also follows from Equation 3, as well as from the unbiasedness and exchangeability of the resampling schemes which imply  $r^\theta(i|\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}) = w_n^i = G_n^\theta(x_{1:n}^i) / \sum_{j=1}^N G_n^\theta(x_{1:n}^j)$ , and hence

$$\begin{aligned} & \frac{\gamma_P^\theta(x_{1:P}^*) N^{-P}}{K_1^\theta(x_1^*) \prod_{n=2}^P r^\theta(b_{n-1}^* | \mathbf{x}_{1:n-1}, \mathbf{a}_{1:n-2}) K_n^\theta(x_n^* | x_{1:n-1}^*)} \\ &= \frac{\prod_{n=1}^P G_n^\theta(x_{1:n}^*)}{N^P \prod_{n=1}^{P-1} w_n^{b_n^*}} = \frac{G_P^\theta(x_{1:P}^*)}{N^P} \prod_{n=1}^{P-1} \sum_{j=1}^N G_n^\theta(x_{1:n}^j) \\ &= w_P^{b_P^*} \widehat{\mathcal{Z}}_P^\theta(\mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}). \end{aligned} \quad (12)$$

The extended target distribution in Equation 9 clearly admits  $\pi_P(\theta, x_{1:P})$ , and hence  $\pi_P(\theta)$ , as a marginal. It can be targeted by exact MCMC algorithms. For instance, employing a Metropolis–Hastings algorithm leads to the *particle marginal Metropolis–Hastings (PMMH)* algorithm introduced by Andrieu et al (2010). It can also be targeted by a Gibbs sampler, leading to the *particle Gibbs* sampler introduced by Andrieu et al (2010) which is summarised in Algorithm 2. We use the convention that we always condition on the most recently sampled value of any parameter.

**Algorithm 2 (particle Gibbs sampler)** At each sweep, sample

- (1)  $(\Theta, \mathbf{X}_{1:P}^*, \mathbf{A}_{1:P-1}^*) \sim \bar{\pi}_P(d[\theta, \mathbf{x}_{1:P}^*, \mathbf{a}_{1:P-1}^*] | x_{1:P}^*, b_{1:P}^*)$ ,
- (2)  $B_P^* \sim \bar{\pi}_P(db_P^* | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ .

Equation 9 shows that Step 1 can be performed by first sampling  $\Theta \sim \pi_P(d\theta | x_{1:P}^*)$  and then sampling  $(\mathbf{X}_{1:P}^*, \mathbf{A}_{1:P-1}^*) \sim \psi_P^\theta(d\mathbf{x}_{1:P}^*, d\mathbf{a}_{1:P-1}^* | x_{1:P}^*, b_{1:P}^*)$ . The latter requires a so called ‘conditional’ SMC algorithm, i.e. an SMC algorithm targeting  $\pi_P^\theta$  which enforces that the  $b_n^*$ th particle at step  $n$  is set to  $x_n^*$ . This necessitates conditional versions of the resampling schemes mentioned in Section 3.1. These can be derived via Andrieu et al (2010, Appendix A) (see also Chopin and Singh, 2013; Lee et al, in prep.). Equation 10 shows that Step 2 amounts to selecting  $B_P^* = i$  with probability  $W_P^{\theta,i}$ . Note the change of variables involved in going from Step 1 to Step 2. Algorithm 2 is initialised by selecting initial values for  $\theta$  and then obtaining  $(X_{1:P}^*, B_{1:P}^*)$  from a standard SMC algorithm.

In the following, we focus on the particle Gibbs sampler for static-parameter estimation via PMCMC methods in PDPS because they can enjoy good mixing properties (Andrieu et al, 2013; Lindsten et al, 2014). In addition, in particle Gibbs samplers, a Metropolis–Hastings kernel updating the static parameters can be applied  $m$  times before performing the relatively computationally expensive SMC-based update of the states. The PMMH algorithm requires an SMC update every time a new static parameter is proposed. Finally, also in contrast to PMMH chains, mixing in particle Gibbs chains can be further improved by the variance-reduction techniques on which we elaborate in the next subsection.

## 5.2 Improving mixing of particle Gibbs chains

In this subsection, we describe variance-reduction techniques needed to make particle Gibbs samplers work in practice. In particular, at the end of this subsection, we propose a novel particle Gibbs step that allows us to rejuvenate the auxiliary variables generated as part of the RSMC algorithm from Subsection 4.3.

Note that the particle Gibbs sampler is not a standard Gibbs sampler in that a single sweep does not involve sampling from all full conditional distributions relative to  $\bar{\pi}_P$ . More precisely, a single sweep does not sample new values for the particles and parent indices associated with the distinguished path from the previous iteration, i.e. for

$$X_1^{B_1^*}, \dots, X_P^{B_P^*}, A_1^{B_2^*}, \dots, A_{P-1}^{B_P^*}.$$

Nonetheless, the distinguished particle path and the corresponding particle indices can still be changed by a single sweep. This is because a different value for  $B_P^*$  sampled in Step 2 changes the particle indices  $B_{1:P-1}^*$  and thus the particles  $X_{1:P}^*$  if we go back to the parametrisation in Step 1, due to the recursive relationship given in Equation 11.

The particle Gibbs sampler is justified by the idea that eventually, over sufficiently many sweeps, all the components of  $\bar{\pi}_P^\theta$  are updated. However, this idea crucially relies on the ability to alleviate sample impoverishment. If sample impoverishment is severe, all step- $P$  particles will share a common ancestor which, by construction, is a particle from the distinguished path of the previous iteration. Thus, the variables associated with the beginning of the distinguished trajectory are not updated.

Two methods for variance reduction in particle Gibbs samplers are summarised below. An extensive review can also be found in Lindsten and Schön (2013). These methods can only be used if the conditional distribution of  $X_{1:n}$  under  $\pi_P^\theta$  is non-degenerate and if the associated conditional density can be evaluated. They are therefore not applicable to state-space models with degenerate or intractable transitions nor to the ‘original’ SMC filter from Subsection 4.2. For inference in PDPS, we instead employ the RSMC filter presented in Subsection 4.3 which does not have such degeneracy as long as the kernels  $q^\theta(\phi_j | \phi_{j-1}, \tau_j, \tau_{j-1})$  are non-degenerate.

*Backward sampling.* Recall that the particle Gibbs sampler samples the step- $P$  index of the conditioning path,  $B_P^*$ , and then recursively determines the particle indices  $B_{1:P-1}^*$  via Equation 11. Therefore one way of improving mixing of the Gibbs chain suggested by Whiteley (2010); Whiteley et al (2010) is to add  $P - 1$  steps to the Gibbs sweep which explicitly sample new values for  $B_{1:P-1}^*$ . Variations of this approach are further analysed by Lindsten and Schön (2012); Chopin and Singh (2013).

**Algorithm 3 (particle Gibbs with backward sampling)** At each sweep,

- (1) sample  $(\theta, \mathbf{X}_{1:P}^-, \mathbf{A}_{1:P-1}^-) \sim \bar{\pi}_P(d[\theta, \mathbf{x}_{1:P}^-, \mathbf{a}_{1:P-1}^-] | x_{1:P}^*, b_{1:P}^*),$
- (2) sample  $B_P^* \sim \bar{\pi}_P(db_P^* | \theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1}),$
- (3) for  $n = P - 1, \dots, 1,$  sample  $B_n^* \sim \bar{\pi}_P(db_n^* | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x_{n+1:P}^*, b_{n+1:P}^*).$

As shown by Whiteley (2010), the conditional distributions in Step 3 take a simple form for state-space models. However, all random variables generated by the conditional

SMC algorithm in Step 1 of Algorithm 3 need be stored to perform backward sampling in Step 3.

Write  $(i_{n+1}, \dots, i_P) := (b_{n+1}^*, \dots, b_P^*)$  and  $i_p := a_p^{i_p+1}$ , for  $p \in \llbracket n \rrbracket$ . Step 3 can then be performed by setting  $B_n^* := i_n$  with probability proportional to

$$\begin{aligned}
& \bar{\pi}_P(i_n | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x_{n+1:P}^*, b_{n+1:P}^*) \\
& \propto \bar{\pi}_P(x_1^{i_1}, \dots, x_P^{i_P}, i_{1:P} | \theta, \mathbf{x}_{1:n} \setminus (x_1^{i_1}, \dots, x_n^{i_n}), \mathbf{a}_{1:n-1} \setminus (a_1^{i_2}, \dots, a_{n-1}^{i_n})) \\
& \propto \frac{\gamma_P^\theta(x_{1:P}^{i_P}) \psi_n^\theta(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1})}{K_1^\theta(x_1^{i_1}) \prod_{p=2}^n r^\theta(i_{p-1} | \mathbf{x}_{1:p-1}, \mathbf{a}_{1:p-2}) K_p^\theta(x_p^{i_p} | x_{1:p-1}^{i_{p-1}})} \\
& \propto w_n^{\theta, i_n} \hat{Z}_n^\theta(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}) \psi_n^\theta(\mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}) \gamma_P^\theta(x_{1:P}^{i_P}) / \gamma_n^\theta(x_{1:n}^{i_n}) \\
& \propto w_n^{\theta, i_n} \gamma_P^\theta(x_{1:P}^{i_P}) / \gamma_n^\theta(x_{1:n}^{i_n}). \tag{13}
\end{aligned}$$

The first and second step in the above equation follow from Equation 9 while the third step in the above equation was derived in the same way as Equation 12.

Note that Step 3 does not entail sampling from full conditional distributions of  $\bar{\pi}_P$ . Such schemes are formally justified as an application of the *partially-collapsed-Gibbs-sampler* framework by Van Dyk and Park (2008) to the particle Gibbs sampler from the previous subsection. More precisely, Steps 3 could formally be interpreted as if the algorithm sampled the variables  $(B_n^*, [\mathbf{X}_{n+1:P}^*, \mathbf{A}_{n:P-1}^*])$  from their respective full conditional distribution relative to  $\bar{\pi}_P$ . Here, the quantities in square brackets indicate random variables that can be discarded ('trimmed') right away as they are not conditioned upon in subsequent steps. Due to the structure of  $\bar{\pi}_P$ , they actually do not need to be sampled in the first place.

*Ancestor sampling.* An alternate variance-reduction method for particle Gibbs samplers was proposed by Lindsten et al (2012). Termed *ancestor sampling*, it achieves an update of the ancestral lineage of the distinguished path using a single forward pass.

Ancestor sampling is outlined in Algorithm 4. Again, we use the convention that we always condition on the most recently sampled value of any parameter.

**Algorithm 4 (particle Gibbs with ancestor sampling)** At each sweep,

- (1) sample  $\theta \sim \pi_P(d\theta | x_{1:P}^*)$ ,
- (2) for  $n = 1, \dots, P$ , sample
  - (i)  $(\mathbf{X}_n^-, \mathbf{A}_{n-1}^-) \sim \bar{\pi}_P(d[\mathbf{x}_n^-, \mathbf{a}_{n-1}^-] | \theta, \mathbf{x}_{1:n-1}^-, \mathbf{a}_{1:n-2}^-, x_{1:P}^*, b_{1:P}^*)$ ,
  - (ii)  $B_n^* \sim \bar{\pi}_P(db_n^* | \theta, \mathbf{x}_{1:n}, \mathbf{a}_{1:n-1}, x_{n+1:P}^*, b_{n+1:P}^*)$ .

The algorithm is initialised identically to Algorithm 2. Step 2i entails sampling  $(\mathbf{X}_n^-, \mathbf{A}_{n-1}^-)$  via the  $n$ th step of a conditional SMC algorithm with distinguished path  $(x_{1:P}^*, b_{1:P}^*)$  and step- $n$  target distribution  $\pi_n^\theta$ . Step 2ii can be performed by setting  $B_n^* := i_n$  with probability proportional to the expression in Equation 13.

Again, Steps 1 and 2 do not entail sampling from full conditional distributions of  $\bar{\pi}_P$  and are justified as a partially collapsed Gibbs steps. More precisely, Steps 1 could formally be interpreted as sampling the variables  $(\theta, [\mathbf{X}_{1:P}^-, \mathbf{A}_{1:P-1}^-])$ , Step 2i as sampling

the variables  $(\mathbf{X}_n^{-\star}, [\mathbf{X}_{n+1:P}^{-\star}], \mathbf{A}_{n-1}^{-\star}, [\mathbf{A}_{n:P-1}^{-\star}])$  and Step 2ii as sampling the variables  $(B_n^\star, [\mathbf{X}_{n+1:P}^{-\star}, \mathbf{A}_{n:P-1}^{-\star}])$  from their respective full conditional distributions relative to  $\bar{\pi}_P$ . Again, the quantities in square brackets indicate random variables that need not actually be sampled.

Below, we derive the probabilities  $G_{n|P}^\theta(x_{1:P}) := \gamma_P^\theta(x_{1:P})/\gamma_P^\theta(x_{1:n})$  needed for the computation of the backward or ancestor sampling weights in Equation 13 for the vRPF and the RSMC filter.

For the vRPF, using the notation from Subsection 4.1, assume that  $n < P$  and let  $\mu(n) := \inf\{m \in \{n+1, \dots, P\} \mid k_m > 0\}$ . If  $\sum_{p=n+1}^P k_p = 0$  and  $k_n > 0$ , then, recalling that  $t_P = T$ ,

$$G_{n|P}^\theta(x_{1:P}) \propto S^\theta(T, \tau_{n,k_n}) g^\theta(y_{(t_n, T]} | \phi_{n,k_n}, \tau_{n,k_n}) / S^\theta(t_n, \tau_{n,k_n}).$$

If  $\sum_{p=n+1}^P k_p > 0$  and  $k_n > 0$  then

$$G_{n|P}^\theta(x_{1:P}) \propto g^\theta(y_{(t_n, \tau_{\mu(n),1})} | \phi_{n,k_n}, \tau_{n,k_n}) f^\theta(\tau_{\mu(n),1} | \tau_{n,k_n}) \\ \times q^\theta(\phi_{\mu(n),1} | \phi_{n,k_n}, \tau_{\mu(n),1}, \tau_{n,k_n}) / S^\theta(t_n, \tau_{n,k_n}).$$

If  $k_n = 0$  then  $(\tau_{n,k_n}, \phi_{n,k_n})$  in the above equations is replaced by  $(\tau_{v(n),k_{v(n)}}, \phi_{v(n),k_{v(n)}})$ .

For the RSMC filter, using the notation from Subsection 4.3 and assuming  $n < P$ , let  $n_b := \inf\{p \in \{n+2, \dots, P\} \mid m_p = b\}$  denote the iteration with the first birth move after step  $n+1$ , with the convention that  $n_b := P+1$  if there is no further jump at steps  $n+2, \dots, P$ . If  $m_{n+1} = b$ ,

$$G_{n|P}^\theta(x_{1:P}) \propto \frac{\mu_P^\theta(m_{1:P} | \#H_P, \tau_{h_P}, \phi_{h_P}, \phi_0)}{\mu_n^\theta(m_{1:n} | \#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)} \frac{f^\theta(\tau_{n_b-1} | \tau_n) q^\theta(\phi_{n_b-1} | \phi_n, \tau_{n_b-1}, \tau_n)}{S^\theta(t_n, \tau_n)} \\ \times \frac{g^\theta(y_{(t_n, \tau_{n_b-1})} | \tau_n, \phi_n)}{g^\theta(y_{[\tau_{n_b-1}, t_n]} | \tau_n, \phi_n)} \prod_{j=n+2}^{n_b-1} Q_{j-1,a}^\theta(\tau_{j-1}, \phi_{j-1} | \#H_j, \tau_{h_j}, \phi_{h_j}, \phi_0).$$

If  $m_{n+1} = a$ ,

$$G_{n|P}^\theta(x_{1:P}) \propto \frac{\mu_P^\theta(m_{1:P} | \#H_P, \tau_{h_P}, \phi_{h_P}, \phi_0)}{\mu_n^\theta(m_{1:n} | \#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0)} \frac{f^\theta(\tau_{n_b-1} | \tau_{h_n(\#H_n-1)})}{S^\theta(t_n, \tau_n) f^\theta(\tau_n | \tau_{h_n(\#H_n-1)})} \\ \times \frac{q^\theta(\phi_{n_b-1} | \phi_{h_n(\#H_n-1)}, \tau_{n_b-1}, \tau_{h_n(\#H_n-1)})}{q^\theta(\phi_n | \phi_{h_n(\#H_n-1)}, \tau_n, \tau_{h_n(\#H_n-1)})} \\ \times \frac{g^\theta(y_{[\tau_n \wedge \tau_{n_b-1}, \tau_{n_b-1}] | \tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)})}{g^\theta(y_{[\tau_n \wedge \tau_{n_b-1}, \tau_n] | \tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)})} g^\theta(y_{[\tau_n, t_n]} | \tau_n, \phi_n) \\ \times \prod_{j=n+1}^{n_b-1} Q_{j-1,a}^\theta(\tau_{j-1}, \phi_{j-1} | \#H_j, \tau_{h_j}, \phi_{h_j}, \phi_0).$$

Again, we use the convention that  $g^\theta(y_I | \tau_j, \phi_j) := 1$ , if  $I = \emptyset$ . To compute these weights, it is again preferable to switch to the parametrisation from the right hand side in Equation 6.

*Auxiliary variable rejuvenation.* The distribution  $\pi_P(\theta, x_{1:P})$  marginally targeted by the particle Gibbs sampler is sometimes itself an ‘artificially’ extended distribution in the sense that it can be factorised as

$$\pi_P(\theta, x_{1:P}) = \tilde{\pi}_P(\theta, \tilde{x})L^\theta(\tilde{z}|\tilde{x}).$$

Here,  $\tilde{\pi}_P(\theta, \tilde{x})$  is the marginal distribution of random variables that are actually of interest while  $L^\theta(\tilde{z}|\tilde{x})$  is the conditional distribution of some auxiliary variables  $\tilde{Z}$ . We assume that  $X_{1:P}$  can be partitioned into  $\tilde{X}$  and  $\tilde{Z}$ . Further, we assume that we can sample from  $L^\theta(\tilde{z}|\tilde{x})$ .

Such a setting arises whenever auxiliary-variable-based SMC schemes, such as SMC samplers with non-trivial backward kernels, are used within a particle Gibbs algorithm. For instance when using the RSMC filter, we are actually only interested in the distribution of  $\tilde{X} = (\#H_P, \tau_{h_P}, \phi_{h_P}, \phi_0)$  and  $\Theta$ , but for algorithmic purposes, the particle Gibbs sampler targets a distribution that also includes  $\tilde{Z} = (M_{1:P}, \tau_{v_P}, \phi_{v_P})$ . Conditioning on these auxiliary variables when sampling  $\Theta$  in a particle Gibbs sweep can become computationally expensive and can induce slow mixing as soon as  $L^\theta$  depends on  $\theta$ .

We propose an additional particle Gibbs step that overcomes these potential difficulties. It is summarised in Algorithm 5, where  $x_{1:P}^* = (\tilde{x}^*, \tilde{z}^*)$  (with some abuse of notation pertaining to the ordering within in both vectors) is the ‘distinguished’ path.

**Algorithm 5 (particle Gibbs w/ auxiliary variable rejuvenation)** At each sweep,

- (1) sample  $\theta \sim \tilde{\pi}_P(d\theta|\tilde{x}^*)$ ,
- (2) sample  $\tilde{z}^* \sim L^\theta(d\tilde{z}^*|\tilde{x}^*)$  and set  $x_{1:P}^* = (\tilde{x}^*, \tilde{z}^*)$ ,
- (3) sample  $(\mathbf{X}_{1:P}^{-*}, \mathbf{A}_{1:P-1}^{-*}) \sim \psi_P^\theta(d[\mathbf{x}_{1:P}^{-*}, \mathbf{a}_{1:P-1}^{-*}]||x_{1:P}^*, b_{1:P}^*)$ ,
- (4) sample  $B_P^* \sim \tilde{\pi}_P(db_P^*|\theta, \mathbf{x}_{1:P}, \mathbf{a}_{1:P-1})$ .

This algorithm is again justified as a partially collapsed Gibbs sampler. Of course, it can be combined with backward sampling by including Step 3 from Algorithm 3 at the end of the Gibbs sweep. Alternatively, ancestor sampling may be used. Steps 3 and 4 are then replaced by Step 2 from Algorithm 4.

Algorithm 5 comes at little or no extra computational cost. It can even offer computational savings compared to a standard particle Gibbs scheme, e.g. when each Gibbs sweep updates the static parameters using the  $m$ -fold convolution of a Metropolis–Hastings kernel (as is often done in practice since static-parameter updates are relatively computationally inexpensive compared to state updates). Algorithm 5 then avoids  $m$  evaluations of  $L^\theta$  at the cost of generating one sample from  $L^\theta$ .

We conclude this section by detailing the implementation of Algorithm 5 for particle Gibbs samplers using the RSMC filter from Subsection 4.3: Step 1 involves sampling  $\Theta$  from its full conditional distribution under  $p(\theta)\tilde{\pi}_P^\theta(\#H_P, \tau_{h_P}, \phi_{h_P}, \phi_0)$ , where  $p(\theta)$  denotes some prior distribution and the second term is defined in Equation 1. Step 2 is performed by sampling new values for  $\tilde{Z} = (M_{1:P}, \tau_{v_P}, \phi_{v_P})$  from

$$L^\theta(\tilde{z}|\tilde{x}) = \mu_n^\theta(m_{1:n}|\#H_n, \tau_{h_n}, \phi_{h_n}, \phi_0) \prod_{j \in V_n} Q_{j,a}^\theta(\tau_j, \phi_j|\#H_{j+1}, \tau_{h_{j+1}}, \phi_{h_{j+1}}, \phi_0).$$

The interaction between Step 2 of Algorithm 5, ancestor sampling, and the structure of the distribution targeted by the RSMC filter is explored in Subsection 6.3.

## 6 Simulation study

### 6.1 General setup

In this section, we apply the particle Gibbs sampler with ancestor sampling and auxiliary variable rejuvenation – based on the RSMC filter from Subsection 4.3 – to the elementary change-point model (Example I) from Subsection 2.2 and to the shot-noise-Cox-process model (Example II) from Subsection 2.3. For easier reference, this algorithm is hereafter called *RSMC-based particle Gibbs (RSMC-PG)* sampler. We compare its performance with that of a *VRPF-based particle Gibbs (VRPF-PG)* sampler also using ancestor sampling and with a RJMCMC algorithm.

For the RSMC filter, a birth move at step  $n$  samples a new jump time  $\tau_n$  uniformly in  $(\tau_{h_n(\#H_n-1)}, t_n]$ . The jump size,  $\phi_n$ , is then sampled from its full conditional posterior distribution given the observations up to time  $t_n \wedge 4\mu_\tau$ , with  $\mu_\tau$  being the prior mean inter-jump time. We use restricted jump-time adjustment moves, i.e. we use Gaussian kernels with variance  $10^{-4}$ , centred around  $\tau_{n-1}$  and truncated to  $((\tau_{h_n(\#H_n-1)} \vee t_{s_{n-1}-1}), t_{s_{n-1}}]$  where  $s_{n-1} := s(\tau_{n-1})$ . Gaussian kernels with this variance, centred around  $\phi_{n-1}$ , are also used for the jump-size adjustments. For Example II, they are truncated to  $(F^\theta(\tau_n, \tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)}), \infty)$ . Likewise, the kernel  $Q_{n-1,a}^\theta$  is a product of independent Gaussians, each with variance  $10^{-4}$ . Its first component is centred around  $\tau_n$  and truncated to  $((\tau_{h_n(\#H_n-1)} \vee t_{s_n-1}), t_{s_n}]$ , where  $s_n := s(\tau_n)$ . The second component is centred around  $\phi_n$  and in the shot-noise-Cox-process example, its support is restricted to  $(F^\theta(\tau_{n-1}, \tau_{h_n(\#H_n-1)}, \phi_{h_n(\#H_n-1)}), \infty)$ . Finally, the conditional distribution of  $i_{1:k_n}$  is taken to be a truncated geometric distribution with parameter 0.3 and with support (for restricted adjustment moves) as given in Subsection 4.3.

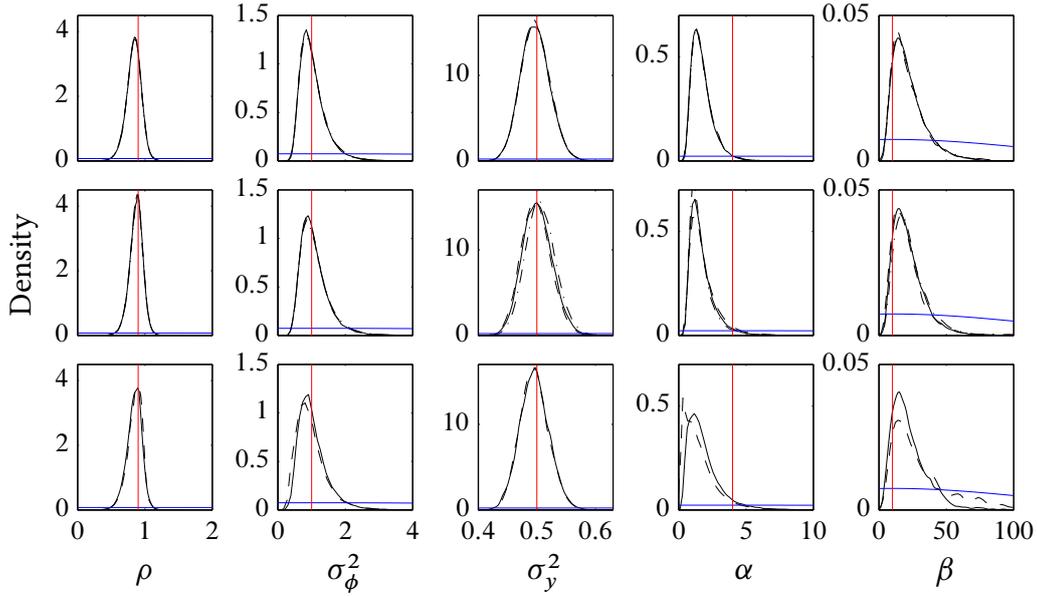
For the VRPF, we propose the number of jumps in  $(t_{n-1}, t_n]$  from a Poisson distribution with mean  $(t_n - t_{n-1})/\mu_\tau$ . The jump times are subsequently sampled independently from a uniform distribution on  $(t_{n-1}, t_n]$  and are then ordered. The corresponding jump sizes are proposed from their full conditional time- $t_n$  posterior distribution.

The step size in both SMC algorithms is set to  $t_n - t_{n-1} = 10$ . Throughout, we use (conditional) systematic resampling and resample only when the effective sample size falls below  $N_{\text{ESS}} := 0.8N$ . The moves that update jumps in the RJMCMC algorithm are those used in Centanni and Minozzo (2006b) (except that for Example I, jump sizes are always sampled from their full conditional posterior distributions).

Within all three algorithms, a new value for the vector of static parameters is proposed using the  $m$ -fold convolution of a Gaussian random-walk Metropolis–Hastings kernel with the same covariance matrix across algorithms. More sophisticated updates for the static parameters could be constructed but we choose not to do so since this paper’s focus is on updating the time-varying parameters.

In what follows, a single ‘iteration’ or ‘sweep’ of one of these algorithms refers to first updating the static parameters (followed by the auxiliary-variable rejuvenation step for the RSMC-PG algorithm) and then updating the jumps using either a conditional SMC update or  $l$  RJMCMC updates. For Example I, we used  $m = l = 500$  and for Example II, we used  $m = l = 1,000$ .

Initial values for the static parameters are sampled from the prior. For Example II, we then divide the first two static parameters by 100 to avoid starting in a region with a



**Figure 3:** Kernel density estimates for the marginal posterior densities of the static parameters in Example I. *Top row:* RSMC-PG algorithm with 100 particles (solid line), 50 particles (dashed line), 25 particles (dash-dotted line). *Middle row:* VRPF-PG sampler with 100 particles (solid line), 50 particles (dashed line), 25 particles (dash-dotted line). *Bottom row:* two RJMCMC chains. Red lines indicate the true parameters; blue lines show the prior densities.

very large number of jumps. This is done to reduce the computational cost for the first iterations in the VRPF-PG and RJMCMC algorithms and also because we have observed that the RSMC-PG sampler can get stuck if started in a region with close to  $P$  jumps. A possible explanation of the latter phenomenon is provided in Subsection 6.3.

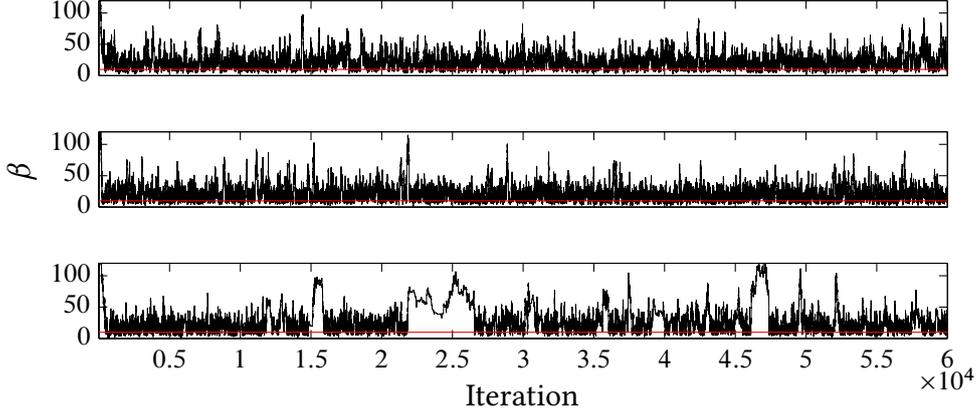
The algorithms are implemented in Matlab on a single 2.66 GHz Intel ‘Westmere’ core using 4 gigabytes of RAM. In each case, the presented results are based on 60,000 iterations of which the first 10,000 are discarded as burn-in.

## 6.2 Simulation results

*Elementary change-point model.* For Example I, we used the simulated data set shown in Figure 1. We chose a Gaussian prior on the static parameters, with covariance matrix  $\text{diag}(10^2, 10^2, 10, 10^3, 10^4)$  and truncated to  $\mathbb{R} \times (0, \infty)^4$ .

As shown in Figure 3, all three algorithms yielded comparable estimates for the marginal posterior distributions of the static parameters even when using only 25 particles. The bivariate correlation structure and sample autocorrelations were also similar but are omitted due to limited space. However, we encountered RJMCMC chains that seemed to get stuck in local modes for a considerable number of iterations. Such a chain is represented by the dashed line in the bottom row of Figure 3 and the corresponding trace plot for the parameter  $\beta$  is shown in Figure 4. We did not encounter such a behaviour in any of the particle Gibbs samplers.

The computing time for the auxiliary-variable rejuvenation and conditional SMC



**Figure 4:** Trace plots for the scale-parameter estimates in Example I. *Top:* RSMC-PG sampler with 100 particles. *Middle:* VRPF-PG sampler with 100 particles. *Bottom:* RJMCMC sampler. Red lines indicate the true parameters.

update (with ancestor sampling) in the RSMC-PG sampler was around 1.6 seconds on average, the conditional SMC update (with ancestor sampling) for the VRPF-PG sampler took around 2.5 seconds and  $l = 500$  individual moves for the RJMCMC algorithm took around 2 seconds. The difference can partially be explained by the fact that the RSMC sampler is more amenable to code vectorisation than the VRPF because at each SMC step, it generates the same number of random variables for each particle.

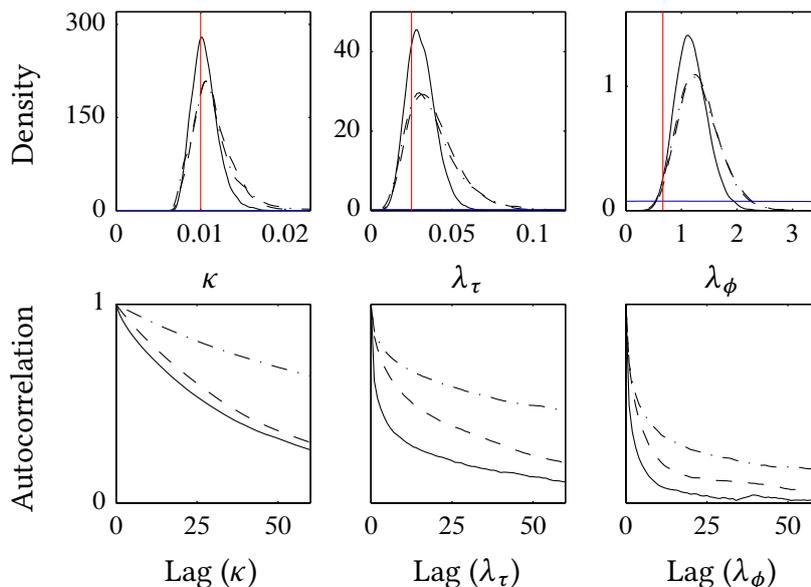
*Shot-noise-Cox-process model.* For Example II, we used the simulated data set shown in Figure 2. We chose a Gaussian prior for the vector of static parameters, with covariance matrix  $\text{diag}(10, 10, 10^2)$  and truncated to  $(0, \infty)^3$ . For the static-parameter updates we switched to a non-centred parametrisation of the jump sizes to improve mixing of the decay parameter  $\kappa$ .

As shown in Figure 5, the estimated marginal posterior densities from all three algorithms have similar modes. However, those obtained from the RSMC-PG sampler are more concentrated. This difference is possibly due to the approximation described in Subsection 4.2 which restricts the number of jumps in any particular interval. In this model, it produces visibly different results because the exponential prior on the interjump times allows large numbers of jumps to be placed close to each other with non-negligible probability. Thus, the posterior distribution of this model has tail regions with large numbers of jumps which the RSMC-PG algorithm rarely enters. This could also contribute to the differences in the autocorrelations in Figure 5. Note that the effect of this approximation can be reduced by decreasing the step size  $t_n - t_{n-1}$ .

### 6.3 Rôle of the auxiliary-variable rejuvenation step

In our simulations, the extra Gibbs step from Algorithm 5 appeared to be crucial to the performance of the RSMC-PG sampler: without it, the algorithm could get stuck in local modes. Below, we give a possible explanation of this phenomenon.

Let  $x_{n+1}^* = (m_{n+1}^*, \tau_{n+1}^*, \phi_{n+1}^*)$  denote the  $(n + 1)$ th component of the distinguished path and let  $x_n^i = (m_n^i, \tau_n^i, \phi_n^i)$  denote the  $i$ th particle at step  $n$ . Then at the  $n$ th step of



**Figure 5:** Static-parameter estimates for Example II. Based on the RSMC-PG algorithm with 100 particles (solid line), the VRPF-PG sampler with 100 particles (dashed line), and an RJ-MCMC sampler (dash-dotted line). *Top row:* kernel density estimates of the marginal posterior densities. Red lines indicate true parameters; blue lines represent priors. *Bottom row:* autocorrelations.

the conditional SMC algorithm, note the following (for any  $i \neq b_n^*$ ).

- (1) If  $m_{n+1}^* = a$ , and if we employ local adjustment moves, the distance  $|\tau_{n+1}^* - \tau_n^i|$  tends to be so big that the  $i$ th ancestor sampling weight is close to zero.
- (2) If  $m_{n+1}^* = b$ , the  $i$ th ancestor sampling weight is only non-zero if  $\tau_n^i < \tau_{n+1}^*$ .

We conjecture that with only local adjustment moves and without Step 2 of Algorithm 5, the algorithm can get stuck because ancestor sampling is relatively ineffective: it rarely changes the distinguished path in Situation 1 and mixing thus relies on replacing the distinguished path in Situation 2. Here, however, if  $\tau_{n+1}^*$  is much smaller than  $t_n$ , the most recent jump in all other particle paths is likely to be located in the interval  $(\tau_{n+1}^*, t_n]$  so that they have zero ancestor sampling weights.

Step 2 of Algorithm 5 circumvents this problem because it can change the SMC step of the birth move which is associated with a particular jump.

This reasoning might also explain why we observed that the RSMC-PG sampler could get stuck in Example II when it was initialised in a region with  $P$  jumps: if the distinguished path has only birth moves then Step 2 of Algorithm 5 cannot change the SMC step of the birth move associated with any jump.

## 7 Summary

In this paper, we have demonstrated that particle Gibbs samplers can be applied to piecewise deterministic processes and have presented a number of methodological developments in doing so. Numerical studies provide a comparative illustration of the performance of the proposed methods.

## References

- Andrieu C, Roberts GO (2009) The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics* 37(2):697–725
- Andrieu C, Doucet A, Holenstein R (2010) Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72(3):269–342, with discussion
- Andrieu C, Lee A, Vihola M (2013) Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *arXiv preprint arXiv:13126432*
- Beaumont MA (2003) Estimation of population growth or decline in genetically monitored populations. *Genetics* 164(3):1139–1160
- Bunch P, Godsill S (2013) Particle smoothing algorithms for variable rate models. *IEEE Transactions on Signal Processing* 61(7):1663–1675
- Carlin BP, Chib S (1995) Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B (Statistical Methodology)* pp 473–484
- Carpenter J, Clifford P, Fearnhead P (1999) An improved particle filter for nonlinear problems. *IEE Proceedings, Radar and Sonar Navigation* 146(1):2–7
- Centanni S, Minozzo M (2006a) Estimation and filtering by reversible jump MCMC for a doubly stochastic Poisson model for ultra-high-frequency financial data. *Statistical Modelling* 6(2):97–118
- Centanni S, Minozzo M (2006b) A Monte Carlo approach to filtering for a class of marked doubly stochastic Poisson processes. *Journal of the American Statistical Association* 101(476):1582–1597
- Chopin N (2002) A sequential particle filter method for static models. *Biometrika* 89(3):539–551
- Chopin N, Singh SS (2013) On the particle gibbs sampler. *arXiv preprint arXiv:13041887v1*
- Chopin N, Jacob PE, Papaspiliopoulos O (2013) SMC<sup>2</sup>: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 75(3):397–426
- Dassios A, Jang J (2003) Pricing of catastrophe reinsurance and derivatives using the Cox process with shot noise intensity. *Finance and Stochastics* 7(1):73–95
- Del Moral P (1995) Nonlinear filtering using random particles. *Theory of Probability and Its Applications* 40(4):690–701
- Del Moral P (1996) Nonlinear filtering: interacting particle solution. *Markov Processes and Related Fields* 2(4):555–580
- Del Moral P (2004) *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York
- Del Moral P, Doucet A, Jasra A (2006) Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(3):411–436
- Del Moral P, Doucet A, Jasra A (2007) Sequential Monte Carlo for Bayesian computation. In: Bernardo J, Bayarri M, Berger J, Dawid A, Heckerman D, Smith A, West M (eds) *Bayesian Statistics 8*, Oxford University Press, USA, pp 115–148
- Del Moral P, Doucet A, Jasra A (2012) On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli* 18(1):252–278

- Douc R, Cappé O, Moulines E (2005) Comparison of resampling schemes for particle filtering. In: *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA 2005)*, Zagreb, Croatia, IEEE, pp 64–69
- Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: Fifteen years later. In: Crisan D, Rozovskii B (eds) *The Oxford Handbook of Nonlinear Filtering*, Oxford Handbooks, Oxford University Press, Oxford, chap 24, pp 656–704
- Doucet A, Freitas Nd, Gordon N (eds) (2001) *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science, Springer, New York
- Godsill S, Vermaak J (2004) Models and algorithms for tracking using trans-dimensional sequential Monte Carlo. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004 (ICASSP'04)*, vol 3, pp 976–969
- Godsill SJ (2001) On the relationship between Markov chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics* 10(2):230–248
- Gordon NJ, Salmond DJ, Smith AFM (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F, Radar and Signal Processing* 140(2):107–113
- Green PJ (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82(4):711–732
- Johansen AM, Doucet A (2008) A note on auxiliary particle filters. *Statistics & Probability Letters* 78(12):1498–1504
- Kantas N, Doucet A, Singh SS, Maciejowski JM (2009) An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In: *15th IFAC Symposium on System Identification*, pp 774–785
- Kitagawa G (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* 5(1):1–25
- Kitagawa G (1998) A self-organizing state-space model. *Journal of the American Statistical Association* 93(443):1203–1215
- Kong A, Liu JS, Wong WH (1994) Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association* 89(425):278–288
- Lee A, Murray LM, Johansen AM (in prep.) Resampling in conditional SMC algorithms
- Lindsten F, Schön TB (2013) Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning* 6(1):1–143
- Lindsten F, Schön TB (2012) On the use of backward simulation in the particle Gibbs sampler. In: *Proceedings of the 37th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan
- Lindsten F, Jordan MI, Schön TB (2012) Ancestor sampling for particle Gibbs. In: *Proceedings of the 2012 Conference on Neural Information Processing Systems (NIPS)*, Lake Tahoe, NV, USA
- Lindsten F, Douc R, Moulines E (2014) Uniform ergodicity of the particle Gibbs sampler. *arXiv preprint arXiv:14010683*
- Liu JS, Chen R (1998) Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association* 93(443):1032–1044

- Martin JS, Jasra A, McCoy E (2012) Inference for a class of partially observed point process models. *Annals of the Institute of Statistical Mathematics* pp 1–25
- Neal RM (2001) Annealed importance sampling. *Statistics and Computing* 11(2):125–139
- Pitt MK, Shephard N (1999) Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association* 94(446):590–599
- Poyiadjis G, Doucet A, Singh SS (2011) Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika* 98(1):65–80
- Rao V, Teh YW (2012) Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. *arXiv preprint arXiv:12023760v1*
- Van Dyk DA, Park T (2008) Partially collapsed Gibbs samplers. *Journal of the American Statistical Association* 103(482):790–796
- Whiteley N (2010) Contribution to the discussion on ‘Particle Markov chain Monte Carlo methods’ by Andrieu, C., Doucet, A., and Holenstein, R. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72(3):306–307
- Whiteley N, Andrieu C, Doucet A (2010) Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods. Bristol University Statistics Research Report 10:04
- Whiteley N, Johansen AM, Godsill S (2011) Monte Carlo filtering of piecewise deterministic processes. *Journal of Computational and Graphical Statistics* 20(1):119–139