A Unified Framework for Sequential Parameter Learning with Regularization in State Space Models

Uriel Moreira Silva* [1] Felipe Carvalho Álvares da Silva [2]
Luiz Henrique Duczmal [3] Denise Bulgarelli Duczmal [4]

ABSTRACT

A unified framework for sequential parameter learning in state space models is proposed. This framework is capable of accommodating several other algorithms found in the literature as special cases, and this generality is achieved mainly by providing an alternative formalism to the role of regularization in this setting. In order to illustrate its flexibility, three algorithms are developed within this framework, including an improved and fully-adapted version of the celebrated Liu and West filter. These regularization techniques are associated with efficient resampling schemes, and their use is illustrated in challenging nonlinear settings with both synthetic and real-world data.

Keywords: *Bayesian inference; Sequential Monte Carlo methods; State space models*

1 Introduction

In this paper we deal with Bayesian inference for a discrete-time *State-Space* Model (SSM) defined by a Markovian transition

$$X_t|(X_{0:t-1} = x_{0:t-1}, \theta) =^d X_t|(X_{t-1} = x_{t-1}, \theta) \sim f(x_t|x_{t-1}, \theta) \tag{1}$$

with initial distribution $X_0 \sim \nu(x_0|\theta)$ and a conditional independence property

$$Y_t|(X_{0:t} = x_{0:t}, Y_{1:t-1} = y_{1:t-1}, \theta) =^d Y_t|(X_t = x_t, \theta) \sim g(y_t|x_t, \theta) \tag{2}$$

holding for all $t = 1, 2, \ldots$. Here $z_{1:j}$ denotes the sequence $(z_1, \ldots, z_j)$ for positive integer j, "$=^d$" denotes equality in distribution and $\theta \in \Theta \subseteq \mathbb{R}^{d_\theta}$ is a vector of parameters indexing the model. For each t, $(X_t, Y_t) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ and $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$, with $d_\theta := \dim(\theta)$, $d_x := \dim(X_t)$ and $d_y := \dim(Y_t)$. We refer to the components of the Markov chain $(X_t)_{t\geq 0}$ as the *states* of the model and to the conditionally independent (given each X_t) sequence $(Y_t)_{t\geq 1}$ as the *observations*.

Since typically in state space models only a sequence $Y_{1:t}$ of observations is available for inference, the states are usually thought of as being "hidden", leading to SSMs sometimes being called *Hidden Markov Models*. The states are frequently conceptualized as *dynamic parameters* in the model, and θ as the vector of *static parameters*, since θ is assumed to be fixed (albeit usually unknown). Classic references on the general theory of discrete-time state space models include West and Harrison (1997), Cappé et al. (2005) and Durbin and Koopman (2012).

In this paper, we focus on the problem of performing inference for static parameters in an *online* setting, i.e. in which estimates are required to be updated sequentially as new observations are made. This is usually referred to as the *sequential parameter learning* problem in the literature. Within the Bayesian inferential paradigm, this can essentially be reduced to computing the marginal posterior distribution

[1](Corresponding author). Department of Statistics, Universidade Federal de Minas Gerais. Av. Pres. Antônio Carlos, 6627, Belo Horizonte MG, 31270-901, Brazil. Email: urielmoreirasilva@gmail.com.
[2]Department of Statistics, Universidade Federal de Minas Gerais. Av. Pres. Antônio Carlos, 6627, Belo Horizonte MG, 31270-901, Brazil. Email: felipe.silva@bancointer.com.br.
[3]Department of Statistics, Universidade Federal de Minas Gerais. Av. Pres. Antônio Carlos, 6627, Belo Horizonte MG, 31270-901, Brazil. Email: duczmal@est.ufmg.br.
[4]Department of Mathematics, Universidade Federal de Minas Gerais. Av. Pres. Antônio Carlos, 6627, Belo Horizonte MG, 31270-901, Brazil. Email: bulgarelli@ufmg.br.

$p(\theta|y_{1:t})$ for each t conditional on data $Y_{1:t} = y_{1:t}$ and prior information $\theta \sim p(\theta)$. For simplicity, here we will adopt the terms "distribution" and "density" interchangeably when confusion is not possible, and implicitly assume that suitable sigma-finite dominating measures exist for all densities associated with $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 1}$ to be well-defined.

Sequential parameter learning techniques usually involve at some stage the computation of the posterior distribution of the states $X_{0:t}$ given the observations $Y_{1:t} = y_{1:t}$ and a (fixed) value of $\theta = \theta^*$, i.e. $p(x_{0:t}|y_{1:t}, \theta^*)$. This subproblem is often solved with the use of *Sequential Monte Carlo* (SMC) techniques, also known as *Particle Filters* (see Doucet and Johansen, 2009, for a full survey). Particle filters work by sequentially propagating samples (known as *particles*, hence the name) of the states over time and assigning probabilities to each of these samples, yielding an empirical distribution approximating the desired posterior.

Given the success and popularity of SMC methods, there is a vast body of literature on sequential parameter learning techniques that build on these methods. Amongst these we highlight Kitagawa (1998), Andrieu et al. (1999), BøLviken et al. (2001), Liu and West (2001), Gilks and Berzuini (2001), Chopin (2002), Fearnhead (2002), Storvik (2002), Vercauteren et al. (2005), Polson et al. (2008), Flury and Shephard (2009), Carvalho et al. (2010), Chopin et al. (2013) and Fulop and Li (2013). Collectively, the variety of fields in which SMC-based techniques are used to deal with sequential parameter learning problems arising in empirical settings also illustrate the effectiveness of this approach. Examples range from tracking (Wang et al., 2009; Ghaeminia et al., 2010; Liang and Piché, 2010; Nemeth et al., 2013) to epidemiology (Rodeiro and Lawson, 2006; Dukic et al., 2012; Lin and Ludkovski, 2014; Liu et al., 2015), ecology (Peters et al., 2010), econometrics (Golightly and Wilkinson, 2006; Carvalho and Lopes, 2007; Fulop and Li, 2013), finance (Yümlü et al., 2015; Jacquier et al., 2016; Warty et al., 2018; Virbickaitė et al., 2019) and even psychometrics (Reichenberg, 2018).

Ubiquitous as they might be, however, sequential Monte Carlo methods inherently suffer from the unavoidable drawback of *weight degeneracy* (Kong et al., 1994). In essence, degeneracy leads to an ever-decreasing efficiency of the method, as over time the probabilities (weights) of newly-generated samples become increasingly concentrated on a fewer set of distinct values. In order to mitigate degeneracy, SMC methods typically incorporate a *resampling* strategy, which replicates sampled points according to their weights and thus prevents the filter's support from collapsing to a single point.

Although in pratice resampling is very effective at mitigating weight degeneracy, it gives rise to another problem: *sample impoverishment*. As sampled points get successively resampled, their corresponding paths coalesce, resulting into very poor approximations of $p(x_{l:t}|y_{1:t}, \theta)$ for $l \ll t$. In the SMC context, sample impoverishment is also known as *path degeneracy* (Andrieu et al., 2005). It is especially damaging in sequential parameter learning, where we usually rely on approximations of functionals of the entire state path $X_{0:t}$.

In this paper we introduce a unified framework that includes most of the methods mentioned above for sequential parameter learning as special cases. The level of generality achieved by this framework is made possible only due to the introduction of an alternative formalism for performing *regularization* (Musso et al., 2001) within the sequential parameter learning setting (Liu and West, 2001). As a part of this unified framework, we also propose three algorithms, which by combining regularization, *full adaptation* (Petetin and Desbouvries, 2013) and minimal entropy resampling methods (Crisan and Lyons, 2002), actively attempt to minimize path degeneracy and consequently its detrimental effects on parameter estimation.

The paper is organized as follows: Section 2 briefly reviews particle filters, Section 3 introduces the unified framework for sequential parameter learning, Section 4 provides different simulation and real-data experiments to assess the performance of the discussed techniques and Section 5 contains some concluding remarks. The paper also has a Supplement showing how the unified framework proposed here can accomodate many of the already existing methods for sequential parameter learning in the literature as special cases.

2 Particle Filters

At a first stage, assume that we have complete knowledge about the static parameter θ. Upon observing $Y_{1:t} = y_{1:t}$, we thus only need to concern ourselves with the estimation of the states $X_{0:t}$. Usually, since only an estimate of the *filtering distribution* $p(x_t|y_{1:t}, \theta)$ is required for each t, this is commonly referred to as the *pure filtering* problem. However, given that this density is just a marginal of the joint distribution $p(x_{0:t}|y_{1:t}, \theta)$, here we will actually focus on the latter, since most of our motivation

and results are based on the entire sequence $X_{0:t}|(Y_{1:t}, \theta)$. Throughout this entire section, dependence on θ is suppressed to simplify notation.

The pure filtering problem can be fundamentally described by the recursion

$$\begin{aligned} p(x_{0:t}|y_{1:t}) &= \frac{p(y_t|x_{0:t}, y_{1:t-1})p(x_t|x_{0:t-1}, y_{0:t-1})p(x_{0:t-1}|y_{1:t-1})p(y_{1:t-1})}{p(y_t|y_{1:t-1})p(y_{1:t-1})} \\ &= p(x_{0:t-1}|y_{1:t-1})\frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|y_{1:t-1})} \\ &\propto p(x_{0:t-1}|y_{1:t-1})f(x_t|x_{t-1})g(y_t|x_t), \end{aligned} \tag{3}$$

where $p(x_t|x_{0:t-1}, y_{1:t-1}) = f(x_t|x_{t-1})$ follows from the Markov property (1) and $p(y_t|x_{0:t}, y_{1:t-1}) = g(y_t|x_t)$ follows from the conditional independence property (2).

Although apparently simple, the density function (3) cannot be evaluated analytically in general scenarios i.e. most non-linear and non-Gaussian models, and usually has to be approximated. A popular and flexible approach to accomplish this is to rely on *Particle Filters*, also known as *Sequential Monte Carlo* (SMC) methods. Particle filters rely on a Monte Carlo (MC) approximation to the joint distribution $p(x_{0:t}|y_{1:t})$ of the form

$$\hat{p}(dx_{0:t}|y_{1:t}) := \sum_{i=1}^{N} w_t^i \delta_{x_{0:t}^i}(dx_{0:t}), \tag{4}$$

where each $x_{0:t}^i$ (usually called a *particle*, *trajectory* or *path*) is drawn from a distribution q approximating p, $(w_t^i)_{i=1}^N$ satisfy $w_t^i \geq 0$ for each i with $\sum_{i=1}^N w_t^i = 1$, and $\delta_a(dx) := d\delta_a(x)/dx$, with δ_a denoting Dirac measure (point mass) at the point a.

The estimator (4) falls into the class of *Importance Sampling* (IS) methods, and the w_t^i's are called *importance weights*. In essence, SMC methods are importance samplers that exploit the sequential nature of state space models in order to efficiently sample particles $(x_{0:t}^i)_{i=1}^N$ and compute their corresponding importance weights $(w_t^i)_{i=1}^N$. The basic assumption in SMC is that the *proposal distribution* q from which we draw samples $(x_{0:t}^i)_{i=1}^N$ from satisfies

$$q(x_{0:t}|y_{1:t}) = q(x_{0:t-1}|y_{1:t-1})q(x_t|x_{0:t-1}, y_{1:t}). \tag{5}$$

A direct implication of (5) is that each x_t^i can be drawn marginally from $q(x_t|x_{0:t-1}^i, y_{1:t})$, thus yielding an $\mathcal{O}(N)$ complexity procedure[1] by conditioning on $x_{0:t-1}^i$ instead of the $\mathcal{O}(tN)$ required for jointly sampling x_t^i and $x_{0:t-1}^i$. The particle produced by SMC is then given by $x_{0:t}^i = (x_{0:t-1}^i, x_t^i)$, and its corresponding importance weight $w_t \equiv w(x_{0:t}, y_{1:t}) := p(x_{0:t}|y_{1:t})/q(x_{0:t}|y_{1:t})$ can be recursively evaluated by

$$w_t = \frac{p(x_{0:t}|y_{1:t})}{q(x_{0:t}|y_{1:t})} \propto \frac{p(x_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1}|y_{1:t-1})}\frac{f(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{0:t-1}, y_{1:t})} = w_{t-1}\frac{f(x_t|x_{t-1})g(y_t|x_t)}{q(x_t|x_{0:t-1}, y_{1:t})}. \tag{6}$$

To ensure that the weights all sum to 1 and to avoid dealing with the proportionality constant (given here by $p(y_t|y_{1:t-1})$, i.e. the *predictive distribution* of Y_t given $Y_{1:t-1} = y_{1:t-1}$ and which is also typically not available in closed form), we evaluate the weights with (6) for each i and then divide each of them by the overall sum of weights (i.e. across all indices $i = 1, \ldots, N$). In an obvious abuse of notation, we accomplish this by letting $w_t^i = w_t^i / \sum_{j=1}^N w_t^j$, where $w_t^i \equiv w(x_{0:t}^i, y_{1:t})$ is the importance weight associated with the particle $x_{0:t}^i$. The set $(x_{0:t}^i, w_t^i)_{i=1}^N$ produced by this procedure defines a discrete distribution that approximates the true law of $X_{0:t}|Y_{1:t}$ arbitrarily well as $N \to +\infty$; see e.g. Crisan and Doucet (2002).

An essential feature of SMC methods is that a *resampling* step is often required to mitigate the so-called *weight degeneracy* (Kong et al., 1994) phenomenon inherent in sequential importance sampling. As time progresses, weight degeneracy implies (in probability) that the weights of the most important particles will be increasingly larger, until eventually only one particle remains in the system. Resampling essentially mitigates this effect by replicating particles according to their importance and therefore increasing (in probability) diversity in the long run. Note however that degeneracy is ultimately unavoidable: with the exception of trivial settings, the particle system $(x_{0:t}^i, w_t^i)_{i=1}^N$ eventually degenerates

[1] Technically, for the procedure of drawing from $q(x_t|x_{0:t-1}^i, y_{1:t})$ to be exactly $\mathcal{O}(N)$, a fixed-dimensional set of sufficient statistics depending on $(x_{0:t-1}^i, y_{1:t})$ and which can be updated recursively must also exist. Otherwise, the complexity of the procedure will also increase over time – not from sampling each particle x_t^i itself, but rather from computing the conditional distribution of $X_t|(X_{0:t-1}^i, Y_{1:t})$ under q prior to sampling.

to a single point for large enough t (see e.g. Chopin et al., 2004). Nevertheless, in practice resampling is usually effective enough to ensure that SMC methods perform satisfactorily well in most scenarios.

A convenient, popular and quite general theoretical framework for explicitly including resampling into SMC algorithms is the *Auxiliary Particle Filter* (APF) of Pitt and Shephard (1999). The authors introduced into their filter additional auxiliary variables $(A_t)_{t\geq 0}$ (hence the name) taking values in $\{1,\dots,N\}$, and formulate resampling as drawing a set $(a_t^i)_{i=1}^N$ and then setting the a_t^i's as the indices of the corresponding sampled particle (here, each a_t^i is a realized value of the random variable A_t). More specifically, for each t we sample a_{t-1}^i with replacement from $\{1,\dots,N\}$ with probability λ_t^i (more on λ_t^i below), set $x_{0:t-1}^i \leftarrow x_{0:t-1}^{a_{t-1}^i}$ and then sample x_t^i from q conditional on both $x_{0:t-1}^{a_{t-1}^i}$ and $y_{1:t}$ for each $i=1,\dots,N$. The ith generated particle in the APF is therefore set to $x_t^i := (x_{0:t-1}^{a_{t-1}^i}, x_t^i)$.

It is important to point out that in APF resampling is done *prior* to sampling a new particle, and that is why here we denote the set of auxiliary variables $(a_{t-1}^i)_{i=1}^N$ with a time index of $t-1$, even though each of them are sampled at time t. Some authors refer to this class of method as a *resample-propagate* filter, and within the APF framework this is essentially done to ensure that the procedure is general (see below) and *adapted*, meaning that we can include information from the most recent observation Y_t not only when sampling the states X_t but also when resampling past trajectories $X_{0:t-1}$. In the terminology and notation of Andrieu et al. (2010), a_{t-1}^i is called the "ancestor" or "parent" index at time $t-1$ of particle $x_{0:t}^i$, since when resampling we effectively replace $x_{0:t-1}^i$ with $x_{0:t-1}^{a_{t-1}^i}$.

Whenever confusion is not possible, to alleviate the notation here we will denote the sample values of the ancestor indices A_t's without explicit reference to their particle indices, i.e. we denote a_t^i generically by a_t. The same is done for functions of these indices and all other random variables in the particle filter, i.e. we write $z_t^{a_{t-1}}$ instead of $z_t^{a_{t-1}^i}$ (here the z_t's are realizations of Z_t). We also write sequences of resampled values generically as either $Z_{0:t-1}^{A_{t-1}}$ or as $z_{0:t-1}^{a_{t-1}}$ for the corresponding realizations.

Formally, resampling in the APF framework can be seen as an additional importance sampling step performed when sampling the states at time t. More specifically, since here we sample $x_{0:t-1}^{a_{t-1}^i}$ from the set $(x_{0:t-1}^i)_{i=1}^N$ with probabilities λ_t^i and, since each $x_{0:t-1}^i$ is weighted by w_{t-1}^i in the approximation $\hat{p}(dx_{0:t-1}|y_{1:t})$ given in (4), the new importance weight assigned to each $x_{0:t-1}^{a_{t-1}^i}$ in the resampling step is then $w_{t-1}^{a_{t-1}^i}/\lambda_{t-1}^{a_{t-1}^i}$. This is equivalent to assuming that the marginal proposal for drawing a_{t-1}^i conditional on $(x_{0:t-1}^i, y_{1:t})$ and then x_t^i conditional on $(x_{0:t-1}^{a_{t-1}^i}, y_{1:t})$ is proportional to $\lambda_{t-1}^{a_{t-1}^i} q(x_t|x_{0:t-1}^{a_{t-1}}, y_{1:t})$. If we omit the particle indices for simplicity, the weight recursion in the APF is therefore given by

$$w_t \propto \frac{w_{t-1}^{a_{t-1}}}{\lambda_t^{a_{t-1}}} \frac{f(x_t|x_{t-1}^{a_{t-1}})g(y_t|x_t)}{q(x_t|x_{0:t-1}^{a_{t-1}}, y_{1:t})}. \tag{7}$$

Starting with $x_0^i \sim \nu(x_0)$ and $w_0^i \propto 1$ for all $i=1,\dots,N$, an APF step from time $t-1$ to t is summarized in Algorithm 1.

Algorithm 1: Auxiliary Particle Filter

for *i = 1* **to** *N* **do**

sample a_{t-1}^i from $\{1,\dots,N\}$ with probability λ_t^i

draw $x_t^i \sim q(x_t|x_{0:t-1}^{a_{t-1}^i}, y_{1:t})$

compute and normalize $w_t^i \propto \frac{w_{t-1}^{a_{t-1}^i}}{\lambda_t^{a_{t-1}^i}} \frac{f(x_t^i|x_{t-1}^{a_{t-1}^i})g(y_t|x_t^i)}{q(x_t^i|x_{0:t-1}^{a_{t-1}^i}, y_{1:t})}$

end

It is important to point out that another equivalent way to define the APF is as a filter that targets a different sequence of distributions, i.e. that does not target $\{p(x_{0:t}|y_{1:t})\}_{t\geq 1}$ directly (Johansen and Doucet, 2008; Doucet and Johansen, 2009). If at time $t-1$ we have a weighted sample $(x_{0:t-1}^i, w_{t-1}^i)_{i=1}^N$, the APF targets an *intermediate* distribution proportional to $\lambda_{t-1}^i q(x_t|x_{0:t-1}^i, y_{1:t})$. Since $p(x_t, x_{0:t-1}^i|y_{1:t}) \propto w_{t-1}^i f(x_t|x_{t-1}^i)g(y_t|x_t)$, the resulting *incremental weights* are given by the ratio between $w_{t-1}^i f(x_t|x_{t-1}^i)g(y_t|x_t)$ and $\lambda_{t-1}^i q(x_t|x_{0:t-1}^i, y_{1:t})$, yielding the same weight recursion as in (7).

Now, there are two basic design choices one must make within the APF framework: the choice of *intermediate weights* λ_t and of the state proposal density $q(x_t|x_{0:t-1}, y_{1:t})$. Usually, these choices are

made in order to keep the importance weights w_t as constant as possible, since then the variance of w_t (conditional on both $X_{0:t-1}$ and $Y_{1:t}$) is minimal (Doucet et al., 2000). The optimal choice in this sense is to let $\lambda_t \propto w_{t-1} p(y_t|x_{t-1})$ and $q(x_t|x_{0:t-1}, y_{1:t}) = p(x_t|x_{t-1}, y_t)$, since in this case the weight recursion (7) becomes

$$w_t \propto \frac{w_{t-1}^{a_{t-1}}}{w_{t-1}^{a_{t-1}} p(y_t|x_{t-1}^{a_{t-1}})} \frac{f(x_t|x_{t-1}^{a_{t-1}}) g(y_t|x_t)}{p(x_t|x_{t-1}^{a_{t-1}}, y_t)} = \frac{1}{p(y_t|x_{t-1}^{a_{t-1}})} \frac{f(x_t|x_{t-1}^{a_{t-1}}) g(y_t|x_t)}{\frac{f(x_t|x_{t-1}^{a_{t-1}}) g(y_t|x_t)}{p(y_t|x_{t-1}^{a_{t-1}})}} = 1. \tag{8}$$

In the terminology of Pitt and Shephard (1999), a filter such that (8) holds is said to be *fully-adapted* (FA) (the converse is a *blind* procedure, i.e. one that does not incorporate any information about the most recent observation y_t). Note that in the above derivation we have used that $p(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1}) g(y_t|x_t)/p(y_t|x_{t-1})$, directly deduced from (1-2).

The main problem associated with full adaptation is that it requires simulating from $p(x_t|x_{t-1}, y_t)$ and being able to evaluate $p(y_t|x_{t-1})$ pointwise, both of which might be unfeasible in practice. In this case, Pitt and Shephard (1999) proposed approximating these densities by taking $\lambda_t \propto w_{t-1} g(y_t|\mu_t)$ and $q(x_t|x_{0:t-1}, y_{1:t}) = f(x_t|x_{t-1})$, where $\mu_t := \mu(X_{0:t-1})$ is any prediction of $X_{0:t-1}$, such as the one-step-ahead conditional expectation, median or mode of $X_t|X_{0:t-1}$. This so-called "lookahead" strategy is in principle readily appliable to any SSM, and if μ_t is close to X_{t-1} the resulting intermediate weights will be close to the optimal ones.

The weight recursion (7) for the lookahead strategy becomes

$$w_t \propto \frac{w_{t-1}^{a_{t-1}}}{w_{t-1}^{a_{t-1}} g(y_t|\mu_t^{a_{t-1}})} \frac{f(x_t|x_{t-1}^{a_{t-1}}) g(y_t|x_t)}{f(x_t|x_{t-1}^{a_{t-1}})} = \frac{g(y_t|x_t)}{g(y_t|\mu_t^{a_{t-1}})}. \tag{9}$$

From (9), we can see that the closer $g(y_t|\mu_t)$ is to $g(y_t|x_t)$, the closer w_t is to being constant, which means that the lookahead strategy is most successful whenever the observations are very informative. Note here that although μ_t is denoted at time t, it is actually a function of $x_{0:t-1}$, and therefore when we resample we also assign to μ_t the ancestor index a_{t-1}.

As mentioned before, the APF framework is quite general, and it includes most particle filters in the literature as special cases. Classical examples are the *Bootstrap Filter* of Gordon et al. (1993), which can be obtained by taking $q(x_t|x_{0:t-1}, y_{1:t}) = f(x_t|x_{t-1})$ and $\lambda_t = w_{t-1}$, and the more general *Sampling Importance Resampling* (SIR) algorithm (Doucet et al., 2000), obtained by taking $\lambda_t = w_{t-1}$ and choosing $q(x_t|x_{0:t-1}, y_{1:t})$ freely.

Finally, although resampling complicates the dynamics of the particle system considerably, estimators $\hat{p}(f)$ of the form (4) can still be proven to converge to $p(f)$ almost surely for a large class of test functions f under suitable regularity conditions (Del Moral, 2004).

3 Sequential Parameter Learning

We now turn to the general situation in which θ is unknown and has to be inferred from the data. As stated previously, our problem is to learn about $\theta \in \Theta$ sequentially, i.e. to compute $p(\theta|y_{1:t})$ for all t. In Sections 3.1 and 3.2 we introduce a unified class of algorithms for dealing with this problem, and in Section 3.3 we illustrate the flexibility allowed by this unified framework by proposing three algorithms for sequential parameter learning.

In the Supplement we also show how the unified framework proposed in this paper accomodates many of the commonly found methods for sequential parameter learning in the literature as special cases.

3.1 A Unified Framework

Let θ_t denote the inference for θ at time t, i.e. the parameter associated with the posterior $p(\theta|y_{1:t})$. Although θ is still inherently static, keeping track of the inference for it across time allows us to implicitly define a sequence $(\theta_t)_{t \geq 0}$ with initial distribution given by the prior $\theta_0 \sim p(\theta)$. The joint posterior for $(X_{0:t}, \theta_{0:t})$ given $Y_{1:t} = y_{1:t}$ admits the recursion

$$\begin{aligned} p(x_{0:t}, \theta_{0:t}|y_{1:t}) = p(\theta_t|x_{0:t}, \theta_{0:t-1}, y_{1:t}) p(y_t|x_{0:t}, \theta_{0:t-1}, y_{1:t-1}) \cdot \\ p(x_t|x_{0:t-1}, \theta_{0:t-1}, y_{1:t-1}) \frac{p(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1}) p(y_{1:t-1})}{p(y_t|y_{1:t-1}) p(y_{1:t-1})} \end{aligned}$$

$$= p(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t}) g(y_t|x_t, \theta_{t-1}) f(x_t|x_{t-1}, \theta_{t-1}) \frac{p(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})}$$
$$\propto p(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1}) f(x_t|x_{t-1}, \theta_{t-1}) g(y_t|x_t, \theta_{t-1}) p(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t}). \tag{10}$$

In the above derivation we implicitly assume that all marginal distributions of $(X_t, Y_t)_{t\geq 0}$ depend only on the most recent value of θ. Analogous to the APF terminology, this property is sometimes referred to as a *perfect adaptation* of the law of $(X_t, Y_t)_{t\geq 0}$ to the most recent value of the sequence $(\theta_t)_{t\geq 0}$, as in e.g. Andrieu et al. (2010).

Without resampling, if we assume that the proposal for drawing $(X_{0:t}, \theta_{0:t})$ satifies

$$q(x_{0:t}, \theta_{0:t}|y_{1:t}) = q(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1}) \cdot q(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t}) q(x_t|x_{0:t-1}, \theta_{0:t-1}, y_{1:t}), \tag{11}$$

we then have the weight recursion

$$\begin{aligned} w_t &:= \frac{p(x_{0:t}, \theta_{0:t}|y_{1:t})}{q(x_{0:t}, \theta_{0:t}|y_{1:t})} \\ &\propto \frac{p(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1})}{q(x_{0:t-1}, \theta_{0:t-1}|y_{1:t-1})} \frac{f(x_t|x_{t-1}, \theta_{t-1}) g(y_t|x_t, \theta_{t-1}) p(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t})}{q(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t}) q(x_t|x_{0:t-1}, \theta_{0:t-1}, y_{1:t})} \\ &= w_{t-1} \frac{f(x_t|x_{t-1}, \theta_{t-1}) g(y_t|x_t, \theta_{t-1})}{q(x_t|x_{0:t-1}, \theta_{0:t-1}, y_{1:t})} \frac{p(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t})}{q(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t})}. \end{aligned} \tag{12}$$

With resampling, under the APF framework, we assign ancestor indices a_{t-1} to all quantities from 0 to $t-1$ (i.e. the sequences $x_{0:t-1}$ and $\theta_{0:t-1}$, the importance weights w_{t-1} and the intermediate weights λ_t). Taking into account the resampling weights λ_t, the weight recursion (12) becomes

$$w_t \propto \frac{w_{t-1}^{a_{t-1}}}{\lambda_t^{a_{t-1}}} \frac{f(x_t|x_{t-1}^{a_{t-1}}, \theta_{t-1}^{a_{t-1}}) g(y_t|x_t, \theta_{t-1}^{a_{t-1}})}{q(x_t|x_{0:t-1}^{a_{t-1}}, \theta_{0:t-1}^{a_{t-1}}, y_{1:t})} \frac{p(\theta_t|x_t, x_{0:t-1}^{a_{t-1}}, \theta_{0:t-1}^{a_{t-1}}, y_{1:t})}{q(\theta_t|x_t, x_{0:t-1}^{a_{t-1}}, \theta_{0:t-1}^{a_{t-1}}, y_{1:t})}. \tag{13}$$

Similar to the APF, the sequential parameter learning framework proposed above can also be interpreted as a procedure that targets an intermediate distribution proportional to $\lambda_t^i \, q(x_t|x_{0:t-1}^i, \theta_{0:t-1}^i, y_{1:t}) \, q(\theta_t|x_t, x_{0:t-1}^i, \theta_{0:t-1}^i, y_{1:t})$ (note that the filter is now a function of both x_t and θ_t). Given that we can decompose $p(x_t, x_{0:t-1}^i, \theta_t, \theta_{0:t-1}^i|y_{1:t}) \propto w_{t-1}^i f(x_t|x_{t-1}^i, \theta_{t-1}^i) g(y_t|x_t, \theta_{t-1}^i) p(\theta_t|x_t, x_{0:t-1}^i, \theta_{0:t-1}^i, y_{1:t})$, the resulting incremental weights satisfy the same recursion (13).

Now, unlike in the APF, pointwise evaluation of the weight recursion (13) requires the ability of not only evaluating the usual ratios w_{t-1}^i/λ_t^i and $f(x_t^i|x_{t-1}^i, \theta_{t-1}^i) g(y_t|x_t^i, \theta_{t-1}^i)/p(x_t^i|x_{0:t-1}^i, \theta_{0:t-1}^i, y_{1:t})$ for each i, but also the ratio $p(\theta_t^i|x_t^i, x_{0:t-1}^i, \theta_{0:t-1}^i, y_{1:t})/q(\theta_t|x_t, x_{0:t-1}^i, \theta_{0:t-1}^i, y_{1:t})$, at least up to a proportionality constant. This usually requires making additional assumptions about the specific (or approximate) form of $p(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t})$ and, as illustrated below in Section 3.3, this is essentially what differentiates one sequential parameter learning algorithm from the other.

The fundamental design choices in the framework proposed here are the intermediate weights λ_t and the state and static parameter proposals $q(x_t|x_{0:t-1}, \theta_{0:t-1}, y_{1:t})$ and $q(\theta_t|x_t, x_{0:t-1}, \theta_{0:t-1}, y_{1:t})$. Starting with $\theta_0^i \sim p(\theta)$, $x_0^i \sim \nu(x_0|\theta_0^i)$ and $w_0^i \propto 1$ for $i = 1, \ldots, N$, a sequential parameter learning step from time $t-1$ to t is summarized in Algorithm 2. It should be clear that the class proposed here also contains the APF described in Algorithm 1 (i.e. without any parameter learning) by simply taking θ_t to be a fixed quantity θ^* for all t, or equivalently by assuming that $p(\theta) = \delta_{\theta^*}(d\theta)$ and $p(d\theta_t|x_t, x_{0:t-1}, y_{1:t}) = \delta_{\theta^*}(d\theta_t)$ for all t.

The main output from Algorithm 2 is the approximation

$$\hat{p}(dx_{0:t}, d\theta_{0:t}|y_{1:t}) := \sum_{i=1}^{N} w_t^i \delta_{(x_{0:t}^i, \theta_{0:t}^i)}(dx_{0:t} d\theta_{0:t}), \tag{14}$$

which is typically referred to as the *histogram-based* estimator of the joint posterior distribution of $(X_{0:t}, \theta_{0:t})$ given $Y_{1:t} = y_{1:t}$. To obtain an approximation to the target $p(\theta|y_{1:t})$, we can simply integrate (14) over the support of $(X_{0:t}, \theta_{0:t-1})$, yielding

$$\hat{p}(d\theta_t|y_{1:t}) := \int_{\mathcal{X}^{t+1}\times\Theta^t} \hat{p}(dx_{0:t}, d\theta_{0:t}|y_{1:t}) dx_{0:t} d\theta_{0:t-1} = \sum_{i=1}^{N} w_t^i \delta_{\theta_t^i}(d\theta_t). \tag{15}$$

Algorithm 2: Sequential Parameter Learning

for $i = 1$ **to** N **do**

sample a_{t-1}^i from $\{1, \ldots, N\}$ with probability λ_t^i

draw $x_t^i \sim q(x_t | x_{0:t-1}^{a_{t-1}^i}, \theta_{0:t-1}^{a_{t-1}^i}, y_{1:t})$

draw $\theta_t^i \sim q(\theta_t | x_t, x_{0:t-1}^{a_{t-1}^i}, \theta_{0:t-1}^{a_{t-1}^i}, y_{1:t})$

compute and normalize $w_t^i \propto \frac{w_{t-1}^{a_{t-1}^i}}{\lambda_t^{a_{t-1}^i}} \frac{f(x_t^i | x_{t-1}^{a_{t-1}^i}, \theta_{t-1}^{a_{t-1}^i}) g(y_t | x_t^i, \theta_{t-1}^{a_{t-1}^i})}{q(x_t^i | x_{0:t-1}^{a_{t-1}^i}, \theta_{0:t-1}^{a_{t-1}^i}, y_{1:t})} \frac{p(\theta_t^i | x_t^i, x_{0:t-1}^{a_{t-1}^i}, \theta_{0:t-1}^{a_{t-1}^i}, y_{1:t})}{q(\theta_t^i | x_t^i, x_{0:t-1}^{a_{t-1}^i}, \theta_{0:t-1}^{a_{t-1}^i}, y_{1:t})}$

end

Proceeding analogously, estimators of any marginal of $p(x_{0:t}, \theta_{0:t} | y_{1:t})$ can be obtained by integrating (14) accordingly. In particular, integrating over the entire path of the static parameters $\theta_{0:t}$ results in the state posterior (4) obtained in the pure filtering context.

Besides the usual histogram-based estimator defined in (15), an alternative estimator of $p(d\theta_t | y_{1:t})$ can be obtained via *Rao-Blackwellization* (Liu and Chen, 1998; Doucet et al., 2000). First, note that we can rewrite the target distribution as

$$\begin{aligned} p(\theta_t | y_{1:t}) &= \int_{\mathcal{X}^{t+1} \times \Theta^t} p(\theta_t, x_{0:t}, \theta_{0:t-1} | y_{1:t}) dx_{0:t} d\theta_{0:t-1} \\ &= \int_{\mathcal{X}^{t+1} \times \Theta^t} p(\theta_t | x_{0:t}, \theta_{0:t-1}, y_{1:t}) p(x_{0:t}, \theta_{0:t-1} | y_{1:t}) dx_{0:t} d\theta_{0:t-1} \\ &= \mathbb{E}_{p(X_{0:t}, \theta_{0:t-1} | Y_{1:t})}[p(\theta_t | X_{0:t}, \theta_{0:t-1}, Y_{1:t})], \end{aligned} \tag{16}$$

i.e. as the expectation of $p(\theta_t | x_{0:t}, \theta_{0:t-1}, y_{1:t})$ taken with respect to $p(x_{0:t}, \theta_{0:t-1} | y_{1:t})$. Now, we can obtain a direct Monte Carlo approximation to (16) by simply replacing $p(x_{0:t}, \theta_{0:t-1} | y_{1:t})$ with $\hat{p}(dx_{0:t}, d\theta_{0:t-1} | y_{1:t})$ in the corresponding integral. This gives

$$\begin{aligned} \bar{p}(d\theta_t | y_{1:t}) &:= \int_{\mathcal{X}^{t+1} \times \Theta^t} p(d\theta_t | x_{0:t}, \theta_{0:t-1}, y_{1:t}) \hat{p}(dx_{0:t}, d\theta_{0:t-1} | y_{1:t}) dx_{0:t} d\theta_{0:t-1} \\ &= \int_{\mathcal{X}^{t+1} \times \Theta^t} p(d\theta_t | x_{0:t}, \theta_{0:t-1}, y_{1:t}) \sum_{i=1}^N w_t^i \delta_{(x_{0:t}^i, \theta_{0:t-1}^i)}(dx_{0:t} d\theta_{0:t-1}) dx_{0:t} d\theta_{0:t-1} \\ &= \sum_{i=1}^N w_t^i p(d\theta_t | x_{0:t}^i, \theta_{0:t-1}^i, y_{1:t}). \end{aligned} \tag{17}$$

The resulting expression for $\bar{p}(d\theta_t | y_{1:t})$ given in (17) is then known as the *Rao-Blackwellized* estimator of the posterior $p(\theta_t | y_{1:t})$.

The Rao-Blackwellized estimator (17) is typically (Liu and Chen, 1998) more efficient than the histogram-based estimator (15) whenever interest lies in approximating only the posterior $p(\theta_t | y_{1:t})$, i.e. the typical setting for sequential parameter learning. However, this comes at the cost of having to evaluate $p(\theta_t | x_{0:t}, \theta_{0:t-1}, y_{1:t})$ pointwise. Moreover, if interest lies in the moments and/or general functionals of $\theta_t | Y_{1:t}$, analytical solutions to the required integrals might be much more involved and sometimes unnatainable when compared to the simpler histogram-based estimator.

Finally, note that we could have also defined our framework to be a *propagate-resample* one, i.e. in which we first sample the states/parameters and then perform the resampling step. However, this might be undesirable since then the resampled sequences $X_{0:t-1}^{A_{t-1}^i}$ and $\theta_{0:t-1}^{A_{t-1}^i}$ will not benefit from current information on Y_t, i.e. the procedure will be *blind* in APF terminology. Works comparing propagate-resample and resample-propagate frameworks from a theoretical standpoint include e.g. Petetin and Desbouvries (2013), and from an empirical standpoint include e.g. Lopes and Tsay (2011). Also, although in Algorithm 2 we first sample the states X_t and only then the parameters θ_t, the procedure is general enough to accomodate methods that have the reverse sampling order, such as Liu and West (2001)'s and Storvik (2002)'s filters; see the Supplement for more details.

3.2 Regularization

Due to the unavoidable degeneracy inherent in sequential importance sampling methods, the resampling step is an integral part of SMC. Despite its benefits, however, resampling has an important

drawback: *sample impoverishment*, also known as *path degeneracy* (Andrieu et al., 2005).

Path degeneracy manifests itself as the coalescence of particles' paths ocurring from successive resampling steps. As an example, consider a functional $Z_{l:t-1}$ defined for integer $0 \leq l \leq t-1$ of $(X_{l:t-1}, \theta_{l:t-1})$ computed recursively along the filter's trajectory. At time t, $(z_{l:t-1}^{a_{t-1}^i})_{i=1}^N$ is the set resampled from the realizations $(z_{l:t-1}^i)_{i=1}^N$ and, due to some $z_{l:t-1}^i$'s naturally having lower weights than others, the resampled set $(z_{l:t-1}^{a_{t-1}^i})_{i=1}^N$ will have fewer distinct values than $(z_{l:t-1}^i)_{i=1}^N$. At time $t+1$, we resample $(z_{l:t}^{a_t^i})_{i=1}^N$ from $(z_{l:t}^i)_{i=1}^N$ and, since each $z_{l:t}^i = (z_{l:t-1}^{a_{t-1}^i}, z_t^i)$, the $z_{l:t-1}^{a_{t-1}^i}$'s are going to be resampled again, given that $(z_{l:t}^{a_t^i})_{i=1}^N = (z_{l:t-1}^{a_{t-1}^{a_t^i}}, z_t^{a_t^i})_{i=1}^N$, and take even fewer distinct values than before. Over time, this is compounded and the paths $z_{l:t}^i$ degenerate (hence the name) to a single point.

By the same argument above, path degeneracy is also progressively worse as l is closer to 0. Whenever $l = t-1$, however, (i.e. when only Z_{t-1} is of interest), sample impoverishment is minimal since the transition from Z_{t-1} to Z_t essentially "replenishes" the number of distinct values the functional Z_{t-1} can take from one step to another. This is why path degeneracy can be mostly ignored whenever interest lies only in state filtering, since the state transition from X_{t-1} to X_t will usually allow for a proper exploration of the state space even when the number of distinct values X_{t-1} is small.

Whenever the state transition does not allow for a proper exploration of the state space, however, path degeneracy can become problematic even if $l = t-1$. This is especially true for sequential parameter learning, since the static parameters for which we are trying to perform inference for usually have no "natural" dynamic. Here, even if we are only interested in the most recent value θ_t, if the parameters are static we implicitly have $\theta_t = \theta_{t-1}$ for all t and eventually $\theta_t = \theta_0$, meaning that at each time t we only resample from an ever-decreasing set of distinct values drawn from the prior $p(\theta)$.

In an attempt to mitigate path degeneracy, several authors have proposed variants of a technique that can be generally defined as *regularization* (Musso et al., 2001). In essence, regularization is a modification of the resampling step to allow for resampled particles to assume values other than the ones specified by the set of current particles. In the above example, this means that with regularization the set of unique values in $(z_{l:t-1}^{a_{t-1}^i})_{i=1}^N$ is different of (and usually contains) the set of unique values in $(z_{l:t-1}^i)_{i=1}^N$, increasing diversity. This also allows for additional exploration of the state space, which is especially important for static parameters, since their support will no longer be limited to a subset of values initially drawn from the prior.

As mentioned before, under the auxiliary variable interpretation presented so far in Section 2, resampling in an APF framework can be understood as sampling ancestor indices $(a_{t-1}^i)_{i=1}^N$ from $\{1, \ldots, N\}$ with probabilities $(\lambda_t^i)_{i=1}^N$ and then setting $z_{l:t-1}^i \leftarrow z_{l:t-1}^{a_{t-1}^i}$. This is equivalent to drawing a set $(\check{z}_{l:t-1})_{i=1}^N$ with replacement from the empirical distribution

$$\check{p}(dz_{l:t-1}|y_{1:t}) := \sum_{i=1}^N \lambda_t^i \delta_{z_{l:t-1}^i}(dz_{l:t-1}), \tag{18}$$

where each $\check{z}_{l:t-1}^i := z_{l:t-1}^{a_{t-1}^i}$ (recall that the probability of each a_{t-1}^i is proportional to λ_t^i and, although each $z_{l:t-1}^i$ is weighted by w_{t-1}^i – implying that the importance weights associated with resampling are going to be proportional to w_{t-1}^i/λ_t^i for each i – the resampling weights themselves are given by λ_t^i).

A regularized version of the empirical resampling distribution (18) is defined as the convolution of $\check{p}(dz_{l:t-1}|y_{1:t})$ with a *regularization kernel* (Silverman, 1986) $K(\cdot)$, i.e.

$$\begin{aligned}
\tilde{p}(dz_{l:t-1}|y_{1:t}) &:= \int K(dz_{l:t-1} - dz_{l:t-1}^*)\check{p}(dz_{l:t-1}^*|y_{1:t})dz_{l:t-1}^* \\
&= \int K(dz_{l:t-1} - dz_{l:t-1}^*) \sum_{i=1}^N \lambda_t^i \delta_{z_{l:t-1}^i}(dz_{l:t-1}^*)dz_{l:t-1}^* \\
&= \sum_{i=1}^N \lambda_t^i K(dz_{l:t-1} - z_{l:t-1}^i).
\end{aligned} \tag{19}$$

Traditionally, $K(\cdot)$ is assumed to be the probability density of a continuous random variable with zero mean and finite second moment taking values in $\mathbb{R}^{(t-1-l)\times d_z}$, where $d_z := \dim(Z_t)$. If we denote the ith draw of $\tilde{p}(dz_{l:t-1}|y_{1:t})$ by $\tilde{z}_{l:t-1}^i$, it is then clear that the set of possible values assumed by each $\tilde{z}_{l:t-1}^i$ is no longer limited to the finite set $(z_{l:t-1}^i)_{i=1}^N$, but rather the uncountable image set of $K(\cdot)$. Therefore, by

regularizing resampled draws, we end up with a set of values $(\tilde{z}^i_{l:t-1})_{i=1}^N$ which is more diverse than the original set of resampled values $(z^{a^i_{t-1}}_{l:t-1})_{i=1}^N$, alleviating the effects of path degeneracy. Note that $dz^*_{l:t-1}$ in the above derivation is only an integration variable; the density $\check{p}$ in the first and second integral is still (18).

As mentioned before, regularization is especially effective in sequential parameter learning due to the fact that it allows for exploration of the parameter space by otherwise static parameters. The first widely succesful example of this is the method proposed by Liu and West (2001), which relies on a Gaussian kernel with location and scale determined by past parameter values and an additional user-defined scale specified via discount factors (see also the Supplement and Section 3.3.1). In order to obtain a more general framework, however, here we will assume that $K(\cdot)$ is any probability distribution density.

We incorporate regularization within the framework we developed in Section 3.1 as an extra importance sampling step performed after resampling and prior to sampling states and parameters. More specifically, after resampling $(x^i_{0:t-1}, \theta^i_{0:t-1})_{i=1}^N$ to $(x^{a^i_{t-1}}_{0:t-1}, \theta^{a^i_{t-1}}_{0:t-1})_{i=1}^N$, for each $i = 1, \ldots, N$ we draw the regularized particles $(\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})$ from $K\big((dx_{0:t-1}, d\theta_{0:t-1}) - (x^{a^i_{t-1}}_{0:t-1}, \theta^{a^i_{t-1}}_{0:t-1})\big)$ and, conditional on $(\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})$, draw the current states x^i_t from $q(x_t|\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$ and parameters θ^i_t from $q(\theta_t|x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})$.

Now, assuming that we can draw exactly from $K(\cdot)$, the importance weights from the regularization procedure are then proportional to the ratio between $K\big((dx_{0:t-1}, d\theta_{0:t-1}) - (\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})\big)$ and $K\big((dx_{0:t-1}, d\theta_{0:t-1}) - (\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})\big)$, i.e. proportional to 1. This is a case of *perfect sampling*, and aside from a specific application shown in the Supplement, in practice this assumption usually holds (although we can generalize the procedure slightly by assuming that the "true" or target regularization kernel is given by $K(\cdot)$ and we can only sample from $K_q(\cdot)$, resulting in importance weights that must then be multiplied by the ratio $K(\cdot)/K_q(\cdot)$).

Finally, the weight recursion of our sequential parameter learning filter with regularization is

$$\begin{aligned} w_t &\propto \frac{w^{a^i_{t-1}}_{t-1}}{\lambda^{a^i_{t-1}}_t} \frac{K\big((dx_{0:t-1}, d\theta_{0:t-1}) - (\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})\big)}{K\big((dx_{0:t-1}, d\theta_{0:t-1}) - (\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})\big)} \cdot \\ &\qquad \frac{f(x_t|\tilde{x}_{t-1}, \tilde{\theta}_{t-1}) g(y_t|x_t, \tilde{\theta}_{t-1})}{q(x_t|\tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})} \frac{p(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})}{q(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})} \\ &= \frac{w^{a^i_{t-1}}_{t-1}}{\lambda^{a^i_{t-1}}_t} \frac{f(x_t|\tilde{x}_{t-1}, \tilde{\theta}_{t-1}) g(y_t|x_t, \tilde{\theta}_{t-1})}{q(x_t|\tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})} \frac{p(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})}{q(\theta_t|x_t, \tilde{x}_{0:t-1}, \tilde{\theta}_{0:t-1}, y_{1:t})}. \end{aligned} \tag{20}$$

Algorithm 3 summarizes a step from $t-1$ to t for the procedure proposed here. This procedure effectively generalizes the sequential parameter learning filter without regularization described in Section 3.1 by taking $K\big((dx_{0:t-1}, d\theta_{0:t-1}) - (x^{a^i_{t-1}}_{0:t-1}, \theta^{a^i_{t-1}}_{0:t-1})\big) = \delta_{(x^{a^i_{t-1}}_{0:t-1}, \theta^{a^i_{t-1}}_{0:t-1})}(dx_{0:t-1} d\theta_{0:t-1})$, since then Algorithm 3 is equivalent to Algorithm 2 with $\tilde{x}^i_{0:t-1}$ replaced by $x^{a^i_{t-1}}_{0:t-1}$ and $\tilde{\theta}^i_{0:t-1}$ replaced by $\theta^{a^i_{t-1}}_{0:t-1}$.

Algorithm 3: Sequential Parameter Learning with Regularization

for $i = 1$ **to** N **do**
- sample a^i_{t-1} from $\{1, \ldots, N\}$ with probability λ^i_t
- draw $(\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}) \sim K\big((x_{0:t-1}, \theta_{0:t-1}) - (x^{a^i_{t-1}}_{0:t-1}, \theta^{a^i_{t-1}}_{0:t-1})\big)$
- draw $x^i_t \sim q(x_t|\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$
- draw $\theta^i_t \sim q(\theta_t|x_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$
- compute and normalize $w^i_t \propto \frac{w^{a^i_{t-1}}_{t-1}}{\lambda^{a^i_{t-1}}_t} \frac{f(x^i_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}) g(y_t|x^i_t, \tilde{\theta}^i_{t-1})}{q(x^i_t|\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})} \frac{p(\theta^i_t|x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})}{q(\theta^i_t|x^i_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})}$

end

Note that we can once again interpret the sequential parameter learning filter with regularization as a filter targeting, at time t, an intermediate distribution proportional to $\lambda^i_t K\big((dx_{0:t-1}, d\theta_{0:t-1}) - (\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})\big) q(x_t|\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t}) q(\theta_t|x_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$. Since $p(x_t, \tilde{x}^i_{0:t-1}, \theta_t, \tilde{\theta}^i_{0:t-1}|y_{1:t}) \propto w^i_{t-1}$ $K\big((dx_{0:t-1}, d\theta_{0:t-1}) - (\tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1})\big) f(x_t|\tilde{x}^i_{t-1}, \tilde{\theta}^i_{t-1}) g(y_t|x_t, \tilde{\theta}^i_{t-1}) p(\theta_t|x_t, \tilde{x}^i_{0:t-1}, \tilde{\theta}^i_{0:t-1}, y_{1:t})$, the resulting incremental weights yield the same weight recursion in (20).

Now, it is important to point out that the ideas of introducing artificial dynamics for parameters, including θ within the state component or even performing Gibbs sampling conditional on sufficient statistics along the filter's trajectory are not novel. The same can be said for employing regularization in order to mitigate path degeneracy; the linkage between these ideas were established as early as Liu and West (2001). However, to our knowledge no effort so far has been made use both these ideas simultaneously, making for 2 distinct sources of dynamics for the parameters θ, and as a result allowing for greater generality and also for the combination of advantages from different algorithms.

As an example, take the filters of Liu and West (2001) (LW filter) and Carvalho et al. (2010) (called Particle Learning, or PL). Both methods are set within an APF framework, but in the former we first sample θ_t and then X_t, and in the latter we first sample X_t and then θ_t. Further, the LW filter relies on regularization-type dynamics for θ_t and on a general lookahead strategy, and even in cases in which we know $p(y_t|x_{t-1}, \theta_{t-1})$ analytically, full adaptation is not possible due to the sampling order adopted. On the other hand, PL always assumes a fully adapted APF framework, and relies on Gibbs-type moves conditional on sufficient statistics for θ_t.

Although the difference between both the LW filter and PL might seem irreconcilable, both of them can be seen as special cases of the sequential parameter learning filter with regularization developed here (see the Supplement for details). Furthermore, within our framework we can also develop a fully adapted version of the LW filter (see Section 3.3.1) and a regularized version of PL (see Section 3.3.2), allowing for both methods to take advantage of the strongest points of the other.

Finally, an important aspect of mitigating path degeneracy is executing the resampling step as efficiently as possible. Specifically, here we advocate the use of the tree-based branching algorithm of Crisan and Lyons (2002), proven to have minimal variance amongst all *unbiased* (i.e. such that the expected number of offspring ξ_t^i of particle $z_{l:t-1}^i$ equals $N \cdot \lambda_t^i$) resampling methods. We also advocate the use of fully-adapted procedures whenever possible, since in most cases full adaptation can drastically reduce the variance of the resampling weights.

3.3 Three particular cases

We now illustrate the flexibility allowed by the unified framework proposed here by introducing three algorithms for sequential parameter learning.

3.3.1 Fully-adapted Liu and West's Filter

The first method proposed will be hereafter referred to as *Fully-adapted Liu and West's* (FALW) *filter*. As its name implies, this method consists of choosing a similar regularization kernel to that of the Liu and West (2001) (LW) filter (see also the Supplement), but adopts a fully-adapted framework instead of the original lookahead APF one. By relying on optimality results from kernel density estimation theory (Silverman, 1986; see also Musso et al., 2001 for a specific application of these results in the context of particle filters), we also select kernel bandwidths automatically and independently of the adoption of discount factors as in the original method.

More specifically, let

$$K\big((dx_{0:t-1},\, d\theta_{0:t-1}) - (x_{0:t-1}^{a_{t-1}^i}, \theta_{0:t-1}^{a_{t-1}^i})\big) = \\ = d\mathcal{N}\big((dx_{t-1}, d\theta_{t-1})|m_{t-1}^{a_{t-1}^i}, h^2 V_{t-1}\big) \cdot \delta_{(x_{0:t-2}^{a_{t-1}^i}, \theta_{0:t-2}^{a_{t-1}^i})}(dx_{0:t-2} d\theta_{0:t-2}), \tag{21}$$

where $d\mathcal{N}(dx|\mu, \Sigma)$ denotes the probability density at dx of a Gaussian random variable with mean μ and variance Σ, and

$$\begin{aligned} m_{t-1}^i &:= a z_{t-1}^i + (1-a)\bar{z}_{t-1}, \\ \bar{z}_{t-1} &:= \sum_{i=1}^N w_{t-1}^i z_{t-1}^i, \\ V_{t-1} &:= \sum_{i=1}^N w_{t-1}^i [z_{t-1}^i - \bar{z}_{t-1}][z_{t-1}^i - \bar{z}_{t-1}]^T, \end{aligned} \tag{22}$$

with $z_{t-1} := (x_{t-1}, \theta_{t-1})$, $h := \{4/[N \cdot (2 + d_z)]\}^{1/(d_z+4)}$, $a := \sqrt{1-h^2}$, $d_z := \dim(z_{t-1})$ and A^T denoting the transpose of matrix A. Note that in this method a is always well defined for $N \geq 2$, since

then $0 \leq h \leq 1$. The rule for defininining the bandwidth h is usually referred to as Silverman (1986)'s "rule-of-thumb".

In order to achieve full adaptation, in FALW we choose the optimal intermediate weights $\lambda_t^i \propto w_{t-1}^i p(y_t|x_{t-1}^i, \theta_{t-1}^i)$ and optimal state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i, y_t)$. By also assuming that

$$p(d\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = \delta_{\tilde{\theta}_{t-1}^i}(d\theta_t), \tag{23}$$

and that $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, the corresponding importance weights (20) are then

$$w_t^i \propto \frac{w_{t-1}^{a_{t-1}^i}}{w_{t-1}^{a_{t-1}^i} p(y_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{p(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i, y_t)} \frac{\delta_{\tilde{\theta}_{t-1}^i}(d\theta_t)}{\delta_{\tilde{\theta}_{t-1}^i}(d\theta_t)} = 1. \tag{24}$$

Note that unlike in the LW filter, in FALW we also regularize past states x_{t-1}^i along with past parameters θ_{t-1}^i, striving for reducing path degeneracy in state and parameter trajectories $(x_{0:t-1}^i, \theta_{t-1}^i)_{i=1}^N$ even further.

Although at first sight the modifications to the original method by Liu and West (2001) that define FALW here might not warrant the definition of an entire new algorithm, we highlight from the discussion at the end of Section 3.1 that full adaptation in this case is only possible due to the formalization of LW moves as draws from a regularization kernel, which essentially allows us to reverse the original sampling order from θ_t first and then x_t to x_t first and then θ_t. This is a situation which we have not encountered outside of our work.

3.3.2 Regularized Particle Learning

The second method introduced in this paper is a regularized version of the Particle Learning (PL) algorithm of Carvalho et al. (2010; see also the Supplement), hereafter referred to as *Regularized Particle Learning* (RPL). The theoretical reasoning for RPL is that, in addition to sampling current parameters conditional on sufficient statistics, regularizing past states, past parameters and even past sufficient statistics would mitigate path degeneracy in their past trajectories even further (see e.g. Chopin et al., 2010, for a specific discussion on path degeneracy in sufficient statistics and its effect on inference over time).

The regularization kernel adopted in RPL is analogous to that of the FALW method, but here we also explicitly include past trajectories of sufficient statistics $S_{0:t-1}$ for resampling and regularization, i.e.

$$\begin{aligned} &K\big((dx_{0:t-1}, d\mathcal{S}_{0:t-1}, d\theta_{0:t-1}) - (x_{0:t-1}^{a_{t-1}^i}, \mathcal{S}_{0:t-1}^{a_{t-1}^i}, \theta_{0:t-1}^{a_{t-1}^i})\big) = \\ &\quad = d\mathcal{N}\big((dx_{t-1}, d\mathcal{S}_{t-1}, d\theta_{t-1})|m_{t-1}^{a_{t-1}^i}, h^2 V_{t-1}\big) \cdot \delta_{(x_{0:t-2}^{a_{t-1}^i}, \mathcal{S}_{0:t-2}^{a_{t-1}^i}, \theta_{0:t-2}^{a_{t-1}^i})}(dx_{0:t-2} d\mathcal{S}_{0:t-2} d\theta_{0:t-2}), \end{aligned} \tag{25}$$

where m_{t-1}^i, a, h and V_{t-1} are defined as in (22) but for $z_{t-1} := (x_{t-1}, \mathcal{S}_{t-1}, \theta_{t-1})$.

In RPL we adopt a general APF framework, by choosing intermediate weights λ_t^i and state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$ to be defined by the user. Since this method is based on Particle Learning, the target parameter distribution is assumed to satisfy

$$p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|\mathcal{S}_t^i), \tag{26}$$

where the sufficient statistics are propagated according to $\mathcal{S}_t^i = \mathcal{S}(\mathcal{S}_{t-1}^{a_{t-1}^i}, x_t^i, y_t)$, as usual (note that it is also possible to include the update of $\mathcal{S}_t^i$ explicitly within the filter's recursions, as done in e.g. Carvalho et al., 2010). Finally, by additionally assuming that the parameter proposal distribution satisfies $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, the weight recursion (20) for RPL is

$$w_t^i \propto \frac{w_{t-1}^{a_{t-1}^i}}{\lambda_t^{a_{t-1}^i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})} \frac{p(\theta_t^i|\mathcal{S}_t^i)}{p(\theta_t^i|\mathcal{S}_t^i)} = \frac{w_{t-1}^{a_{t-1}^i}}{\lambda_t^{a_{t-1}^i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i) g(y_t|x_t^i, \tilde{\theta}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})}. \tag{27}$$

Note that the regularization of past static parameters θ_{t-1}^i in RPL only affects the importance weights w_t^i and current sampled states x_t^i, since the current parameters θ_t^i are sampled independently from $p(\theta|\mathcal{S}_t^i)$. This method also allows for full adaptation by choosing optimal importance weights $\lambda_t^i \propto w_{t-1}^i p(y_t|x_{t-1}^i, \theta_{t-1}^i)$ and optimal state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i, y_t)$.

3.3.3 Hybrid FALW-RPL Filter

The last method introduced here is a hybrid between the FALW and RPL algorithms of the two previous sections, and that we thus name as *Hybrid FALW-RPL* algorithm. Inspired by the Hybrid LW-PL algorithm of Chen et al. (2010; see also the Supplement), which is a hybrid between the original Liu and West and Particle Learning methods, this technique has the benefit of allowing for Gibbs updates whenever sufficient statistics are available for a subset φ_t^i of the static parameter vector θ_t^i, while also allowing regularization-based inference for the rest of the parameters. Similar to RPL, past states, parameters and even sufficient statistics are regularized in this method, and, similar to FALW, a fully-adapted APF framework is adopted. All of these choices contribute to an effort of mitigating path degeneracy as much as possible.

Let $\theta = (\phi, \varphi)$, where ϕ is the subset for which we perform LW-type regularization moves and φ is the subset for which $p(\varphi|\mathcal{S}_t)$ is available. The regularization kernel adopted here is the same as in RPL, i.e.

$$\begin{aligned} K\big((dx_{0:t-1}, d\mathcal{S}_{0:t-1},\, d\theta_{0:t-1}) - (x_{0:t-1}^{a_{t-1}^i}, \mathcal{S}_{0:t-1}^{a_{t-1}^i}, \theta_{0:t-1}^{a_{t-1}^i})\big) = \\ = d\mathcal{N}\big((dx_{t-1}, d\mathcal{S}_{t-1}, d\theta_{t-1})|m_{t-1}^{a_{t-1}^i}, h^2 V_{t-1}\big) \cdot \delta_{(x_{0:t-2}^{a_{t-1}^i}, \mathcal{S}_{0:t-2}^{a_{t-1}^i}, \theta_{0:t-2}^{a_{t-1}^i})}(dx_{0:t-2} d\mathcal{S}_{0:t-2} d\theta_{0:t-2}), \end{aligned} \tag{28}$$

since we regularize both ϕ_{t-1}^i and φ_{t-1}^i, i.e. the complete set θ_{t-1}^i. Similar to RPL, the definitions of m_{t-1}^i, a, h and V_{t-1} are analogous to those in (22) but for $z_{t-1} := (x_{t-1}, \mathcal{S}_{t-1}, \theta_{t-1})$, and sufficient statistics are propagated according to $\mathcal{S}_t^i = \mathcal{S}(\mathcal{S}_{t-1}^{a_{t-1}^i}, x_t^i, y_t)$,

Now, in order to obtain a fully-adapted procedure here we must choose the optimal intermediate weights $\lambda_t^i \propto w_{t-1}^i p(y_t|x_{t-1}^i, \theta_{t-1}^i)$ and optimal state proposal $q(x_t|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i, y_t)$. Since in this method we perform LW-type moves for ϕ and PL moves for φ, the target parameter distribution is the same as in Chen et al. (2010), i.e.

$$p(d\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = \delta_{\tilde{\phi}_{t-1}^i}(d\phi_t) p(d\varphi_t|\mathcal{S}_t^i). \tag{29}$$

Finally, by taking $q(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = p(\theta_t|x_t^i, \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t})$, we have the following importance weights (20):

$$\begin{aligned} w_t^i &\propto \frac{w_{t-1}^{a_{t-1}^i}}{\lambda_t^{a_{t-1}^i}} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i) g(y_t|x_t^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)}{q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\phi}_{0:t-1}^i, \tilde{\varphi}_{t-1}^i, y_{1:t})} \frac{\delta_{\tilde{\phi}_{t-1}^i}(d\phi_t) p(\varphi_t|\mathcal{S}_t^i)}{\delta_{\tilde{\phi}_{t-1}^i}(d\phi_t) p(\varphi_t|\mathcal{S}_t^i)} \\ &= \frac{w_{t-1}^{a_{t-1}^i}}{w_{t-1}^{a_{t-1}^i} p(y_t|\tilde{x}_{t-1}^{k_i}, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)} \frac{f(x_t^i|\tilde{x}_{t-1}^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i) g(y_t|x_t^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i)}{p(x_t^i|\tilde{x}_{t-1}^i, \tilde{\phi}_{t-1}^i, \tilde{\varphi}_{t-1}^i, y_t)} \\ &= 1. \end{aligned} \tag{30}$$

4 Experiments

In this section we illustrate the performance of the three methods proposed in this paper within the unified framework by means of numerical experiments with both simulation-based and real-world data.

Given that efficient resampling techniques often prove imperative to mitigate path degeneracy (see e.g. Andrieu et al., 1999, 2005, 2010; Chopin et al., 2010; Kantas et al., 2015), we adopted the minimal-entropy branching resampling scheme of Crisan and Lyons (2002) for all the experiments performed here.

4.1 Ecological Model

For the first experiment, we illustrate the difference in performance in sequential parameter estimation that might result from full adaptation. This is done by comparing the LW filter (Liu and West, 2001; see also the Supplement) with its fully-adapted counterpart introduced here in Section 3.3.1, the Fully Adapted Liu and West (FALW) filter. Our target for inference is the *θ-logistic* model (Peters et al., 2010), also known as the θ-Ricker model (Polansky et al., 2009). Here we adopt the parameterization of Polansky et al. (2009) in favor of the one in Peters et al. (2010) in order to ensure that the state process $(X_t)_{t\geq 0}$ has a stable equilibrium/stationary distribution.

Let

$$X_t = X_{t-1} + r\left\{1 - \left[\frac{\exp(X_{t-1})}{K}\right]^{\tau}\right\} + \sigma_U U_t, \qquad U_t \sim \mathcal{N}(0,1), \tag{31}$$

$$Y_t = X_t + \sigma_V V_t, \qquad V_t \sim \mathcal{N}(0,1), \tag{32}$$

with priors

$$X_0 \sim \mathcal{N}(0,4), \qquad r =^d \tau \sim G(2,10), \qquad K \sim G(1,0.1), \qquad \sigma_U^2 =^d \sigma_V^2 \sim IG(2,1),$$

where $\mathcal{N}(\mu,\sigma^2)$ denotes a Normal distribution with mean μ and variance σ^2, $G(a,b)$ denotes a Gamma distribution with shape a and rate b (i.e. with expectation a/b), and $IG(c,d)$ denotes an Inverse-Gamma distribution with shape c and scale d. Here we also assume that X_0, $(U_t)_{t\geq 1}$ and $(V_t)_{t\geq 1}$ are mutually and serially independent. The prior distributions chosen here for each component of θ are similar to those used by Chopin et al. (2013, supplement).

The state space for this model is $\mathcal{X} = \mathbb{R} := (-\infty, +\infty)$ and the static parameter is (note the inclusion of the initial state, X_0, as a fixed parameter) given by $\theta = (X_0, r, K, \tau, \sigma_U^2, \sigma_V^2)$, taking values in $\Theta = \mathbb{R} \times \mathbb{R}_+^5$, where $\mathbb{R}_+ := (0, +\infty)$. Here, if we let

$$F(x_{t-1}^i, \theta_{t-1}^i) := x_{t-1}^i + r_{t-1}^i\left\{1 - \left[\frac{\exp(x_{t-1}^i)}{K_{t-1}^i}\right]^{\tau_{t-1}^i}\right\},$$

we then have

$$f(x_t^i | x_{t-1}^i, \theta_{t-1}^i) = d\mathcal{N}\left(x_t^i \middle| F(x_{t-1}^i, \theta_{t-1}^i), (\sigma_U^2)_{t-1}^i\right), \qquad g(y_t | x_t^i, \theta_{t-1}^i) = d\mathcal{N}\left(y_t \middle| x_t^i, (\sigma_V^2)_{t-1}^i\right).$$

For this simulation study, we generated a series of $n = 1{,}000$ observations of the model given by (31-32) with $X_0 = \log(1.27)$, $r = 0.15$, $K = 6.2$, $\tau = 0.1$, $\sigma_U^2 = 0.47^2$ and $\sigma_V^2 = 0.39^2$ (these parameter values are the same ones used in an experiment performed by Peters et al., 2010). We then performed $M = 50$ independent runs of the LW and FALW filters with a support of $N = 50{,}000$ particles. We chose $\delta = 0.99$ for the LW filter, and adopted the rule-of-thumb bandwidth of Silverman (1986) with a Gaussian kernel for the FALW method. In this model, full adaptation is possible by choosing

$$\lambda_t^i \propto w_{t-1}^i p(y_t | x_{t-1}^i, \theta_{t-1}^i) = w_{t-1}^i d\mathcal{N}\left(y_t \middle| F(x_{t-1}^i, \theta_{t-1}^i), (\sigma_U^2)_{t-1}^i + (\sigma_V^2)_{t-1}^i\right)$$

and

$$p(x_t^i | x_{t-1}^i, \theta_{t-1}^i, y_t) = d\mathcal{N}\left(x_t^i \middle| \frac{(\sigma_U^2)_{t-1}^i \cdot y_t + (\sigma_V^2)_{t-1}^i \cdot F(x_{t-1}^i, \theta_{t-1}^i)}{(\sigma_U^2)_{t-1}^i + (\sigma_V^2)_{t-1}^i}, \frac{(\sigma_U^2)_{t-1}^i \cdot (\sigma_V^2)_{t-1}^i}{(\sigma_U^2)_{t-1}^i + (\sigma_V^2)_{t-1}^i}\right),$$

and these are the design choices for FALW. For the LW method, a lookahead APF strategy is adopted, with intermediate weights $\lambda_t^i \propto w_{t-1}^i g(y_t | \mu_{t-1}^i, m_{t-1}^i)$ and blind state proposal $q(x_t^i | \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = f(x_t | \tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)$, with $\mu_{t-1}^i = F(x_{t-1}^i, m_{t-1}^i)$ and m_{t-1}^i as defined in (22).

Note that since the parameter space is constrained here (i.e. Θ is a proper subset of $\mathbb{R}^{d_\theta}$) and the Gaussian kernels actually map from $\mathbb{R}^{d_\theta}$ to $\mathbb{R}^{d_\theta}$, we must perform regularization in a transformed scale, i.e. by working with $\check{r} := \log(r)$, $\check{K} := \log(K)$, $\check{\tau} := \log(\tau)$, $\check{\sigma}_U^2 := \log(\sigma_U^2)$ and $\check{\sigma}_V^2 := \log(\sigma_V^2)$ instead of r, K, τ, σ_U^2 and σ_V^2 directly.

Figure 1 contains the estimated marginal posteriors for each component of θ at $t = 1{,}000$ based on both LW and FALW methods, and Figure 2 contains the corresponding posterior traces at each time point. For both figures, we also implemented a particle Markov Chain Monte Carlo (pMCMC, Andrieu et al. 2010) for comparison (implementation details for the pMCMC algorithm can be found in the Supplement). Overall, we can see that for all of the parameters there is a high instability in the LW filter estimates (we obtained similar results by varying the discount factor up to $\delta = 0.5$), whereas for FALW the estimates are much more consistent across runs. FALW distributions are also more consistent with the pMCMC posteriors, and estimates for both LW and FALW seem to converge to the pMCMC posterior means for all of the parameters.

In order to add to the visual evidence contained in Figures 1 and 2, we also estimated the *Effective Sample Size* (ESS) metric of Carpenter et al. (1999) across all $M = 50$ runs for each parameter. The ESS provides us with a measure of the equivalent approximate number of i.i.d. variables supporting the

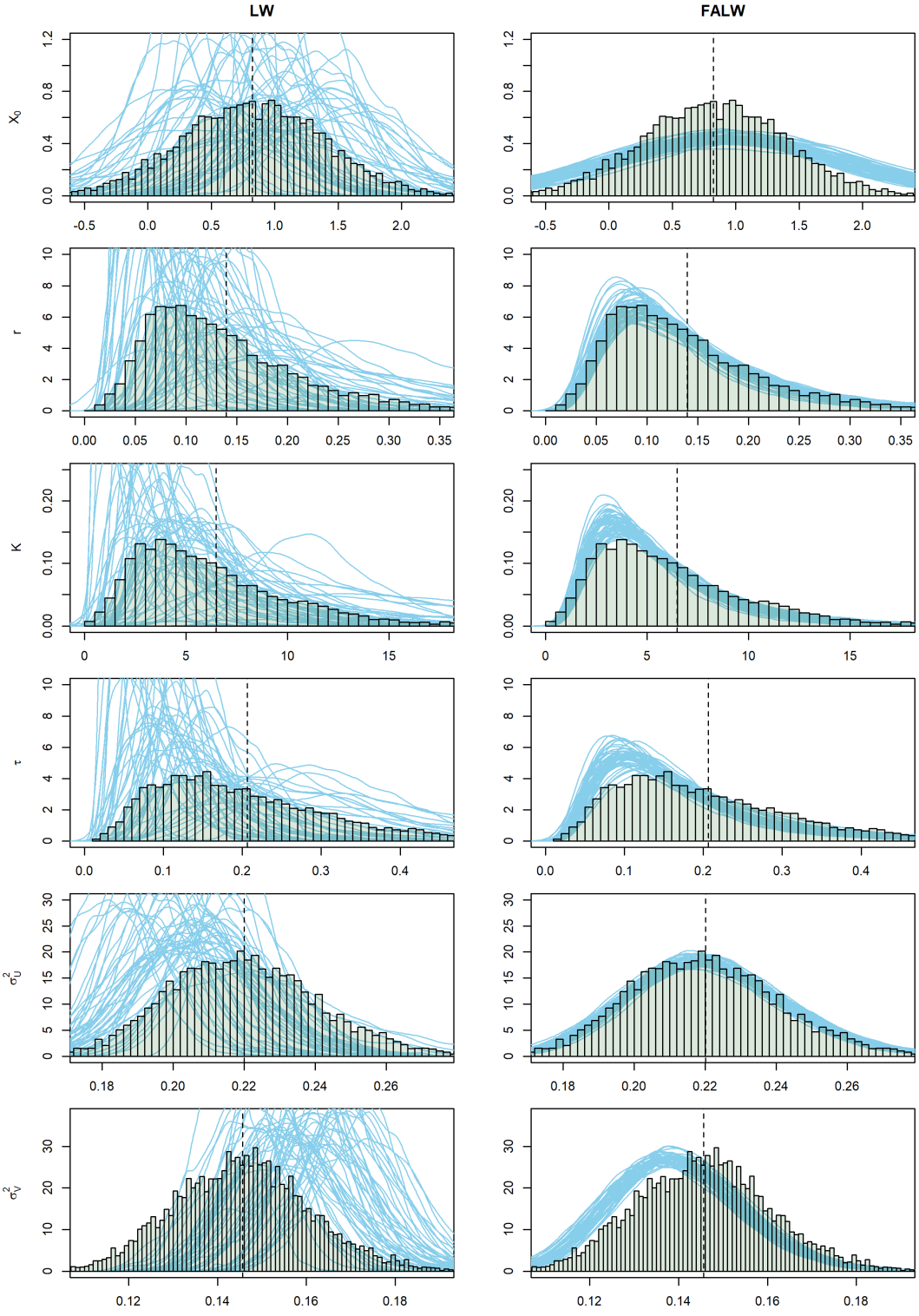

Figure 1: θ-logistic model (31-32): kernel density estimates (solid blue lines) of the posterior distributions at $t = 1{,}000$ for X_0, r, K, τ, σ_U^2 and σ_V^2 based on $M = 50$ independent runs of the Liu West filter (left column) and the Fully-Adapted Liu and West filter (right column) methods. The filters were run with $N = 50{,}000$ particles, and the true parameter values are $X_0 = \log(1.27) \simeq 0.2391$, $r = 0.15$, $K = 6.2$, $\tau = 0.1$, $\sigma_U^2 = 0.47^2 \simeq 0.2209$ and $\sigma_V^2 = 0.39^2 \simeq 0.1521$. Histogram bars are from the corresponding pMCMC run, and vertical dashed lines are the pMCMC posterior means.

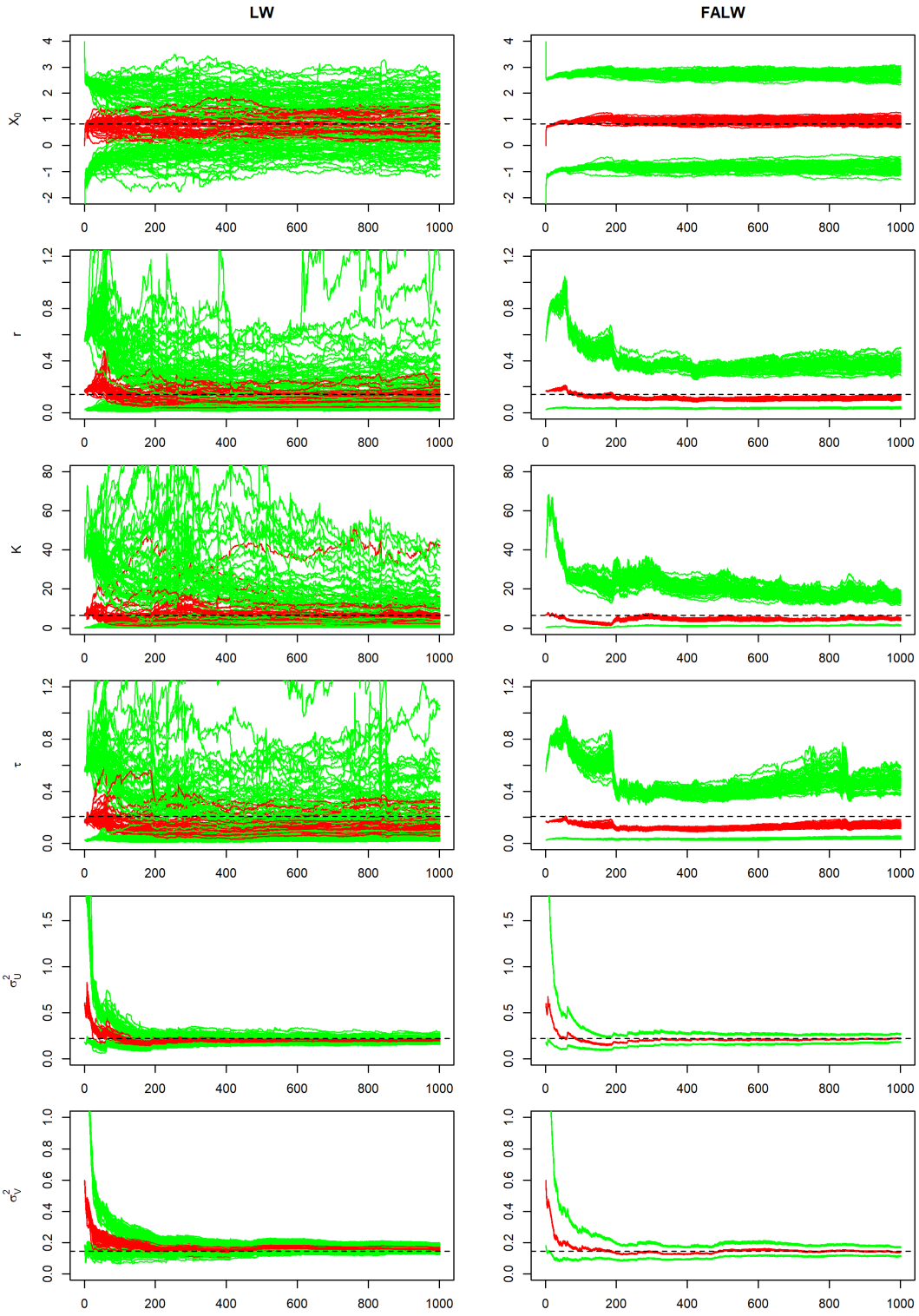

Figure 2: θ-logistic model (31-32): trace plots of the posterior distributions for X_0, r, K, τ, σ_U^2 and σ_V^2 based on $M = 50$ independent runs of the Liu West filter (left column) and the Fully-Adapted Liu and West filter (right column) methods. For each row, upper and lower solid green lines are the 2.5th and 97.5th posterior percentiles, and solid red lines are the posterior medians of LW or FALW at each time point. Black dashed lines are the pMCMC posterior means, and true parameter values are $X_0 = \log(1.27) \simeq 0.2391$, $r = 0.15$, $K = 6.2$, $\tau = 0.1$, $\sigma_U^2 = 0.47^2 \simeq 0.2209$ and $\sigma_V^2 = 0.39^2 \simeq 0.1521$. The filters were run with $N = 50{,}000$ particles.

target estimate (here the posterior mean of each parameter) for each method, so that the higher the ESS the more consistent across runs an estimate is (note that this is different from the more usual ESS metric of Kong et al., 1994, which is computed from the importance weights and instead measures the overall instability of the particle system). For LW, the ESS for each component of $\theta = (X_0, r, K, \tau, \sigma_U^2, \sigma_V^2)$ is, respectively, $(1, 2, 4, 1, 2, 2)$ and for FALW the corresponding ESSs are $(49, 53, 70, 57, 170, 110)$, corroborating the evidence that FALW-based estimates are much more consistent than those from the LW filter.

Regarding CPU time consumption, both methods perform equally well, with each run of FALW taking about a second longer in average. In a Core i7 CPU 860 running at 2.80 GHz, the mean CPU time of a LW run is 153.36 seconds and the mean CPU time of a FALW run is 181.14 seconds.

4.2 Nonlinear Seasonal Model

Next, we investigate the effect of regularization of past states, parameter values and sufficient statistics in parameter estimates over time. We do this by comparing the Particle Learning (PL) algorithm of Carvalho et al. (2010, see also the Supplement) with the Regularized Particle Learning (RPL) method introduced here in Section 3.3.2 in the *Nonlinear Seasonal Model* (NLSM), a model that was first proposed by Netto et al. (1978) and is widely popular as a toy example in the particle filter literature in general (see e.g. Gordon et al., 1993; Kitagawa, 1987; Cappé et al., 2005).

Let

$$X_t = \frac{X_{t-1}}{2} + 25\frac{X_{t-1}}{1+X_{t-1}^2} + 8\cos(1.2t) + \sigma_V V_t, \qquad V_t \sim \mathcal{N}(0,1), \tag{33}$$

$$Y_t = \frac{X_t^2}{20} + \sigma_W W_t, \qquad W_t \sim \mathcal{N}(0,1), \tag{34}$$

with priors

$$X_0 \sim \mathcal{N}(0,5), \qquad \sigma_V^2 \sim IG(1/2,1/2), \qquad \sigma^2 \sim IG(1/2,1/2),$$

where $(V_t)_{t\geq 0}$ and $(W_t)_{t\geq 0}$ are assumed to be mutually and serially independent, with $X_0 \perp\!\!\!\perp V_t$ and $X_0 \perp\!\!\!\perp W_t$ for all t.

The state space for the NLSM model is once again $\mathcal{X} = \mathbb{R}$, the static parameter vector is $\theta = (\sigma_V^2, \sigma_W^2)$ and the parameter space is $\Theta = \mathbb{R}_+^2$. Here,

$$f(x_t^i | x_{t-1}^i, \theta_{t-1}^i) = d\mathcal{N}\left(x_t^i \middle| \frac{x_{t-1}^i}{2} + 25\frac{x_{t-1}^i}{1+(x_{t-1}^i)^2} + 8\cos(1.2t), (\sigma_V^2)_{t-1}^i\right)$$

and

$$g(y_t | x_t^i, \theta_{t-1}^i) = d\mathcal{N}\big(y_t | x_t^i, (\sigma_W^2)_{t-1}^i\big).$$

For this experiment, we take $\sigma_V^2 = 10$ and $\sigma_W^2 = 1$ as the true parameter values, simulate a series of $n = 500$ observations and then perform $M = 50$ independent runs of the PL and RPL methods using $N = 50{,}000$ particles. Design choices for both the PL and RPL methods here are SIR intermediate weights $\lambda_t^i = w_{t-1}^i$ and blind state proposal $q(x_t^i | \tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = f(x_t | \tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)$.

The posterior distributions for σ_V^2 and σ_W^2 for the Gibbs sampling steps of both PL and RPL methods are available in closed form, and are given by

$$\sigma_V^2 | (X_{0:t}, Y_{1:t}) \sim IG(a_t/2, b_t/2), \qquad \sigma_W^2 | (X_{0:t}, Y_{1:t}) \sim IG(b_t/2, c_t/2),$$

where the sufficient statistics $\mathcal{S}_t = (a_t, b_t, c_t, d_t)$, satisfy

$$\begin{aligned}
a_t &:= a_0 + t = a_0 + (t-1) + 1 = a_{t-1} + 1, \\
b_t &:= b_0 + \sum_{k=1}^{t}(X_k - F_k)^2 = b_0 + \sum_{k=1}^{t-1}(Y_k - F_k)^2 + (Y_t - F_t)^2 = b_{t-1} + (Y_t - F_t)^2, \\
c_t &:= c_0 + t = c_0 + (t-1) + 1 = c_{t-1} + 1, \\
d_t &:= d_0 + \sum_{k=1}^{t}\left(Y_k - \frac{X_k^2}{20}\right)^2 = d_0 + \sum_{k=1}^{t-1}\left(Y_k - \frac{X_k^2}{20}\right)^2 + \left(Y_t - \frac{X_t^2}{20}\right)^2 = d_{t-1} + \left(Y_t - \frac{X_t^2}{20}\right)^2
\end{aligned}$$

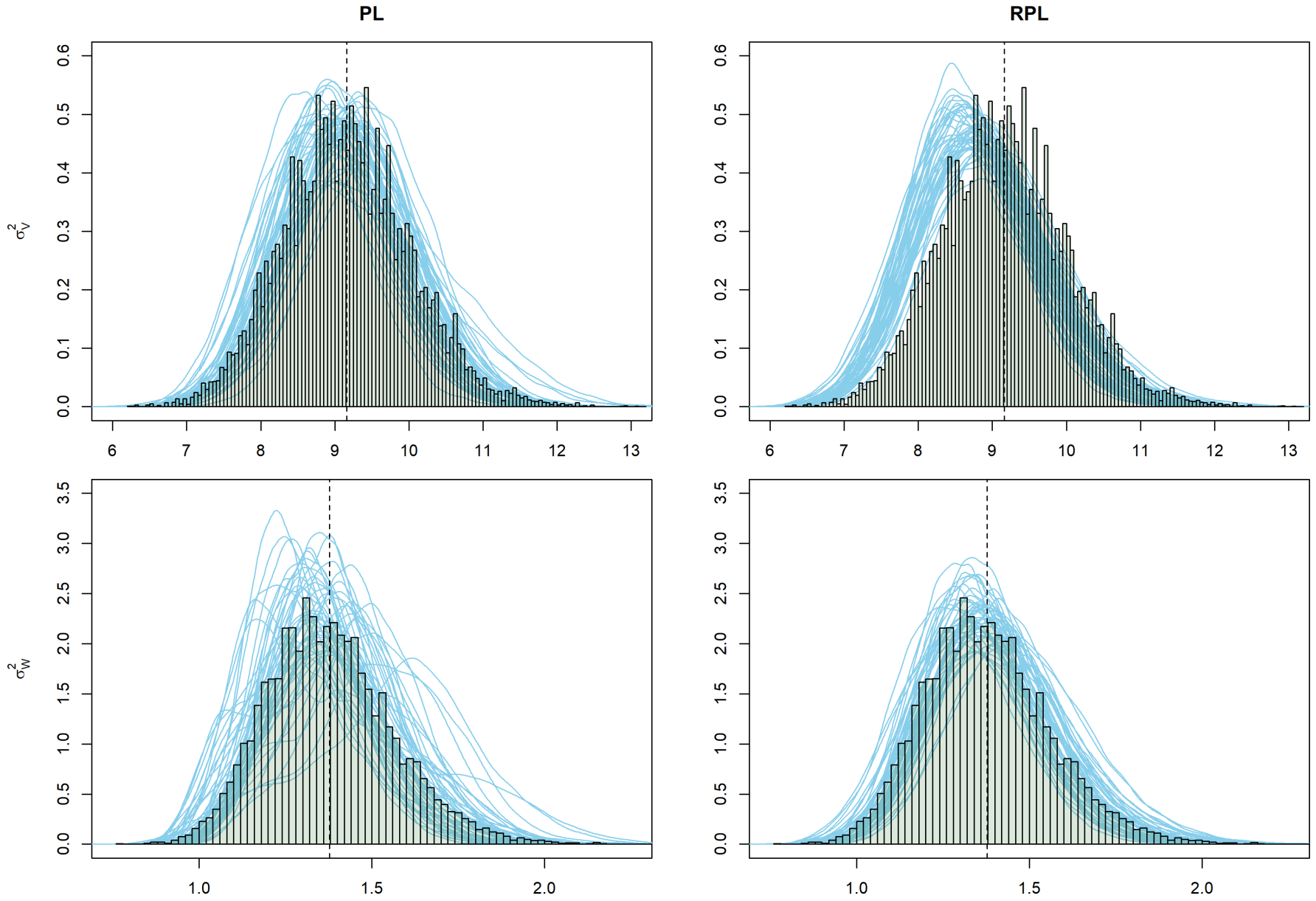

Figure 3: NLSM model (33-34): kernel density estimates (solid blue lines) of the posterior distributions at $t = 500$ for σ_V^2 and σ_W^2 based on $M = 50$ independent runs of the Particle Learning (left column) and the Regularized Particle Learning (right column) methods. The filters were run with $N = 50{,}000$ particles, and the true parameter values are $\sigma_V^2 = 10$ and $\sigma_W^2 = 1$. Histogram bars are from the corresponding pMCMC run, and vertical dashed lines are the pMCMC posterior means.

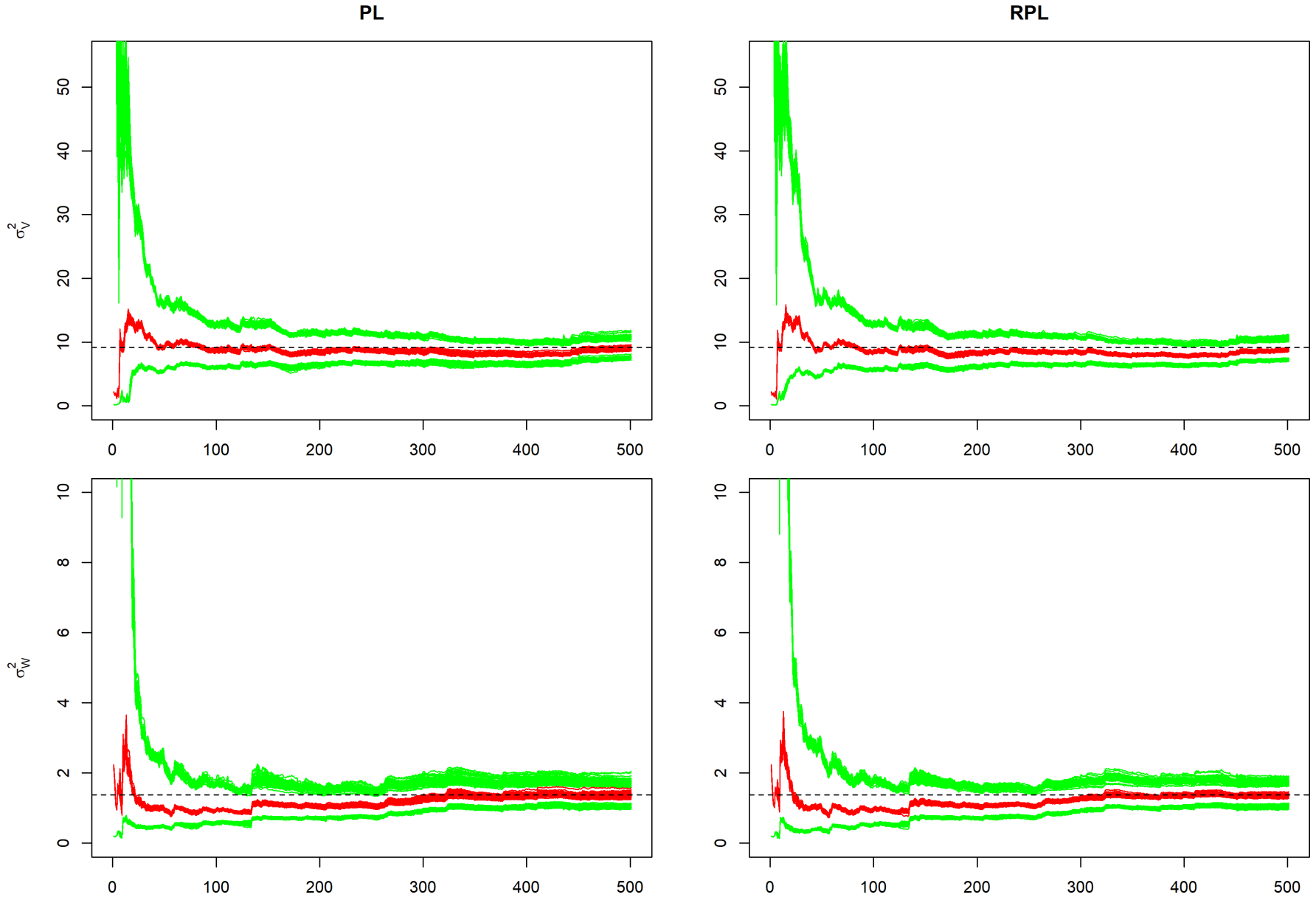

Figure 4: NLSM model (33-34): trace plots of the posterior distributions for σ_V^2 and σ_W^2 based on $M = 50$ independent runs of the Particle Learning (left column) and the Regularized Particle Learning (right column) methods. For each row, upper and lower solid green lines are the 2.5th and 97.5th posterior percentiles, and solid red lines are the posterior medians of PL or RPL at each time point. Black dashed lines are the pMCMC posterior means, and true parameter values are $\sigma_V^2 = 10$ and $\sigma_W^2 = 1$. The filters were run with $N = 50{,}000$ particles.

for each t, with $F_t := X_{t-1}/2 + 25X_{t-1}/(1+X_{t-1}^2) + 8\cos(1.2t)$ and starting at $a_0 = b_0 = c_0 = d_0 = 1$. Since the parameter space Θ is once again restricted, we have to work with log-variances $\check{\sigma}_V^2 = \log(\sigma_V^2)$ and $\check{\sigma}_W^2 = \log(\sigma_W^2)$ for the regularization steps.

Figure 3 contains the estimated marginal posteriors for each component of θ at $t = 500$ based on both PL and RPL methods, and Figure 4 contains the corresponding posterior traces at each time point. Visually, both the final posterior density estimates and their traces are very similar between both methods, with RPL showing slightly stabler estimates. The ESS, however, shows evidence of a more pronounced difference between the methods: ESSs for $\theta = (\sigma_V^2, \sigma_W^2)$ are $(16, 9)$ for PL and $(32, 18)$ for RPL (an average twofold improvement). Overall, both methods show agreement with the pMCMC estimates, with median posterior across chains converging to the pMCMC posterior means for all components of θ.

Regarding CPU time consumption, RPL is about three times slower than PL, with an average PL run taking 22.37 seconds and an average RPL run taking 62.09 seconds.

4.3 Gamma-Poisson Model

Now, we investigate the effect full-adaptation might have in sequential parameter inference, but combined with also regularizing past states, parameters and sufficient statistics. The algorithms we compare in this experiment are the Hybrid LW-PL of Chen et al. (2010; see also the Supplement) and the Hybrid FALW-RPL introduced in Section 3.3.3. The target model for inference is the *Gamma-Poisson* model (Jørgensen et al., 1996; Storvik, 2002).

Let

$$X_t|(X_{t-1}, \theta) \sim G(X_{t-1}/\sigma^2, 1/\sigma^2), \tag{35}$$
$$Y_t|(X_t, \theta) \sim \text{Pois}(\lambda X_t), \tag{36}$$

with priors

$$\lambda \sim G(1, 1), \qquad \sigma^2 \sim IG(3, 2),$$

where $\text{Pois}(\mu)$ denotes a Poisson distribution with mean μ.

Here the state space is $\mathcal{X} = \mathbb{R}_+$, the static parameter vector is $\theta = (\lambda, \sigma^2)$, and the parameter space is $\Theta = \mathbb{R}_+^2$. Also,

$$f(x_t^i|x_{t-1}^i, \theta_{t-1}^i) = dG\left(x_t^i \middle| \frac{x_{t-1}^i}{(\sigma^2)_{t-1}^i}, \frac{1}{(\sigma^2)_{t-1}^i}\right), \qquad g(y_t|x_t^i, \theta_{t-1}^i) = d\text{Pois}\big(y_t|\lambda_{t-1}^i \cdot x_t^i\big),$$

where $dG(x|a, b)$ and $d\text{Pois}(y|\mu)$ are the probability density functions of, respectively, $G(a, b)$ and $\text{Pois}(\mu)$ random variables, evaluated at x and y.

For this experiment, we take $\lambda = 0.50$ and $\sigma^2 = 0.40$ as the true parameter values, simulate a series of $n = 500$ observations and then perform $M = 50$ independent runs of the Hybrid LW-PL and Hybrid FALW-RPL methods using $N = 50{,}000$ particles. Here, the parameter for which we perform LW-type moves is $\phi = \sigma^2$ and the one for which we perform Gibbs-type moves is $\varphi = \lambda$. We chose $\delta = 0.99$ for the Hybrid LW-PL filter. Here, full adaptation is possible by choosing

$$\lambda_t^i \propto w_{t-1}^i p(y_t|x_{t-1}^i, \theta_{t-1}^i) = w_{t-1}^i d\text{NegBin}\Big(y_t \Big| \frac{1}{1 + (\sigma^2)_{t-1}^i \cdot \lambda_{t-1}^i}, y_t + \frac{x_{t-1}^i}{(\sigma^2)_{t-1}^i}\Big)$$

and

$$p(x_t^i|x_{t-1}^i, \theta_{t-1}^i, y_t) = dG\left(x_t^i \middle| y_t + \frac{x_{t-1}^i}{(\sigma^2)_{t-1}^i}, \lambda_{t-1}^i + \frac{1}{(\sigma^2)_{t-1}^i}\right),$$

where $d\text{NegBin}(x|p, r)$ denotes the probability density function of a Negative Binomial random variable at x with probability p and size r, i.e. with expectation $r \cdot p/(1-p)$. For the Hybrid LW-PL method, a lookahead APF strategy is adopted, with intermediate weights given by $\lambda_t^i \propto w_{t-1}^i g(y_t|\mu_{t-1}^i, m_{t-1}^i, \varphi_{t-1}^i)$ and blind state proposal $q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = f(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)$, with $\mu_{t-1}^i = x_{t-1}^i$ and m_{t-1}^i as defined in (22).

The posterior distribution for λ for the Gibbs sampling steps in both methods is available in closed form, and is given by

$$\lambda|(X_{0:t}, Y_{1:t}) \sim G(a_t, b_t),$$

where the sufficient statistics $\mathcal{S}_t = (a_t, b_t)$, satisfy

$$a_t := a_0 + \sum_{k=1}^{t} Y_k = a_0 + \sum_{k=1}^{t-1} Y_k + Y_t = a_{t-1} + Y_t,$$
$$b_t := b_0 + \sum_{k=1}^{t} X_k = b_0 + \sum_{k=1}^{t-1} X_k + X_t = b_{t-1} + X_t$$

for each t, starting at $a_0 = b_0 = 1$. Since the parameter space Θ is again restricted, we have to work with $\check{\lambda} = \log(\lambda)$ and $\check{\sigma}^2 = \log(\sigma^2)$ for the regularization steps.

Figure 5 contains the estimated marginal posteriors for each component of θ at $t = 500$ based on both Hybrid LW-PL and Hybrid FALW-RPL methods, and Figure 6 contains the corresponding posterior traces at each time point. Here the evidence in favor of full adaptation and regularization of past states, parameters and sufficient statistics is clearer, with the Hybrid LW-PL estimates displaying higher variability across runs than those of the Hybrid FALW-RPL algorithm, especially for the λ parameter (the one which we update via Gibbs-type moves). The ESS estimates corroborate this difference: ESSs for $\theta = (\lambda, \sigma^2)$ are $(67, 53)$ for the Hybrid LW-PL and $(85, 119)$ for the Hybrid FALW-RPL method. Posterior for both methods show agreement with the pMCMC histograms, and chains seem to converge to pMCMC posterior means.

Regarding CPU time consumption, an average Hybrid LW-PL run takes about 69.34 seconds, and an average Hybrid FALW-RPL run takes about 106.37 seconds.

4.4 AR(1) + Noise Model

For the next experiment, we illustrate the performance of all 3 algorithms proposed in this paper in the same scenario. Consider the state space model defined by

$$X_t = \phi X_{t-1} + \sigma_U U_t, \qquad U_t \sim \mathcal{N}(0, 1) \tag{37}$$
$$Y_t = X_t + \sigma_V V_t, \qquad V_t \sim \mathcal{N}(0, 1) \tag{38}$$

with priors

$$X_0|\sigma_U^2 \sim \mathcal{N}(0, \sigma_U^2), \qquad \phi|\sigma_U^2 \sim \mathcal{N}(0.50, \sigma_U^2), \qquad \sigma_U^2 =^d \sigma_V^2 \sim IG(1/2, 1/2).$$

Here we assume that $X_0 \perp\!\!\!\perp U_t \perp\!\!\!\perp V_s$ for all t, s, and that $(U_t)_{t\geq 0}$ and $(V_t)_{t\geq 0}$ are serially independent.

The state space here is $\mathcal{X} = \mathbb{R} := (-\infty, +\infty)$ and the static parameter is $\theta = (\phi, \sigma_U^2, \sigma_V^2)$, taking values in $\Theta = \mathbb{R} \times \mathbb{R}_+^2$. Since equations (37-38) describe a (Gaussian) autoregressive process $(X_t)_{t\geq 0}$ of order 1 observed via $(Y_t)_{t\geq 0}$ with (Gaussian) noise $\sigma_V V_t$, the model is commonly referred to as the (Gaussian) *AR(1) + noise* model. Here $f(x_t^i|x_{t-1}^i, \theta_{t-1}^i) = d\mathcal{N}(x_t^i|\phi_{t-1}^i x_{t-1}^i, (\sigma_U^2)_{t-1}^i)$ and $g(y_t|x_t^i, \theta_{t-1}^i) = d\mathcal{N}(y_t|x_t^i, (\sigma_V^2)_{t-1}^i)$.

We reproduce here the specific configuration adopted by Nemeth et al. (2016). By taking $\phi = 0.90$, $\sigma_U^2 = 0.70^2$ and $\sigma_V^2 = 1$ as the true parameter values, we simulate a series of size $n = 1{,}000$ of model (37-38) and then perform $M = 50$ independent runs of the FALW (Section 3.3.1), RPL (Section 3.3.2) and Hybrid FALW-RPL (Section 3.3.3) methods, using $N = 50{,}000$ particles. Here, we adopted the rule-of-thumb bandwidth of Silverman (1986) with a Gaussian kernel for all of the methods.

In this model, the optimal importance weights and state proposal distribution are available in closed-form as

$$\lambda_t^i \propto w_{t-1}^i p(y_t|x_{t-1}^i, \theta_{t-1}^i) = w_{t-1}^i d\mathcal{N}(\phi_{t-1}^i x_{t-1}^i, (\sigma_U^2)_{t-1}^i + (\sigma_V^2)_{t-1}^i)$$

and

$$p(x_t|x_{t-1}^i, \theta_{t-1}^i, y_t) = d\mathcal{N}\left(x_t \middle| \frac{(\sigma_U^2)_{t-1}^i \cdot y_t + (\sigma_V^2)_{t-1}^i \cdot \phi_{t-1}^i x_{t-1}^i}{(\sigma_U^2)_{t-1}^i + (\sigma_V^2)_{t-1}^i}, \frac{(\sigma_U^2)_{t-1}^i \cdot (\sigma_V^2)_{t-1}^i}{(\sigma_U^2)_{t-1}^i + (\sigma_V^2)_{t-1}^i}\right).$$

We adopt these weights and state proposals for all FALW, RPL and Hybrid FALW-RPL methods.

The posterior distributions for ϕ, σ_U^2 and σ_V^2 given the states and observations are also available in closed-form, and are given by

$$\phi|(X_{0:t}, Y_{1:t}, \sigma_U^2) \sim \mathcal{N}(m_t, \sigma_U^2 \cdot C_t), \quad \sigma_U^2|(X_{0:t}, Y_{1:t}) \sim IG(a_t/2, b_t/2), \quad \sigma_V^2|(X_{0:t}, Y_{1:t}) \sim IG(c_t/2, d_t/2),$$

where the sufficient statistics $\mathcal{S}_t = (m_t, C_t, a_t, b_t, c_t, d_t)$ satisfy, for $t \geq 1$,

$$m_t := m_{t-1} + \frac{1}{D_t} C_{t-1} F_t (X_t - F_t m_{t-1}),$$

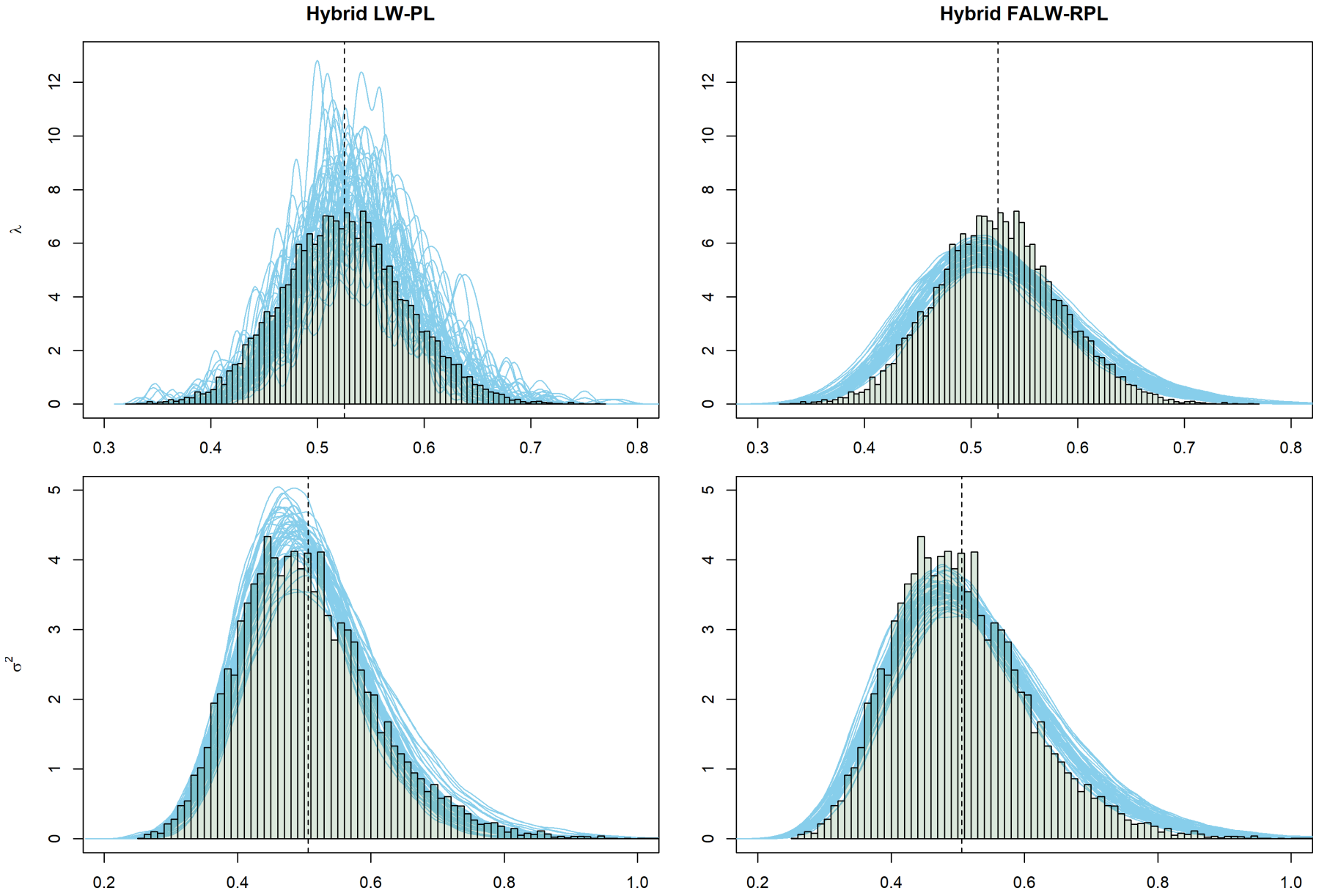

Figure 5: Gamma-Poisson model (35-36): kernel density estimates (solid blue lines) of the posterior distributions at $t = 500$ for λ and σ^2 based on $M = 50$ independent runs of the Hybrid LW-PL (left column) and the Hybrid FALW-RPL (right column) methods. The filters were run with $N = 50{,}000$ particles, and the true parameter values are $\lambda = 0.50$ and $\sigma^2 = 0.40$. Histogram bars are from the corresponding pMCMC run, and vertical dashed lines are the pMCMC posterior means.

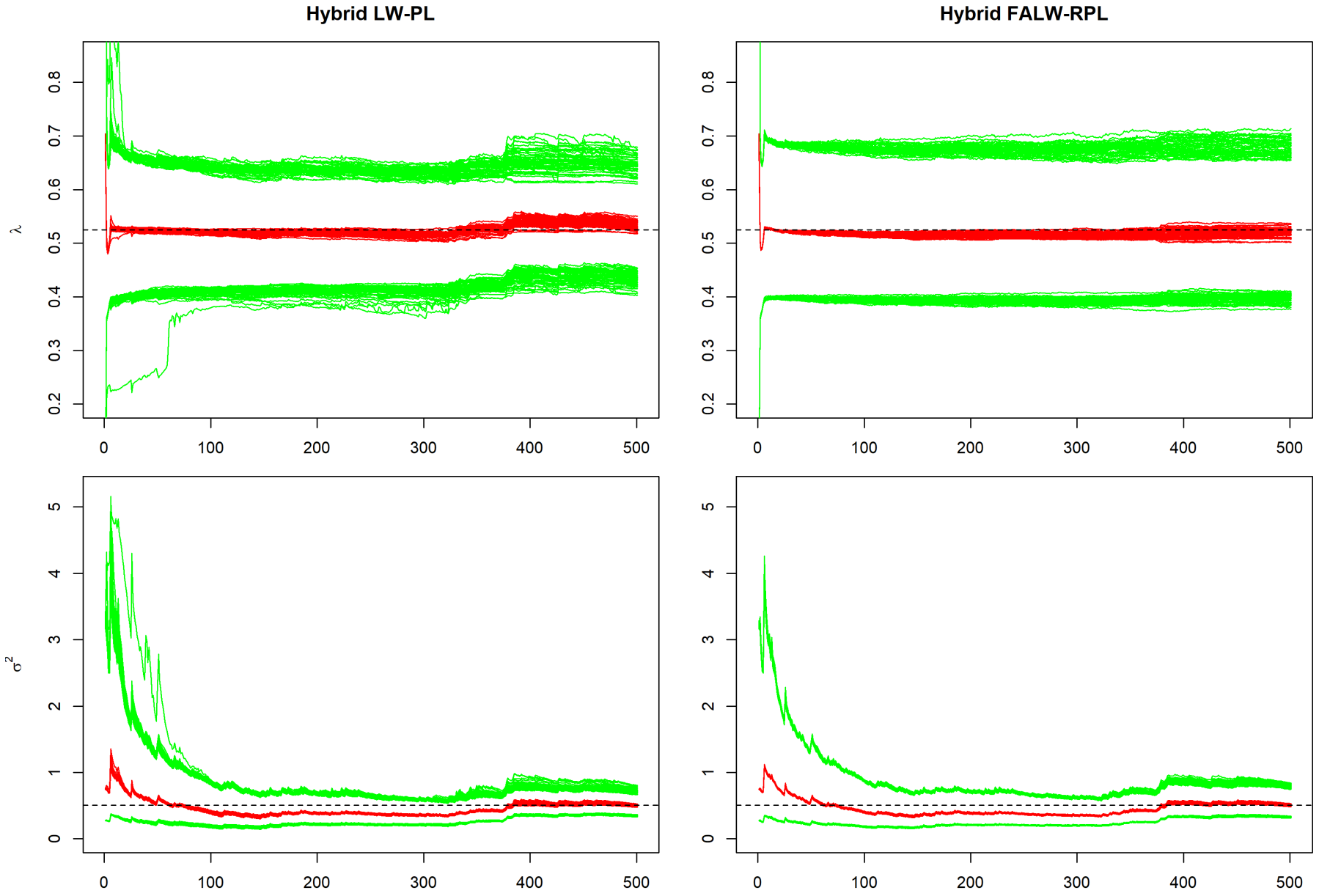

Figure 6: Gamma-Poisson model (35-36): trace plots of the posterior distributions for λ and σ^2 based on $M = 50$ independent runs of the Hybrid LW-PL (left column) and the Hybrid FALW-RPL (right column) methods. For each row, upper and lower solid green lines are the 2.5th and 97.5th posterior percentiles, and solid red lines are the posterior medians of each method at each time point. Black dashed lines are the pMCMC posterior means, and true parameter values are $\lambda = 0.50$ and $\sigma^2 = 0.40$. The filters were run with $N = 50{,}000$ particles.

$$
\begin{aligned}
C_t &:= C_{t-1} - \frac{1}{D_t} C_{t-1}^2 F_t^2, \\
a_t &:= a_{t-1} + 1, \\
b_t &:= b_{t-1} + \frac{1}{D_t}(X_t - F_t m_{t-1})^2, \\
c_t &:= c_{t-1} + 1, \\
d_t &:= b_{t-1} + (Y_t - X_t)^2,
\end{aligned}
$$

with $F_t = X_{t-1}$ and $D_t = C_{t-1}F_t^2 + 1$, and initial conditions $m_0 = 0.50$, $C_0 = 1$ and $a_0 = b_0 = c_0 = d_0 = 1$. See Storvik (2002) for a complete derivation of these recursions.

Since the parameter space Θ in this model is also restricted, we work with the transformed parameters $\check{\sigma}_U^2 = \log(\sigma_U^2)$ and $\check{\sigma}_V^2 = \log(\sigma_V^2)$. Finally, for the Hybrid FALW-RPL method we chose to perform Gibbs-type moves for (ϕ, σ_U^2) and LW-type moves for σ_V^2, and therefore for this method we only compute and regularize the sufficient statistics m_t, C_t, a_t and b_t.

Figure 7 contains the estimated marginal posteriors for each component of θ at $t = 1{,}000$ based on the FALW, Hybrid-FALW and RPL methods, and Figure 8 contains the corresponding posterior traces at each time point. Qualitatively, the performance for the three algorithms is strikingly similar, with FALW showing a slightly greater variability compared to the other ones. All of the methods agree with the pMCMC posterior distributions, and converge to the pMCMC posterior means.

The ESSs show a numerical difference between all three methods, with RPL being more consistent across runs. Respectively, for $\theta = (\phi, \sigma_U^2, \sigma_V^2)$, the ESSs for FALW are $(128, 80, 140)$, the ESSs for Hybrid FALW-RPL are $(173, 102, 164)$ and the ESSs for RPL are $(178, 105, 202)$. This is consistent with previous findings (see e.g. Carvalho et al. 2010 for some examples and further references) and, as argued before, indicates that Gibbs-type moves should be performed whenever posterior distributions for θ given sufficient statistics are available (along with regularization and efficient resampling, as already discussed in Section 3.2). With respect to CPU time consumption, an average FALW run takes about 79.93 seconds, a Hybrid FALW-RPL run takes 145.33 seconds and a RPL run takes 166.57 seconds (about twice the average time for a FALW run).

4.5 Stochastic Volatility Model

For our final experiment, we consider inference on a *Stochastic Volatility* (SV) model (Taylor, 1982; Dahlin and Schön, 2019) using real-world data from the NASDAQ OMXS30 index.

Let

$$
\begin{aligned}
X_t &= \phi X_{t-1} + \tau U_t, & U_t &\sim \mathcal{N}(0,1), & (39)\\
Y_t &= \sigma \exp(X_t/2) V_t, & V_t &\sim \mathcal{N}(0,1), & (40)
\end{aligned}
$$

with priors

$$
X_0|\tau^2 \sim \mathcal{N}(0, \tau^2), \quad \phi|\tau^2 \sim \mathcal{N}(0.9356, 0.10 \cdot \tau^2), \quad \tau^2 \sim IG(8/2, 0.12/2), \quad \sigma^2 \sim IG(5.1640/2, 5.2164/2).
$$

Here, once again $(U_t)_{t\geq 0}$ and $(V_t)_{t\geq 0}$ are assumed to be serially and mutually independent, and also independent of X_0. These particular prior hyperparameter values comes from matching the first and second moments of the experiment in the priors used in Dahlin and Schön (2019), although the actual priors used here were chosen to allow for closed-form posterior distributions for performing Gibbs-type updates.

The state space for the SV model is $\mathcal{X} = \mathbb{R}$, the static parameters are $\theta = (\phi, \tau^2\sigma^2)$ and the parameter space is $\Theta = \mathbb{R} \times \mathbb{R}_+^2$. Here,

$$
f(x_t^i|x_{t-1}^i, \theta_{t-1}^i) = d\mathcal{N}(x_t^i|\phi_{t-1}^i x_{t-1}^i, (\tau^2)_{t-1}^i), \qquad g(y_t|x_t^i, \theta_{t-1}^i) = d\mathcal{N}(y_t|0, (\sigma^2)_{t-1}^i \exp(x_t^i/2)).
$$

For this experiment, we use the daily series of stock prices of the NASDAQ OMXS30 index from January 2, 2012 to January 2, 2014. For each day t after January 2, 2012, the daily log-return is defined by $y_t := \log(P_t/P_{t-1})$, where P_t is the daily closing price, $t = 1, \ldots, 500$. We perform $M = 50$ independent runs of the Regularized Particle Learning method (RPL) of Section 3.3.2 using $N = 50{,}000$ particles. The design choices we make for the RPL method are SIR intermediate weights $\lambda_t^i = w_{t-1}^i$ and blind state proposal $q(x_t^i|\tilde{x}_{0:t-1}^i, \tilde{\theta}_{0:t-1}^i, y_{1:t}) = f(x_t|\tilde{x}_{t-1}^i, \tilde{\theta}_{t-1}^i)$.

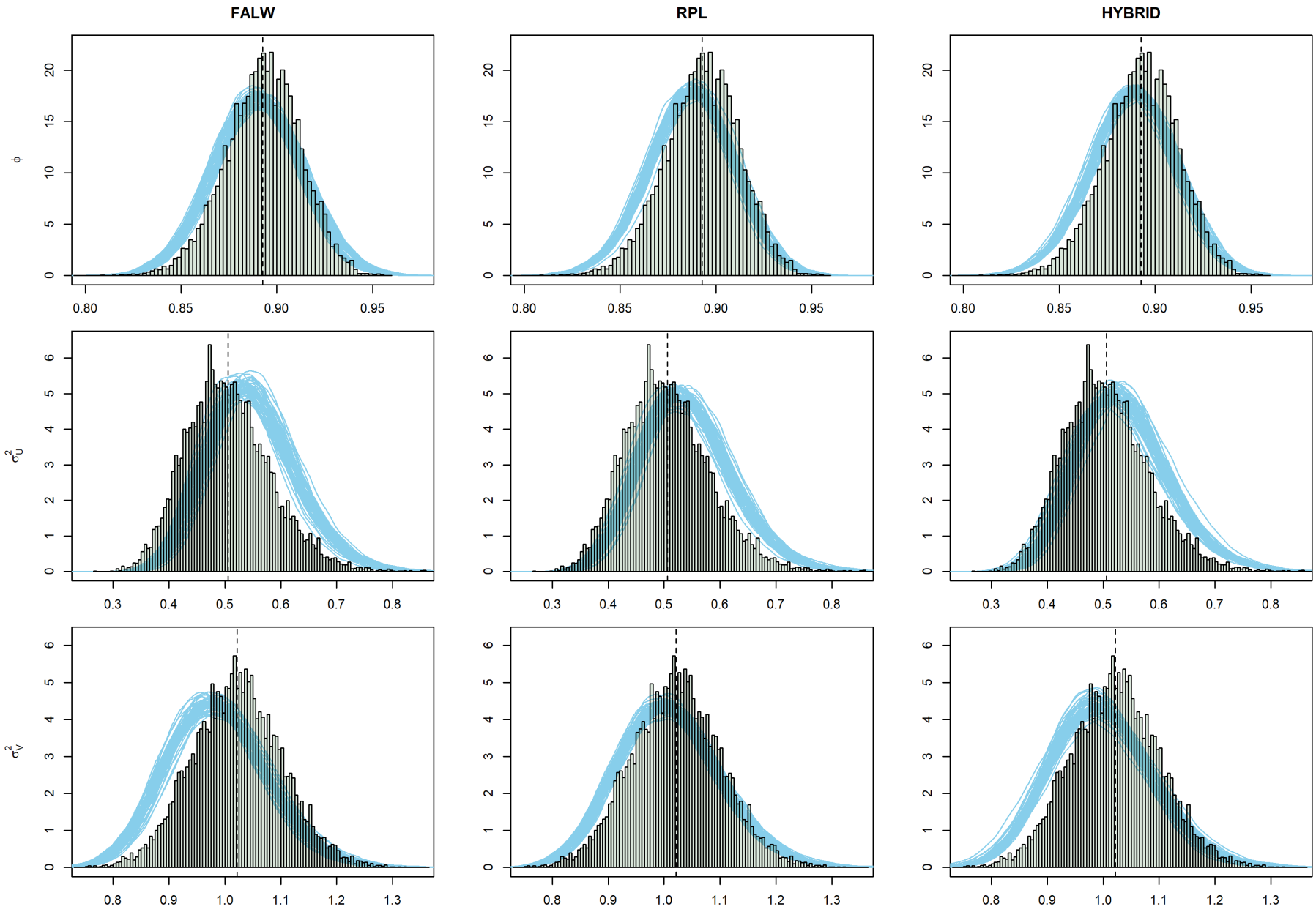

Figure 7: AR(1) + noise model (37-38): kernel density estimates (solid blue lines) of the posterior distributions at $t = 1{,}000$ for ϕ, σ_U^2 and σ_V^2 based on $M = 50$ independent runs of the FALW (left column), RPL (center column) and the Hybrid FALW-RPL (right column) methods. The filters were run with $N = 50{,}000$ particles, and the true parameter values are $\phi = 0.90$, $\sigma_U^2 = 0.70^2 = 0.49$ and $\sigma_V^2 = 1$. Histogram bars are from the corresponding pMCMC run, and vertical dashed lines are the pMCMC posterior means.

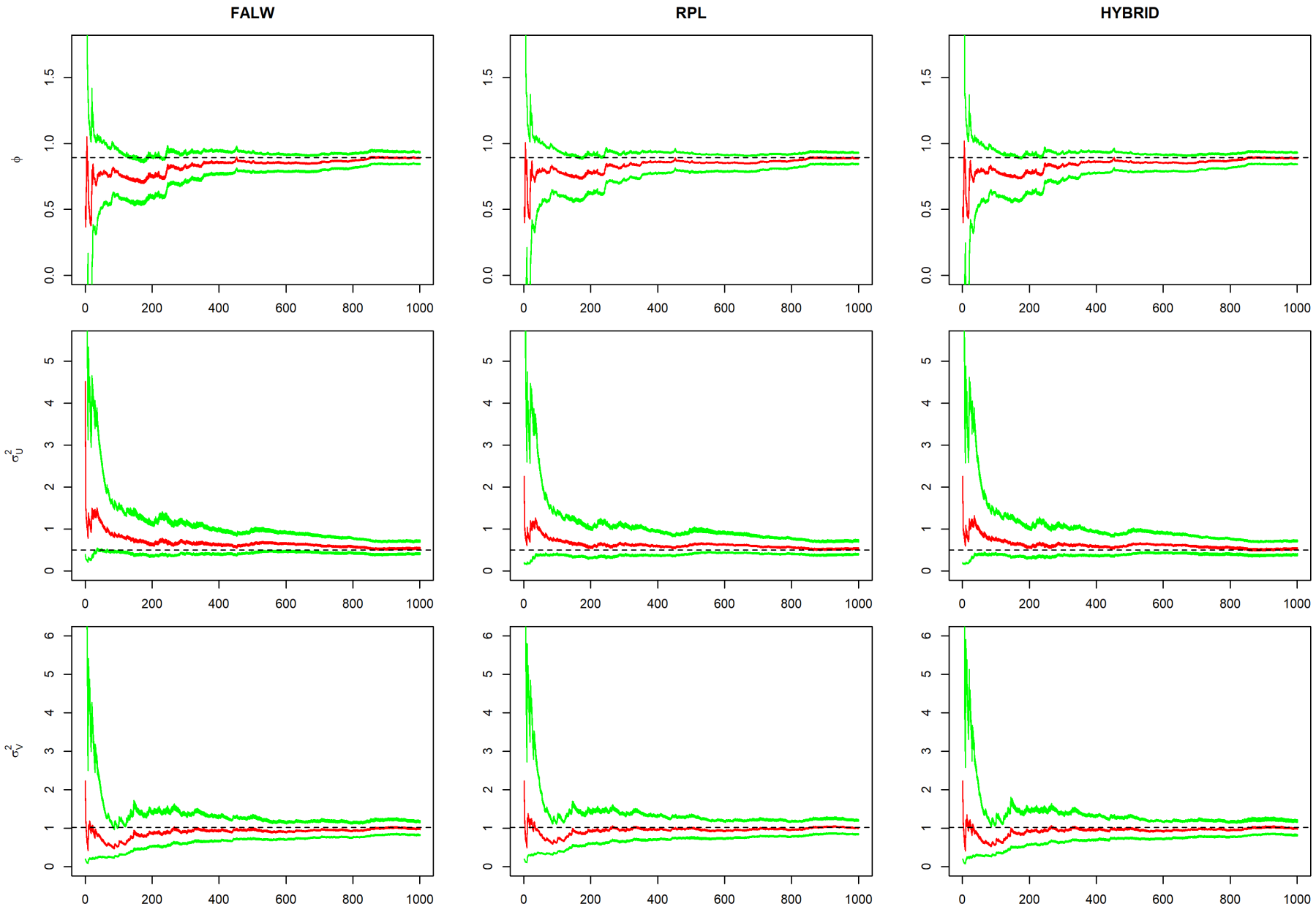

Figure 8: AR(1) + noise model (37-38): trace plots of the posterior distributions for ϕ, σ_U^2 and σ_V^2 based on $M = 50$ independent runs of the FALW (left column), RPL (center column) and the Hybrid FALW-RPL (right column) methods. For each row, upper and lower solid green lines are the 2.5th and 97.5th posterior percentiles, and solid red lines are the posterior medians of each method at each time point. Black dashed lines are the pMCMC posterior means, and true parameter values are $\phi = 0.90$, $\sigma_U^2 = 0.70^2 = 0.49$ and $\sigma_V^2 = 1$. The filters were run with $N = 50{,}000$ particles.

The posterior distributions for performing the Gibbs sampling steps in RPL are available in closed form for all of the parameters. They are given by

$$\tau^2|(X_{0:t}, Y_{1:t}) \sim IG(a_t/2, b_t/2), \quad \phi|(X_{0:t}, Y_{1:t}, \tau^2) \sim \mathcal{N}(m_t, \tau^2 C_t), \quad \sigma^2|(X_{0:t}, Y_{1:t}) \sim IG(c_t/2, d_t/2),$$

where the sufficient statistics $\mathcal{S}_t = (m_t, C_t, a_t, b_t, c_t, d_t)$, satisfy

$$\begin{aligned}
m_t &= m_{t-1} + C_{t-1} \cdot \frac{F_t}{D_t} \cdot (x_t - F_t m_{t-1}), \\
C_t &= C_{t-1} - C_{t-1}^2 \cdot \frac{F_t^2}{D_t}, \\
a_t &= a_{t-1} + 1, \\
b_t &= b_{t-1} + \frac{(x - F_t m_{t-1})^2}{D_t}, \\
c_t &= c_{t-1} + 1, \\
d_t &= \frac{Y_t^2}{\exp(X_t)},
\end{aligned}$$

for each t, with $F_t := X_{t-1}$ and $D_t := C_{t-1} \cdot F_t^2$, starting at $m_0 = 0.9356, C_0 = 0.10, a_0 = 8, b_0 = 0.12, c_0 = 5.1640, d_0 = 5.2164$ (see also Storvik, 2002). Since the parameter space Θ is once again restricted, we have to work with log-variances $\check{\tau}^2 = \log(\tau^2)$ and $\check{\sigma}^2 = \log(\sigma^2)$ for the regularization steps.

Figure 9 contains the estimated marginal posteriors for each component of θ at $t = 500$ based on the RPL method, as well as the corresponding posterior traces at each time point. The estimates show a certain regularity across runs, and the ESSs for $\theta = (\phi, \tau^2, \sigma^2)$ are $(153, 222, 247)$. The corresponding median across runs of the posterior estimates (which are the posterior expectations in each run) are $(1.0773, 0.9743, 0.0202)$, similar to those found by Dahlin and Schön (2019)(although they rely on a different parameterization and technique). The RPL posteriors also agree with the pMCMC posterior distributions, and seem to converge to the pMCMC posterior means.

Figure 10 contains the log-returns and the boxplots of posterior state expectations in each run for each day. These results are also qualitatively similar to those of Dahlin and Schön (2019), and are consistent with the parameter estimates obtained here: log-volatilities are slow-moving due to their high autocorrelation (given by ϕ) and relatively small variance (given by τ^2). Average CPU consumption in this experiment was about 83.12 seconds for each RPL run.

5 Conclusions

In this paper we introduced a framework for sequential parameter learning in state space models capable of accommodating several other algorithms found in the literature as special cases. In order to illustrate the flexibility and usefulness of the proposed framework, we also developed three new algorithms: an improved and fully-adapted version of Liu and West (2001)'s filter, a regularized version of Carvalho et al. (2010)'s Particle Learning method and a hybrid between the two inspired by the ideas of Chen et al. (2010).

To analyze the performance of the sequential parameter learning methods proposed in this work, we considered a series of simulation-based numerical experiments. First, we examined the performance of the Liu and West (2001)'s filter and its fully-adapted counterpart introduced here in a highly nonlinear θ-logistic model (Peters et al., 2010; Polansky et al., 2009). Afterwards, we assessed inference using the Particle Learning method of Carvalho et al. (2010) and its regularized counterpart developed here in the Nonlinear Seasonal State Space Model (Netto et al., 1978; Andrieu et al., 2010). Next, we also explored the performance of the hybrid filter of Chen et al. (2010) and our fully-adapted version of this filter in the context of a Gamma-Poisson model (Jørgensen et al., 1996). Finally, we compared all three proposed algorithms together in the context of an AR(1) + noise model (Nemeth et al., 2016). To add to these experiments using simulation-based data, we also estimated a Stochastic Volatility model using real-world data from the NASDAQ OMXS30 index (Dahlin and Schön, 2019; McTaggart et al., 2019). Further experiments using the methods proposed here can also be found in Silva (2020).

In closing, we note that although much of the content of this paper has already been discussed and presented elsewhere, to our knowledge no effort has been made to unify such a diverse set of methods

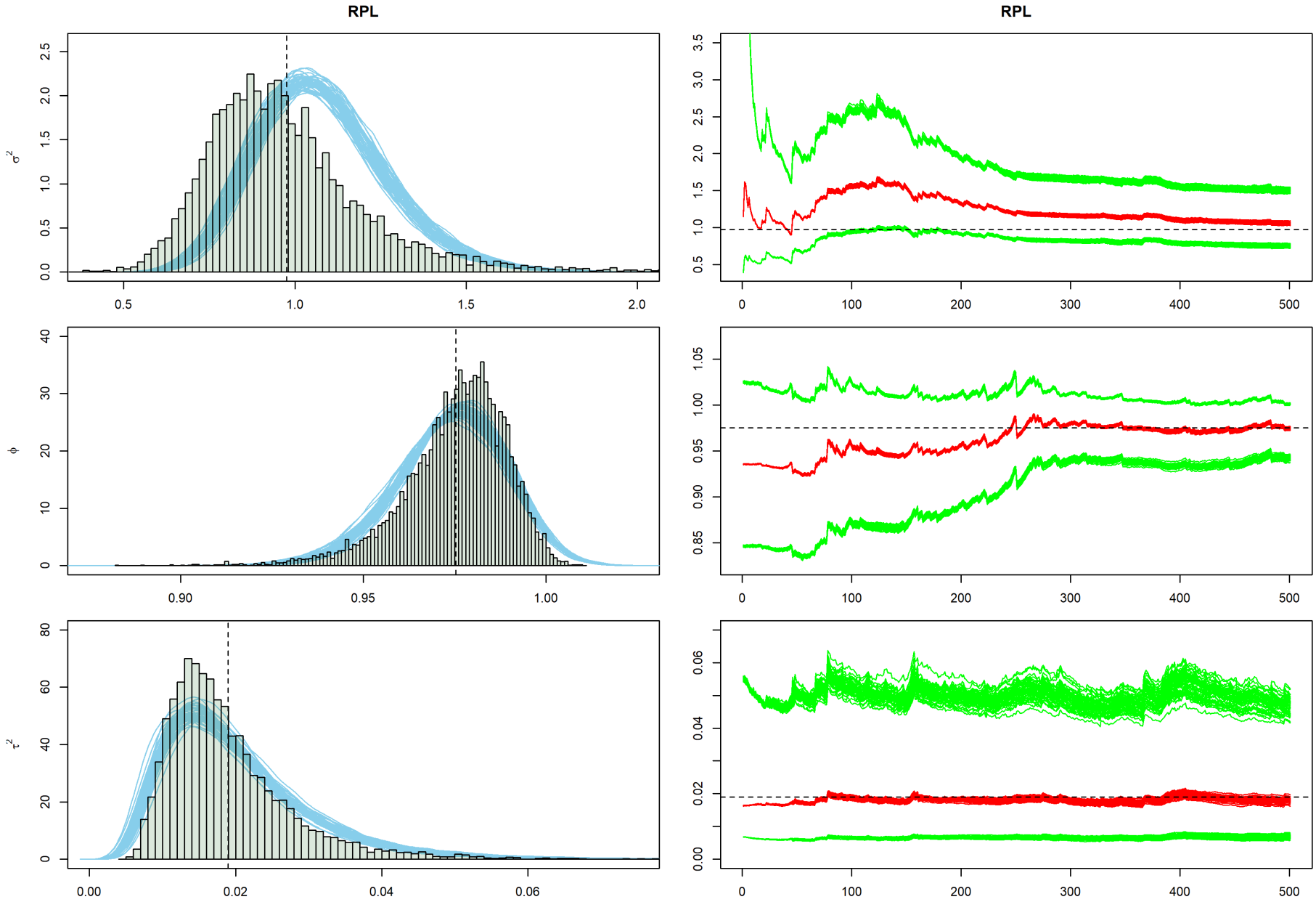

Figure 9: SV model (39-40): kernel density estimates (left column) and trace plots (right column) for ϕ, τ^2 and σ^2 based on $M = 50$ independent runs of the Regularized Particle Learning method. The kernel density estimates (solid blue lines) are for the RPL posterior distributions at $t = 500$. Histogram bars are from the corresponding pMCMC run, and vertical dashed lines are the pMCMC posterior means. The trace plots contain the 2.5th and 97.5th RPL posterior percentiles (solid green lines) and posterior medians (solid red lines) at each time point. Black dashed lines are the pMCMC posterior means. The filters were run with $N = 50{,}000$ particles.

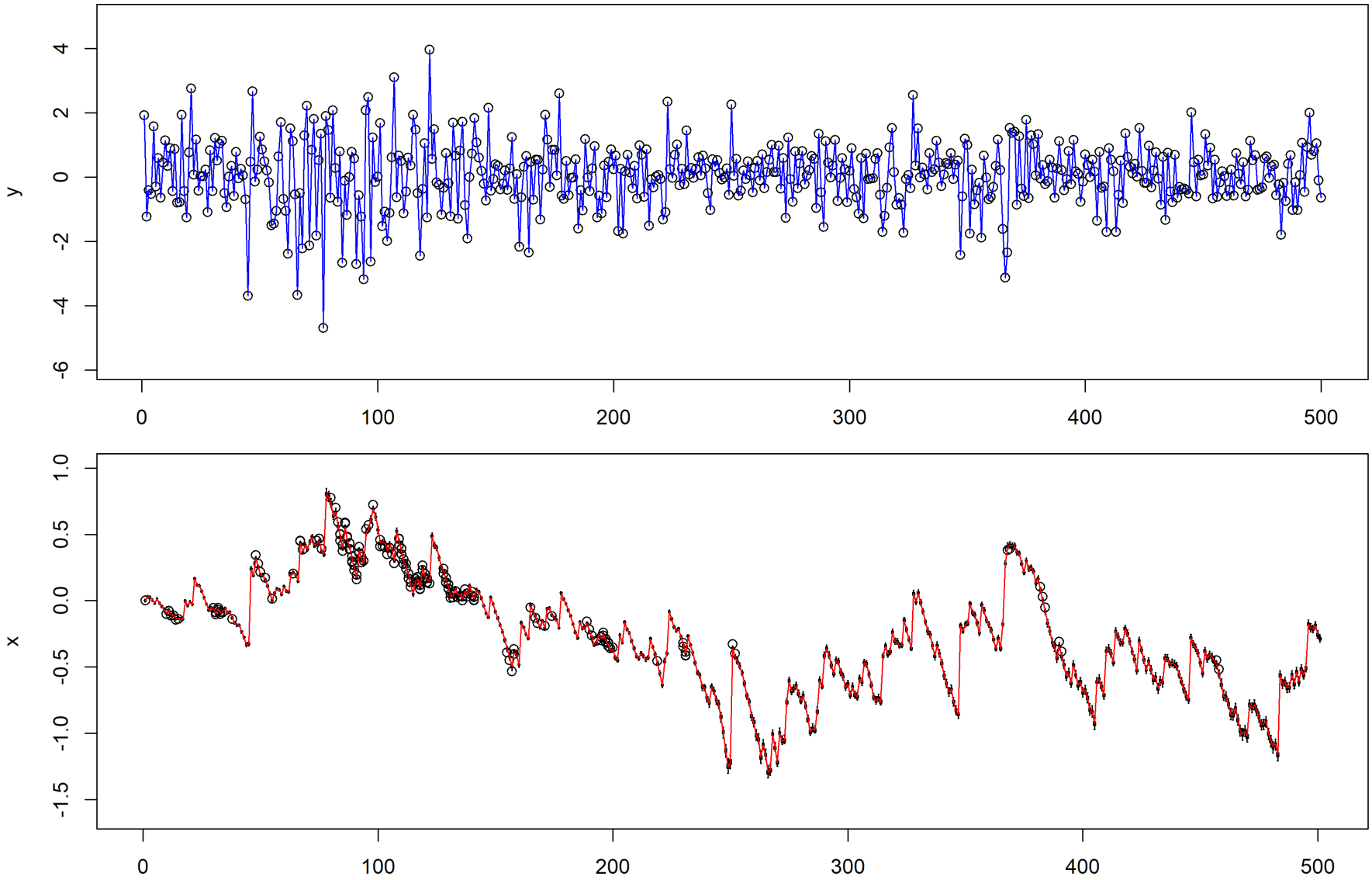

Figure 10: SV model (39-40): log-returns (top row) and posterior log-volatility estimates (bottom row) at each time point. In the top row, black points and solid blue lines represent the daily log-returns of the NASDAQ OMXS30 index in the period from January 2, 2012 to January 2, 2014. In the bottom row, box plots of the posterior estimates for each of the $M = 50$ independent runs of the Regularized Particle Learning method are in black, with the corresponding medians as red solid lines. The filters were run with $N = 50{,}000$, and for each run the posterior estimates taken to compute the boxplots are the posterior means of the states at each time point.

in a single framework. We single out this effort as the main contribution of our work. Moreover, our reinterpretation of the role of regularization in the sequential parameter learning setting, inspired by the works of Liu and West (2001) and Andrieu et al. (2010), has not appeared anywhere else in the literature.

For future work, hopefully the formalism developed for the unified framework proposed here will allow for further exploration of the analytical and theoretical properties of general sequential parameter learning algorithms, and also lead to a better understanding of the relationship between these methods. As far as the practical aspect of these methods is concerned, directions for future research also include drawing from the results on kernel density estimation literature (such as local or adaptive kernel bandwidths; see e.g. Silverman 1986 for an early review) in order to improve upon the regularization techniques typically employed in these algorithms.

6 Acknowledgments

Uriel Silva's research was partially funded by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and Luiz Duczmal's research was partially funded by Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), grant PPM-00596-17. We sincerely thank the Associate Editor and two anonymous referees for their careful reading and insightful comments that led to significant improvements in the paper.

References

Andrieu, C., De Freitas, N., and Doucet, A. (1999). Sequential MCMC for Bayesian model selection. In *Proceedings of the IEEE Signal Processing Workshop on Higher-Order Statistics. SPW-HOS'99*, pages 130–134. IEEE.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C., Doucet, A., and Tadic, V. B. (2005). On-line Parameter Estimation in General State-space Models. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 332–337. IEEE.

BøLviken, E., Acklam, P. J., Christophersen, N., and StøRdal, J.-M. (2001). Monte Carlo Filters for Non-linear State Estimation. *Automatica*, 37(2):177–183.

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer Science & Business Media.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved Particle Filter for Nonlinear Problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7.

Carvalho, C. M., Johannes, M. S., Lopes, H. F., Polson, N. G., et al. (2010). Particle Learning and Smoothing. *Statistical Science*, 25(1):88–106.

Carvalho, C. M. and Lopes, H. F. (2007). Simulation-based Sequential Analysis of Markov Switching Stochastic Volatility Models. *Computational Statistics & Data Analysis*, 51(9):4526–4542.

Chen, H., Petralia, F., and Lopes, H. F. (2010). Sequential Monte Carlo Estimation of DSGE Models. Technical report, The University of Chicago Booth School of Business.

Chopin, N. (2002). A Sequential Particle Filter Method for Static Models. *Biometrika*, 89(3):539–552.

Chopin, N. et al. (2004). Central Limit Theorem for Sequential Monte Carlo Methods and its Application to Bayesian Inference. *The Annals of Statistics*, 32(6):2385–2411.

Chopin, N., Iacobucci, A., Marin, J.-M., Mengersen, K., Robert, C. P., Ryder, R., and Schäfer, C. (2010). On Particle Learning. *arXiv preprint arXiv:1006.0554*.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). SMC^2: an Efficient Algorithm for Sequential Analysis of State Space Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426.

Crisan, D. and Doucet, A. (2002). A Survey of Convergence Results on Particle Filtering Methods for Practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746.

Crisan, D. and Lyons, T. (2002). Minimal Entropy Approximations and Optimal Algorithms. *Monte Carlo Methods and Applications*, 8(4):343–356.

Dahlin, J. and Schön, T. B. (2019). Getting Started with Particle Metropolis-Hastings for Inference in Nonlinear Dynamical Models. *Journal of Statistical Software*, 88(CN2):1–41.

Del Moral, P. (2004). *Feynman-Kac Formulae*. Springer.

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

Doucet, A. and Johansen, A. M. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. *Handbook of Nonlinear Filtering*, 12(656-704):3.

Dukic, V., Lopes, H. F., and Polson, N. G. (2012). Tracking Epidemics with Google Flu Trends Data and a State-space SEIR Model. *Journal of the American Statistical Association*, 107(500):1410–1426.

Durbin, J. and Koopman, S. J. (2012). *Time Series Analysis by State Space Methods*. Oxford University Press.

Fearnhead, P. (2002). Markov Chain Monte Carlo, Sufficient Statistics, and Particle Filters. *Journal of Computational and Graphical Statistics*, 11(4):848–862.

Flury, T. and Shephard, N. (2009). Learning and Filtering via Simulation: Smoothly Jittered Particle Filters. Economics Series Working Papers 469, University of Oxford, Department of Economics.

Fulop, A. and Li, J. (2013). Efficient Learning via Simulation: A Marginalized Resample-move Approach. *Journal of Econometrics*, 176(2):146–161.

Ghaeminia, M. H., Shabani, A. H., and Shokouhi, S. B. (2010). Adaptive Motion Model for Human Tracking Using Particle Filter. In *2010 20th International Conference on Pattern Recognition*, pages 2073–2076. IEEE.

Gilks, W. R. and Berzuini, C. (2001). Following a Moving Target: Monte Carlo Inference for Dynamic Bayesian Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.

Golightly, A. and Wilkinson, D. J. (2006). Bayesian Sequential Inference for Nonlinear Multivariate Diffusions. *Statistics and Computing*, 16(4):323–338.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113.

Jacquier, E., Polson, N., and Sokolov, V. (2016). Sequential Bayesian Learning for Merton's Jump Model with Stochastic Volatility. *arXiv preprint arXiv:1610.09750*.

Johansen, A. M. and Doucet, A. (2008). A note on auxiliary particle filters. *Statistics & Probability Letters*, 78(12):1498–1504.

Jørgensen, B., Lundbye-Christensen, S., Song, X.-K., and Sun, L. (1996). A Longitudinal Study of Emergency Room Visits and Air Pollution for Prince George, British Columbia. *Statistics in Medicine*, 15(7-9):823–836.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). On Particle Methods for Parameter Estimation in State-space Models. *Statistical Science*, 30(3):328–351.

Kitagawa, G. (1987). Non-Gaussian State—space Modeling of Nonstationary Time Series. *Journal of the American Statistical Association*, 82(400):1032–1041.

Kitagawa, G. (1998). A Self-organizing State-space Model. *Journal of the American Statistical Association*, pages 1203–1215.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*, 89(425):278–288.

Liang, C. and Piché, R. (2010). Mobile Tracking and Parameter Learning in Unknown Non-line-of-sight Conditions. In *2010 13th International Conference on Information Fusion*, pages 1–6. IEEE.

Lin, J. and Ludkovski, M. (2014). Sequential Bayesian Inference in Hidden Markov Stochastic Kinetic Models with Application to Detection and Response to Seasonal Epidemics. *Statistics and Computing*, 24(6):1047–1062.

Liu, J. and West, M. (2001). Combined Parameter and State Estimation in Simulation-based Filtering. In *Sequential Monte Carlo Methods in Practice*, pages 197–223. Springer.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93(443):1032–1044.

Liu, Y.-Y., Li, S., Li, F., Song, L., and Rehg, J. M. (2015). Efficient Learning of Continuous-time Hidden Markov Models for Disease Progression. In *Advances in Neural Information Processing Systems*, pages 3600–3608.

Lopes, H. F. and Tsay, R. S. (2011). Particle Filters and Bayesian Inference in Financial Econometrics. *Journal of Forecasting*, 30(1):168–209.

McTaggart, R., Daroczi, G., and Leung, C. (2019). Quandl: API Wrapper for Quandl.com.

Musso, C., Oudjane, N., and Le Gland, F. (2001). Improving Regularised Particle Filters. In *Sequential Monte Carlo Methods in Practice*, pages 247–271. Springer.

Nemeth, C., Fearnhead, P., and Mihaylova, L. (2013). Sequential Monte Carlo Methods for State and Parameter Estimation in Abruptly Changing Environments. *IEEE Transactions on Signal Processing*, 62(5):1245–1255.

Nemeth, C., Fearnhead, P., and Mihaylova, L. (2016). Particle Approximations of the Score and Observed Information Matrix for Parameter Estimation in State-space Models with Linear Computational Cost. *Journal of Computational and Graphical Statistics*, 25(4):1138–1157.

Netto, M. A., Gimeno, L., and Mendes, M. (1978). A New Spline Algorithm for Non-linear Filtering of Discrete Time Systems. *IFAC Proceedings Volumes*, 11(1):2123–2130.

Peters, G. W., Hosack, G. R., and Hayes, K. R. (2010). Ecological Non-linear State Space Model Selection via Adaptive Particle Markov Chain Monte Carlo (AdPMCMC). *arXiv preprint arXiv:1005.2238*.

Petetin, Y. and Desbouvries, F. (2013). Optimal SIR Algorithm vs. Fully Adapted Auxiliary Particle Filter: a Non Asymptotic Analysis. *Statistics and Computing*, 23(6):759–775.

Pitt, M. K. and Shephard, N. (1999). Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590–599.

Polansky, L., De Valpine, P., Lloyd-Smith, J. O., and Getz, W. M. (2009). Likelihood Ridges and Multimodality in Population Growth Rate Models. *Ecology*, 90(8):2313–2320.

Polson, N. G., Stroud, J. R., and Müller, P. (2008). Practical Filtering with Sequential Parameter Learning. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):413–428.

Reichenberg, R. (2018). Dynamic Bayesian Networks in Educational Measurement: Reviewing and Advancing the State of the Field. *Applied Measurement in Education*, 31(4):335–350.

Rodeiro, C. L. V. and Lawson, A. B. (2006). Online Updating of Space-time Disease Surveillance Models via Particle Filters. *Statistical Methods in Medical Research*, 15(5):423–444.

Silva, U. M. (2020). *A General Framework for Sequential Parameter Learning With Regularization*. PhD thesis, Universidade Federal de Minas Gerais.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, volume 26. CRC Press.

Storvik, G. (2002). Particle Filters for State-space Models with the Presence of Unknown Static Parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289.

Taylor, S. J. (1982). Financial Returns Modelled by the Product of Two Stochastic Processes – a Study of the Daily Sugar Prices 1961-75. *Time Series Analysis: Theory and Practice*, 1:203–226.

Vercauteren, T., Toledo, A. L., and Wang, X. (2005). Online Bayesian Estimation of Hidden Markov Models with Unknown Transition Matrix and Applications to IEEE 802.11 Networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005 (ICASSP'05)*, volume 4, pages IV–13. IEEE.

Virbickaitė, A., Lopes, H. F., Concepción Ausín, M., and Galeano, P. (2019). Particle Learning for Bayesian Semi-parametric Stochastic Volatility Model. *Econometric Reviews*, 38(9):1007–1023.

Wang, W.-p., Liao, S., and Xing, T.-w. (2009). Particle Filter for State and Parameter Estimation in Passive Ranging. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 3, pages 257–261. IEEE.

Warty, S. P., Lopes, H. F., and Polson, N. G. (2018). Sequential Bayesian Learning for Stochastic Volatility with Variance-gamma Jumps in Returns. *Applied Stochastic Models in Business and Industry*, 34(4):460–479.

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. Springer Series in Statistics. Springer, New York, 2nd ed edition.

Yümlü, M. S., Gürgen, F. S., Cemgil, A. T., and Okay, N. (2015). Bayesian Changepoint and Time-varying Parameter Learning in Regime-switching Volatility Models. *Digital Signal Processing*, 40:198–212.