Stat Comput
DOI 10.1007/s11222-011-9289-1

Sequential parameter learning and filtering in structured autoregressive state-space models

Raquel Prado · Hedibert F. Lopes

Received: 11 January 2011 / Accepted: 7 September 2011

Abstract We present particle-based algorithms for sequential filtering and parameter learning in state-space autoregressive (AR) models with structured priors. Non-conjugate priors are specified on the AR coefficients at the system level by imposing uniform or truncated normal priors on the moduli and wavelengths of the reciprocal roots of the AR characteristic polynomial. Sequential Monte Carlo algorithms are considered and implemented for on-line filtering and parameter learning within this modeling framework. More specifically, three SMC approaches are considered and compared by applying them to data simulated from different state-space AR models. An analysis of a human electroencephalogram signal is also presented to illustrate the use of the structured state-space AR models in describing biomedical signals.

Keywords State-space autoregressions · Structured priors · Sequential filtering and parameter learning

1 Introduction

We consider models of the form $y_t = x_t + \epsilon_t$, with x_t an AR(p) process with coefficients $\phi_1, \ldots, \phi_p$. These models have been widely used in a number of applied areas including econometrics, and biomedical and environmental signal processing. In many of these applied settings it is important to incorporate prior information on physically meaningful quantities that may represent quasiperiodic or low frequency trends related to latent processes of the observed time series y_t. For example, brain signals such as electroencephalograms (EEGs) can be thought as a superposition of latent processes where each of them describes brain activity in a specific frequency band (e.g., Prado 2010a, 2010b) and so, it is desirable to consider priors on the model parameters that can provide qualitative and quantitative information on the quasiperiodic latent processes that characterize these signals.

Following an approach similar to that proposed in Huerta and West (1999b) for AR models at the observational level, we specify priors on the real and complex reciprocal roots of the AR characteristic polynomial at the state level of the state-space model. Such prior structure is non-conjugate, and so filtering and parameter inference are not available in closed form or via direct simulation. Huerta and West (1999b) proposed a Markov chain Monte Carlo (MCMC) algorithm for inference in structured AR models that can be extended to the case of structured state-space AR models, however, such algorithm would not be useful in settings where $p(x_t, \boldsymbol{\phi}, v, w|y_{1:t})$—with v and w the observational and system variances, respectively—needs to be updated sequentially over time.

We consider particle-based algorithms for on-line parameter learning and filtering in state-space AR models with truncated normal or uniform priors on the moduli and the wavelengths of the autoregressive reciprocal roots. The paper focuses on a sequential Monte Carlo algorithm based on the particle learning approach of Carvalho et al. (2010); however, algorithms based on the approaches of Liu and

R. Prado (✉)
Department of Applied Mathematics and Statistics, Baskin School of Engineering, University of California Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, USA
e-mail: raquel@ams.ucsc.edu

H.F. Lopes
University of Chicago Booth School of Business, 5807 South Woodlawn Avenue, Chicago, IL 60637, USA
e-mail: hlopes@chicagobooth.edu

Published online: 12 October 2011

West (2001) and Storvik (2002) are also considered. We compare the performance of these SMC algorithms in various simulation settings. Comparisons to MCMC algorithms are also presented. The particle-based algorithms can also be used to obtain inference on standard AR models with structured priors, providing an alternative on-line approach to the MCMC scheme of Huerta and West (1999b).

2 Structured AR models

Consider an AR(p) plus noise model, or AR state-space model, given by

$$y_t = x_t + \epsilon_t, \quad \epsilon_t \sim N(0, v), \tag{1}$$

$$x_t = \sum_{i=1}^{p} \phi_i x_{t-i} + w_t, \quad w_t \sim N(0, w), \tag{2}$$

where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)'$ is the p-dimensional vector of AR coefficients, v is the observational variance, and w is the variance at the state level. It is assumed that $\boldsymbol{\phi}$, v and w are unknown with a prior structure that will be described below. Note that the model specified in (1) and (2) can also be written in the following dynamic linear model (DLM) or state-space form

$$y_t = \mathbf{F}'\mathbf{z}_t + \epsilon_t, \tag{3}$$

$$\mathbf{z}_t = \mathbf{G}(\boldsymbol{\phi})\mathbf{z}_{t-1} + \boldsymbol{\eta}_t, \tag{4}$$

with $\mathbf{F}' = (1, 0, \ldots, 0)$; $\mathbf{z}_t = x_{t:(t-p+1)} = (x_t, x_{t-1}, \ldots, x_{t-p+1})'$ the p-dimensional vector of state parameters;

$$\mathbf{G}(\boldsymbol{\phi}) = \begin{pmatrix} \boldsymbol{\phi}_{1:(p-1)} & \phi_p \\ \mathbf{I}_{p-1} & \mathbf{0}_{p-1} \end{pmatrix} \tag{5}$$

the state evolution matrix, with $\mathbf{I}_{p-1}$ the $(p-1) \times (p-1)$ identity matrix and $\mathbf{0}_{p-1}$ the $(p-1)$ vector of zeros; $\epsilon_t \sim N(0, v)$; and $\boldsymbol{\eta}_t \sim N(\mathbf{0}, \mathbf{W})$ with $\mathbf{W} = w\mathbf{F}\mathbf{F}'$.

2.1 Prior structure

We assume a prior structure such that $p(\boldsymbol{\phi}, v, w) = p(\boldsymbol{\phi})p(v)p(w)$. More specifically, it is assumed that the parameters v and w have standard inverse-gamma prior distributions $p(v) = IG(v|a_v, b_v)$ and $p(w) = IG(w|a_w, b_w)$ for some known values of a_v, b_v, a_w, and b_w. In addition, considering an approach similar to that proposed in Huerta and West (1999b), we specify the prior structure on $\boldsymbol{\phi}$ via the autoregressive reciprocal characteristic roots as follows. For a given $\boldsymbol{\phi}$, the AR characteristic polynomial is defined by

$$\Phi(u) = 1 - \phi_1 u - \cdots - \phi_p u^p, \tag{6}$$

where u is a complex number. The AR process is stationary if all the roots of $\Phi(u)$ (real or complex) lie outside the unit circle, or equivalently, if the moduli of all the reciprocal roots of $\Phi(u)$ are below one. Note that the eigenvalues of $\mathbf{G}(\boldsymbol{\phi})$ are the reciprocal roots of the AR characteristic polynomial $\Phi(u)$. Assume that the number of real roots and the number of complex roots of (6) are known a priori and set to R and $2C$, respectively. Then, denoting (r_j, λ_j) the modulus and wavelength of the jth complex reciprocal pair, for $j = 1 : C$, r_j the modulus of the jth real reciprocal root, for $j = (C+1) : (C+R)$, and $\boldsymbol{\alpha} = \{\alpha_j; j = 1 : (C+R)\}$ such that $\alpha_j = (r_j, \lambda_j)$ for $j = 1 : C$ and $\alpha_j = r_j$ for $j = (C+1) : (C+R)$, we have the following mapping from $\boldsymbol{\phi}$ to $\boldsymbol{\alpha}$

$$\Phi(u) = \prod_{j=1}^{C} (1 - r_j e^{-2\pi i/\lambda_j} u)(1 - r_j e^{2\pi i/\lambda_j} u) \times \prod_{j=(C+1)}^{(C+R)} (1 - r_j u). \tag{7}$$

The prior on $\boldsymbol{\phi}$ is that implied by the following prior distribution on $\boldsymbol{\alpha}$:

- *Prior on the complex roots.* For each $j = 1 : C$ we assume that $r_j \sim g_j^c(r_j)$, and $\lambda_j \sim h_j(\lambda_j)$, where $g_j^c(\cdot)$ is a continuous density over an interval (l_j^r, u_j^r), such that $0 < l_j^r < u_j^r < 1$—and so, stationarity is assumed—and $h_j(\cdot)$ is a continuous density over an interval $(l_j^\lambda, u_j^\lambda)$ such that $2 < l_j^\lambda < u_j^\lambda < \lambda^*$, for a fixed and known value λ^*.
- *Prior on the real roots.* Similarly, for each $j = (C+1) : (C+R)$ we assume that $r_j \sim g_j^r(r_j)$, where $g_j^r(\cdot)$ is a continuous density over an interval (l_j^r, u_j^r) with $-1 < l_j^r < u_j^r < 1$. Again, this prior is consistent with the stationarity assumption since $|r_j| < 1$.

We now discuss some choices of $g_j^c(\cdot)$, $h_j(\cdot)$, and $g_j^r(\cdot)$. Huerta and West (1999b) consider the natural default (reference) prior on r_j, for $j = (C+1) : (C+R)$, and so $g_j^r(r_j) = U(r_j|-1, 1)$, which is the formal reference prior on r_j truncated to the stationary region $(-1, 1)$. Similarly, choices of $g_j^c(\cdot)$ and $h_j(\cdot)$ include uniform priors and margins for λ_j based on uniform priors for the corresponding frequency $2\pi/\lambda_j$. Again, Huerta and West (1999b) propose using the natural default prior or "component reference prior" induced by assuming a uniform prior on the implied AR(2) coefficients, $2r_j \cos(2\pi/\lambda_j)$ and $-r_j^2$, but restricted to the finite support of λ_j for propriety. More specifically, such prior is defined as $g_j^c(r_j) \propto r_j^2$, so that the marginal for r_j is $Beta(\cdot|3, 1)$, and $h_j(\lambda_j) \propto \sin(2\pi/\lambda_j)/\lambda_j^2$. In applied settings it is important to consider informative prior distributions on all or some of the AR characteristic roots. For instance, when modeling brain signals such as electroencephalograms, it may be desirable to characterize

different types of brain activity by constraining the frequencies of some of the complex reciprocal roots to certain frequency bands (e.g., Prado 2010a, 2010b). In such cases it is useful to consider uniform priors of the form $g_j^c(\lambda_j) = U(\lambda_j|l_j^\lambda, u_j^\lambda)$, or truncated normal priors $g_j^c(\lambda_j) = TN(\lambda_j|m_j^\lambda, C_j^\lambda, \mathcal{R}_j^\lambda)$, with $\mathcal{R}_j^\lambda = (l_j^\lambda, u_j^\lambda)$ and with l_j^λ, u_j^λ, m_j and C_j known. If these quasiperiodic components are also expected to be fairly persistent, their modulus can also be constrained by taking $g_j^c(r_j) = U(r_j|l_j^r, u_j^r)$, or $g_j^c(r_j) = TN(r_j|m_j^r, C_j^r, \mathcal{R}_j^r)$ with $\mathcal{R}_j^r = (l_j^r, u_j^r)$ and $l_j^r > 0.5$. Uniform and truncated normal priors on (r_j, λ_j) lead to priors on the implied AR coefficients that can be approximated by bivariate truncated normal distributions (Prado 2010b). Similarly, uniform or truncated normal priors can be used on the reciprocal real roots. Such priors will be considered here.

Once the priors are specified and observations begin to arrive, we are interested in obtaining $p(x_t, \boldsymbol{\alpha}, v, w|y_{1:t})$ sequentially over time. However, the class of priors just described leads to posterior distributions that are not available in closed form. Huerta and West (1999b) proposed a reversible jump MCMC algorithm to obtain posterior samples of $\boldsymbol{\alpha}$ and v in standard AR models with a prior structure similar to that described above, but with additional point masses at 1, -1 and 0 placed on the moduli of the real and complex reciprocal roots to account for nonstationarity and model order uncertainty. The algorithm of Huerta and West (1999b) works very well and has been applied to a broad range of time series data sets, however, it is not useful in settings where inference needs to be performed online. In addition, the methods of Huerta and West (1999b) were developed for standard AR models at the observational level, this is, models of the form $y_t = \sum_{j=1}^p \phi_j y_{t-j} + \epsilon_t$, instead of the AR state-space models given in (1) and (2). The latter were considered in West (1997) in the context of studying several oxygen isotope time series from cores recorded at various geographical locations. Specifically, West (1997) describes a MCMC algorithm to obtain samples from $p(x_{1:T}, \boldsymbol{\phi}, v, w|y_{1:T})$ assuming conjugate normal priors on the AR coefficients. Therefore, the approach of West (1997) does not allow us to sequentially update $p(x_t, \boldsymbol{\alpha}, v, w|y_{1:t})$ in cases where structured priors are considered on the AR parameters of the latent process x_t.

In Sect. 3 we propose algorithms for on-line parameter learning and filtering within the class of models specified by (1), (2), and the prior structure on the AR characteristic reciprocal roots described above. We consider three algorithms: an algorithm based on the particle learning (PL) approach of Carvalho et al. (2010); an algorithm based the approach of Liu and West (2001); and finally, a scheme based on the sequential importance sampling resampling (SISR) approach of Storvik (2002). In Sect. 4 we compare the performance of these SMC algorithms by applying them to data simulated from various AR plus noise models. We also compare the performance of the PL-based scheme to a Markov chain Monte Carlo algorithm. Finally, we include the analysis of a portion of a human electroencephalogram to illustrate the use of the structured AR plus noise models and the algorithms for on-line filtering and parameter estimation.

3 Sequential filtering and parameter learning

Here we derive sequential filtering and parameter learning for our class of structured autoregressive state-space models. We focus on the particle learning approach of Carvalho et al. (2010) and Lopes et al. (2010), but we also consider two additional algorithms: one based on the scheme of Liu and West (2001) and another one following the SISR approach of Storvik (2002). For recent overviews of particle filters and SMC methods the readers are referred to Cappé et al. (2007), Doucet and Johansen (2011) and Lopes and Tsay (2011).

3.1 Particle learning algorithm

Carvalho et al. (2010) develop parameter learning, filtering and smoothing algorithms in rather general state-space models. In particular, methods are provided to deal with conditionally Gaussian dynamic linear models (CDLMs), as well as models that are conditionally Gaussian but nonlinear at the state level. These algorithms have two main features. First, similar to other approaches such as those presented in Fearnhead (2002) and Storvik (2002), conditional sufficient statistics are used to represent the posterior distribution of the parameters. Furthermore, sufficient statistics for the latent states are also used whenever the model structure allows it. This reduces the variance of the sampling weights, resulting in algorithms with increased efficiency. Second, in contrast with other particle based schemes that first propagate and then resample the particles, the particle learning (PL) scheme of Carvalho et al. (2010) follows a "resample-and-then-propagate" framework that helps delaying the decay in the particle approximation often found in algorithms based on sampling importance resampling (SIR).

It is worth noting that particle degeneracy (or particle impoverishment) is inherent to all particle filter algorithms as $t \to \infty$ for fixed number of particles M (Del Moral et al. 2006), including particle learning. The main advantage of PL and Storvik filters, as will be seen in what follows, is that their rate of decaying in particle diversity is slower when compared to Liu and West (2001) algorithm. In our simulated and real applications the sample sizes are moderate with values ranging from $t = 100$ and $t = 400$. We found that Liu and West filter had smaller effective sample sizes than those obtained with PL and Storvik filters in all the examples. Smaller effective sample sizes are associated with particle degeneracy, and so the filters of PL and Storvik had better performance than the filter of Liu and West. For further discussion on particle degeneracy when learning about

states and parameters see, amongst others, Chopin et al. (2011), Del Moral et al. (2006), Flury and Shephard (2009) and Poyiadjis et al. (2011), as well as, Lopes et al. (2010) and its discussion.

Here we propose a scheme based on PL for CDLMs of Carvalho et al. (2010) that will allow us to achieve online posterior learning and filtering in structured AR(p) plus noise models. Consider a state-space model whose structure is that given in (3) and (4), with $\epsilon_t \sim N(0, V_t(\boldsymbol{\theta}))$ and $\boldsymbol{\eta}_t \sim N(\mathbf{0}, \mathbf{W}_t(\boldsymbol{\theta}))$, for some fixed parameters $\boldsymbol{\theta}$. In other words, conditional on $\boldsymbol{\theta}$, it is assumed that the model is fully specified at time t by the quadruple $\{\mathbf{F}, \mathbf{G}(\boldsymbol{\theta}), V_t(\boldsymbol{\theta}), \mathbf{W}_t(\boldsymbol{\theta})\}$. In the case of the AR plus noise models, the fixed parameters are the moduli and frequencies of the AR reciprocal roots, denoted by $\boldsymbol{\alpha}$, and the variances w and v. Therefore, $\boldsymbol{\theta} = (\boldsymbol{\alpha}, w, v)$, $\mathbf{G}(\boldsymbol{\theta}) = \mathbf{G}(\boldsymbol{\phi}) = \mathbf{G}(q(\boldsymbol{\alpha}))$—where the mapping from $\boldsymbol{\phi}$ to $\boldsymbol{\alpha}$ is given in (7)—$V_t(\boldsymbol{\theta}) = v$, and $\mathbf{W}_t(\boldsymbol{\theta}) = w\mathbf{F}\mathbf{F}'$. We begin by assuming that at time t we have a set of equally weighted particles $\{(\mathbf{z}_t, \boldsymbol{\theta}, \mathbf{s}_t^z, \mathbf{s}_t^\theta)^{(m)}; m = 1 : M\}$, where $\mathbf{s}_t^z$ and $\mathbf{s}_t^\theta$ are the sufficient statistics for the states and the parameters, respectively. These particles can be used to approximate $p(x_t, \boldsymbol{\alpha}, w, v|y_{1:t})$. Then, at time $t+1$ and for each m the following steps are performed.

Particle learning for AR state-space models

Step 1. Sample an index k^m from $\{1, \ldots, M\}$ with probability

$$\Pr(k^m = k) \propto p(y_{t+1}|(\mathbf{s}_t^z, \boldsymbol{\theta})^{(k)}).$$

Step 2. Propagate the states via

$$\mathbf{z}_{t+1}^{(m)} \sim p(\mathbf{z}_{t+1}|(\mathbf{z}_t, \boldsymbol{\theta})^{(k^m)}, y_{t+1}).$$

Step 3. Propagate the sufficient statistics for states via

$$\mathbf{s}_{t+1}^{z,m} = \mathcal{K}(\mathbf{s}_t^{z,(k^m)}, \boldsymbol{\theta}^{(k^m)}, y_{t+1})$$

Step 4. Propagate $\mathbf{s}_{t+1}^{\theta,m}$ and sample $\boldsymbol{\theta}^{(m)}$ using a conditional structure on the reciprocal roots.

In Step 3, $\mathcal{K}(\cdot)$ denotes the Kalman filter recursion given as follows. Let $\mathbf{s}_t^z = (\mathbf{m}_t, \mathbf{C}_t)$ be the Kalman filter first and second moments at time t conditional on $\boldsymbol{\theta}$. Then, in a model defined by $\{\mathbf{F}, \mathbf{G}(\boldsymbol{\theta}), V_t(\boldsymbol{\theta}), \mathbf{W}_t(\boldsymbol{\theta})\}$, $\mathbf{s}_{t+1}^z = (\mathbf{m}_{t+1}, \mathbf{C}_{t+1})$ are obtained via

$$\mathbf{m}_{t+1} = \mathbf{a}_{t+1} + \mathbf{A}_{t+1}(y_{t+1} - \mathbf{F}'\mathbf{a}_{t+1}) \tag{8}$$

$$\mathbf{C}_{t+1}^{-1} = \mathbf{R}_{t+1}^{-1} + \mathbf{F}V_{t+1}^{-1}(\boldsymbol{\theta})\mathbf{F}', \tag{9}$$

where $\mathbf{a}_{t+1} = \mathbf{G}(\boldsymbol{\theta})\mathbf{m}_t$, $\mathbf{R}_{t+1} = \mathbf{G}(\boldsymbol{\theta})\mathbf{C}_t\mathbf{G}'(\boldsymbol{\theta}) + \mathbf{W}_t(\boldsymbol{\theta})$, $\mathbf{A}_{t+1} = \mathbf{R}_{t+1}\mathbf{F}Q_{t+1}^{-1}$ and $Q_{t+1} = \mathbf{F}'\mathbf{R}_{t+1}\mathbf{F} + V_{t+1}(\boldsymbol{\theta})$. Finally, parameter learning is achieved by Step 4 and the new set of particles given by $\{(\mathbf{z}_{t+1}, \boldsymbol{\theta}, \mathbf{s}_{t+1}^z, \mathbf{s}_{t+1}^\theta)^{(m)}; m = 1 : M\}$ can be used to approximate $p(x_{t+1}, \boldsymbol{\alpha}, w, v|y_{1:(t+1)})$.

We now discuss in detail Steps 1 to 4 in the specific case of AR(p) plus noise models with the structured priors specified in Sect. 2. First, given the model structure we have that $\Pr(k^m = k)$ in Step 1 of the algorithm is proportional to

$$p(y_{t+1}|(\mathbf{s}_t^z, \boldsymbol{\theta})) = N(y_{t+1}|\boldsymbol{\phi}'\mathbf{m}_t, \boldsymbol{\phi}'\mathbf{C}_t\boldsymbol{\phi} + v + w). \tag{10}$$

In the general PL algorithm the weights are computed via $p(y_{t+1}|(\mathbf{z}_t, \boldsymbol{\theta}))$, however, as pointed out by Carvalho et al. (2010), using (10) is more efficient.

Then, the propagation of $\mathbf{z}_{t+1}$ in Step 2 is done as follows. First note that $\mathbf{z}_{t+1} = (x_{t+1}, x_t, \ldots, x_{t-p+2})'$ and $\mathbf{z}_t = (x_t, x_{t-1}, \ldots, x_{t-p+2}, x_{t-p+1})'$. Therefore, the last $p-1$ entries of $\mathbf{z}_{t+1}$ are the first $p-1$ entries of $\mathbf{z}_t$, so we just replace these components accordingly and sample the first element of $\mathbf{z}_{t+1}$, x_{t+1}, from

$$p(x_{t+1}|\mathbf{z}_t, \boldsymbol{\theta}, y_{t+1}) = N(\mathbf{m}_{t+1}^*(1), \mathbf{C}_{t+1}^*(1,1)),$$

where $\mathbf{m}_{t+1}^*(1)$ is the first component of the vector $\mathbf{m}_{t+1}^*$ and $\mathbf{C}_{t+1}^*(1,1)$ is the first component of the matrix $\mathbf{C}_{t+1}^*$, with

$$\mathbf{m}_{t+1}^* = \mathbf{G}(\boldsymbol{\theta})\mathbf{z}_t + \mathbf{A}_{t+1}^* e_{t+1}^*,$$

$$\mathbf{C}_{t+1}^* = \mathbf{W}_t(\boldsymbol{\theta}) - \mathbf{A}_{t+1}^* Q_{t+1}^* (\mathbf{A}_{t+1}^*)',$$

$e_{t+1}^* = (y_{t+1} - \mathbf{F}'\mathbf{G}(\boldsymbol{\theta})\mathbf{z}_t)$, $\mathbf{A}_{t+1}^* = \mathbf{W}_t(\boldsymbol{\theta})\mathbf{F}(Q_{t+1}^*)^{-1}$, and $Q_{t+1}^* = (\mathbf{F}'\mathbf{W}_t(\boldsymbol{\theta})\mathbf{F} + v)$.

Finally, propagating the sufficient statistics for the states in Step 3 is straightforward from the Kalman filter recursions, while propagating the sufficient statistics for the parameters in Step 4 is more involved, as it requires considering the conditional structure of AR(p) processes. This is explained below.

3.1.1 Parameter learning

We now derive the recursions to update $\mathbf{s}_{t+1}^\theta$ the sufficient statistics for $\boldsymbol{\theta}$. First, note that the reciprocal roots are not identified in the formal sense given that the model remains unchanged under any permutation of the indexes that identify such roots. Therefore, to deal with this issue a particular ordering is imposed on the roots. More specifically, we order the reciprocal roots by decreasing moduli, i.e., $\boldsymbol{\alpha} = \{(r_1, \lambda_1), \ldots, (r_C, \lambda_C), r_{C+1}, \ldots, r_{C+R}\}$, with $r_1 > \cdots > r_C$ and $r_{C+1} > \cdots > r_{C+R}$. In addition, we use $\boldsymbol{\alpha}_{(-j)}$ to denote the set with the ordered moduli and frequencies for all the roots except for that indexed by j (which could be complex or real), and $\boldsymbol{\alpha}_{k:l}$ to denote the set of the ordered moduli and frequencies for all the roots except the first $(k-1)$ and the last $C+R-l$.

Updating the sufficient statistics for the complex reciprocal roots Take the j-th complex reciprocal root with modulus

r_j and characteristic wavelength λ_j. Then, conditional on all the remaining reciprocal roots compute

$$x_{(-j),t} = \prod_{l\le C, l\neq j}(1 - r_l e^{-2\pi i/\lambda_l}B)(1 - r_l e^{2\pi i/\lambda_l}B) \times \prod_{l=C+1}^{C+R}(1 - r_l B)x_t, \quad (11)$$

where B is the backshift operator. Under the model, it follows that $x_{(-j),t}$ is an AR(2) with coefficients $\phi_1^{(j)} = 2r_j\cos(2\pi/\lambda_j)$ and $\phi_2^{(j)} = -r_j^2$, and variance w. Therefore, we can write

$$\mathbf{x}_{(-j),t} = \mathbf{X}_{(-j),t}\boldsymbol{\phi}^{(j)} + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim N(0, w\mathbf{I}_{t-2}) \quad (12)$$

with $\mathbf{x}_{(-j),t} = (x_{(-j),3}, \ldots, x_{(-j),t})'$, $\mathbf{I}_{t-2}$ the $(t-2)\times(t-2)$ identity matrix, $\boldsymbol{\phi}^{(j)} = (\phi_1^{(j)}, \phi_2^{(j)})'$, and $\mathbf{X}_{(-j),t} = (\mathbf{x}_{(-j),2:(t-1)}, \mathbf{x}_{(-j),1:(t-2)})$ the $(t-2)\times 2$ design matrix. Equation (12) can then be combined with the prior distribution on $\phi_1^{(j)}$ and $\phi_2^{(j)}$, or equivalently, the prior on r_j and λ_j. For example, if we assume that $\boldsymbol{\phi}^{(j)}$ has a truncated normal prior, say $\boldsymbol{\phi}^{(j)} \sim TN(\mathbf{m}_{j,0}^{\phi}, \mathbf{C}_{j,0}^{\phi}, \mathcal{R}_j^{\phi})$, we have that, up to time t, $\boldsymbol{\phi}^{(j)} \sim TN(\mathbf{m}_{j,t}^{\phi}, \mathbf{C}_{j,t}^{\phi}, \mathcal{R}_j^{\phi})$, with

$$(\mathbf{C}_{j,t}^{\phi})^{-1} = (\mathbf{C}_{j,0}^{\phi})^{-1} + \frac{1}{w}\mathbf{X}'_{(-j),t}\mathbf{X}_{(-j),t} \quad (13)$$

$$(\mathbf{C}_{j,t}^{\phi})^{-1}\mathbf{m}_{j,t}^{\phi} = (\mathbf{C}_{j,0}^{\phi})^{-1}\mathbf{m}_{j,0}^{\phi} + \frac{1}{w}\mathbf{X}'_{(-j),t}\mathbf{x}_{(-j),t}. \quad (14)$$

Now, $\mathbf{X}'_{(-j),t}\mathbf{X}_{(-j),t}$ and $\mathbf{X}'_{(-j),t}\mathbf{x}_{(-j),t}$ are given by

$$\mathbf{X}'_{(-j),t}\mathbf{X}_{(-j),t} = \begin{pmatrix} \sum_{l=3}^{t} x_{(-j),l-1}^2 & \sum_{l=3}^{t} x_{(-j),l-1}x_{(-j),l-2} \\ \sum_{l=3}^{t} x_{(-j),l-1}x_{(-j),l-2} & \sum_{l=3}^{t} x_{(-j),l-2}^2 \end{pmatrix},$$

and

$$\mathbf{X}'_{(-j),t}\mathbf{x}_{(-j),t} = \left(\sum_{l=3}^{t} x_{(-j),l-1}x_{(-j),l}, \sum_{l=3}^{t} x_{(-j),l-2}x_{(-j),l}\right)'.$$

And so, the vector of sufficient statistics to update $\boldsymbol{\phi}^{(j)}$, or equivalently (r_j, λ_j), at time $t+1$ for $j = 1:C$ is given by

$$\mathbf{s}_{(-j),t+1} = \left(\sum_{l=3}^{t+1} x_{(-j),l-1}^2, \sum_{l=3}^{t+1} x_{(-j),l-1}x_{(-j),l-2}, \sum_{l=3}^{t+1} x_{(-j),l-2}^2, \sum_{l=3}^{t+1} x_{(-j),l-1}x_{(-j),l}, \sum_{l=3}^{t+1} x_{(-j),l-2}x_{(-j),l}\right),$$

which can be written as $\mathbf{s}_{(-j),t+1} = (s_{(-j),t+1,1}, \ldots, s_{(-j),t+1,5})'$ with

$$\begin{aligned} s_{(-j),t+1,1} &= s_{(-j),t,1} + x_{(-j),t}^2, \\ s_{(-j),t+1,2} &= s_{(-j),t,2} + x_{(-j),t}x_{(-j),t-1}, \\ s_{(-j),t+1,3} &= s_{(-j),t,3} + x_{(-j),t-1}^2, \\ s_{(-j),t+1,4} &= s_{(-j),t,4} + x_{(-j),t}x_{(-j),t+1}, \\ s_{(-j),t+1,5} &= s_{(-j),t,5} + x_{(-j),t-1}x_{(-j),t+1}. \end{aligned}$$

Updating the sufficient statistics for the real reciprocal roots This is simpler than updating the sufficient statistics for the complex reciprocal roots. Take the j-th real root (i.e., take any j such that $C+1 \le j \le C+R$) with modulus r_j and compute

$$x_{(-j),t} = \prod_{l=1}^{C}(1 - r_l e^{-2\pi i/\lambda_l}B)(1 - r_l e^{2\pi i/\lambda_l}B) \times \prod_{l\ge C+1, l\neq j}^{C+R}(1 - r_l B)x_t,$$

which under the model assumptions follows an AR(1) process with coefficient r_j, and so

$$\mathbf{x}_{(-j),t} = \mathbf{X}_{(-j),t}r_j + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim N(\mathbf{0}, w\mathbf{I}_{t-1}),$$

with $\mathbf{x}_{(-j),t} = (x_{(-j),2}, \ldots, x_{(-j),t})'$, and $\mathbf{X}_{(-j),t} = (x_{(-j),1}, \ldots, x_{(-j),t-1})'$. If a truncated normal prior is assumed on r_j, this is $r_j \sim TN(m_{j,0}^{\phi}, C_{j,0}^{\phi}, \mathcal{R}_j^{\phi})$, we have that, $r_j \sim TN(m_{j,t}^{\phi}, C_{j,t}^{\phi}, \mathcal{R}_j^{\phi})$, with $C_{j,t}^{\phi}$ and $m_{j,t}^{\phi}$ computed via (13) and (14). In this case $\mathbf{X}'_{(-j),t}\mathbf{X}_{(-j),t} = \sum_{l=2}^{t} x_{(-j),l-1}^2$, and $\mathbf{X}'_{(-j),t}\mathbf{x}_{(-j),t} = \sum_{l=2}^{t} x_{(-j),l-1}x_{(-j),l}$ and so, the vector of sufficient statistics to update r_j at time $t+1$ for $j \ge C+1$ is given by

$$\mathbf{s}_{(-j),t+1} = \left(\sum_{l=2}^{t+1} x_{(-j),l-1}^2, \sum_{l=2}^{t+1} x_{(-j),l-1}x_{(-j),l}\right) = (s_{(-j),t,1} + x_{(-j),t}^2, s_{(-j),t,2} + x_{(-j),t}x_{(-j),t+1})'.$$

Updating the sufficient statistics for the variances Given the inverse-gamma priors on v and w, the sufficient statistics associated with these parameters, denoted as $\mathbf{s}_{t+1}^v$ and $\mathbf{s}_{t+1}^w$, respectively, are given by $\mathbf{s}_{t+1}^v = (\nu_{t+1}^v, d_{t+1}^v)'$ and $\mathbf{s}_{t+1}^w = (\nu_{t+1}^w, d_{t+1}^w)'$ with $\nu_{t+1}^v = \nu_t^v + 1$, $\nu_{t+1}^w = \nu_t^w + 1$, $d_{t+1}^v = d_t^v + (y_{t+1} - x_{t+1})^2$, and

$$d_{t+1}^w = d_t^w + \left(x_{t+1} - \sum_{j=1}^{p}\phi_j x_{t+1-j}\right)^2.$$

Propagation of $\mathbf{s}_{t+1}^{\theta,m}$ *and sampling of* $\boldsymbol{\theta}^{(m)}$ Based on the results presented above, Step 4 of the algorithm is performed as follows:

- Begin with $j = 1$. Compute $\mathbf{s}_{(-1),t+1}^{(m)} = \mathcal{S}_1(\mathbf{s}_{(-1),t}^{(k^m)}, \boldsymbol{\alpha}_{(-1)}^{(k^m)}, v^{(k^m)}, w^{(k^m)}, \mathbf{z}_{t+1}^{(m)}, y_{t+1})$ as described above; sample $\boldsymbol{\phi}^{(1),m} \sim TN(\mathbf{m}_{1,t+1}^{\phi,m}, \mathbf{C}_{1,t+1}^{\phi,m}, \mathcal{R}_1^{\phi})$, where $\mathbf{m}_{1,t+1}^{\phi,m}$ and $\mathbf{C}_{1,t+1}^{\phi,m}$—which are computed using (13) and (14)—are functions of $\mathbf{s}_{(-1),t+1}^{(m)}$; and set $r_1^{(m)} = \sqrt{-\phi_2^{(1),m}}$, and $\lambda_1^{(m)} = 2\pi / \arccos(\phi_1^{(1),m} / 2\sqrt{-\phi_2^{(1),m}})$.
- For the remaining complex reciprocal roots, i.e., for $j = 2, \ldots, C$: compute $\mathbf{s}_{(-j),t+1}^{(m)} = \mathcal{S}_j(\mathbf{s}_{(-j),t}^{(k^m)}, \boldsymbol{\alpha}_{1:(j-1)}^{(m)}, \boldsymbol{\alpha}_{(j+1):(R+C)}^{(k^m)}, v^{(k^m)}, w^{(k^m)}, \mathbf{z}_{t+1}^{(m)}, y_{t+1})$; sample $\boldsymbol{\phi}^{(j),m}$ from a truncated normal distribution $TN(\mathbf{m}_{j,t+1}^{\phi,m}, \mathbf{C}_{j,t+1}^{\phi,m}, \mathcal{R}_j^{\phi})$,—again this is done using (13) and (14) and noting that $\mathbf{m}_{j,t+1}^{\phi,m}$ and $\mathbf{C}_{j,t+1}^{\phi,m}$ are functions of $\mathbf{s}_{(-j),t+1}^{(m)}$; set

  $$r_j^{(m)} = \sqrt{-\phi_2^{(j),m}},$$

  and

  $$\lambda_j^{(m)} = 2\pi / \arccos(\phi_1^{(j),m} / 2\sqrt{-\phi_2^{(j),m}}).$$

- For $j = (C+1), \ldots, (C+R)$, i.e., for each of the R real reciprocal roots: compute $\mathbf{s}_{(-j),t+1}^{(m)} = \mathcal{S}_j(\mathbf{s}_{(-j),t}^{(k^m)}, \boldsymbol{\alpha}_{1:(j-1)}^{(m)}, \boldsymbol{\alpha}_{(j+1):(R+C)}^{(k^m)}, v^{(k^m)}, w^{(k^m)}, \mathbf{z}_{t+1}^{(m)}, y_{t+1})$; sample $r_j^{(m)}$ from a truncated normal distribution $TN(m_{j,t+1}^{\phi,m}, C_{j,t+1}^{\phi,m}, \mathcal{R}_j)$.
- Compute $\mathbf{s}_{t+1}^{v,m} = \mathcal{S}_v(\mathbf{s}_t^{v,(k^m)}, \mathbf{z}_{t+1}^{(m)}, y_{t+1})$ and sample $(v^{(m)} | \mathbf{s}_{t+1}^{v,m}) \sim IG(\frac{v_{t+1}^v}{2}, \frac{d_{t+1}^v}{2})$.
- Compute $\mathbf{s}_{t+1}^{w,m} = \mathcal{S}_w(\mathbf{s}_t^{w,(k^m)}, \boldsymbol{\alpha}^{(m)}, \mathbf{z}_{t+1}^{(m)}, y_{t+1})$ and sample $(w^{(m)} | \mathbf{s}_{t+1}^{w,m}) \sim IG(\frac{v_{t+1}^w}{2}, \frac{d_{t+1}^w}{2})$.

Note that running this algorithm when the state x_t follows an AR(p) process with C pairs of complex characteristic roots and R real roots requires storing $5C + 2R + 2$ sufficient statistics per particle for the parameters, in addition to storing the sufficient statistics for the states.

3.2 Liu and West algorithm

Liu and West (2001) proposed a general algorithm that extends auxiliary variable particle filters for state variables (Pitt and Shephard 1999) to include model parameters. This algorithm uses ideas of kernel smoothing (West 1993) to develop a method that artificially evolves the fixed parameters without information loss.

Assume that at time t, the posterior $p(\mathbf{z}_t, \boldsymbol{\gamma} | y_{1:t})$, where $\boldsymbol{\gamma} = f(\boldsymbol{\alpha}, v, w)$ for some function $f(\cdot)$, is summarized by a weighted set of M particles $\{(\mathbf{z}_t, \boldsymbol{\gamma}_t, \omega_t)^{(m)}; m = 1 : M\}$, with $\sum_{m=1}^M \omega_m = 1$. We discuss some choices of $f(\cdot)$ later in this section. The algorithm of Liu and West (2001) for the AR plus noise model would be as follows.

Algorithm of Liu and West

Step 1. Sample an index k^m from $\{1, \ldots, M\}$ with probability

$$\Pr(k^m = k) \propto \omega_t^k \, p(y_{t+1} | \mathbf{m}_{t+1}^{(k)}, \boldsymbol{\mu}_t^{(k)}),$$

with $\mathbf{m}_{t+1}^{(k)} = E(\mathbf{z}_{t+1} | \mathbf{z}_t^{(k)}, \boldsymbol{\mu}_t^{(k)})$, $\boldsymbol{\mu}_t^{(k)} = a\boldsymbol{\gamma}_t^{(k)} + (1-a)\bar{\boldsymbol{\gamma}}_t$ and $\bar{\boldsymbol{\gamma}}_t = \sum_{m=1}^M \omega_t^{(m)} \boldsymbol{\gamma}_t^{(m)}$.

Step 2. Sample $\boldsymbol{\gamma}_{t+1}^{(m)}$ from $N(\boldsymbol{\mu}_t^{(k^m)}, (1-a)^2 \mathbf{V}_t)$ with

$$\mathbf{V}_t = \sum_{m=1}^M \omega_t^{(m)} (\boldsymbol{\gamma}_t^{(m)} - \bar{\boldsymbol{\gamma}}_t)(\boldsymbol{\gamma}_t^{(m)} - \bar{\boldsymbol{\gamma}}_t)'.$$

Step 3. Sample the states from $p(\mathbf{z}_{t+1}^{(m)} | \mathbf{z}_t^{(k^m)}, \boldsymbol{\gamma}_{t+1}^{(m)})$.

Step 4. Compute the new weights via

$$\omega_{t+1}^{(m)} \propto \frac{p(y_{t+1} | \mathbf{z}_{t+1}^{(m)}, \boldsymbol{\gamma}_{t+1}^{(m)})}{p(y_{t+1} | \mathbf{m}_{t+1}^{(k^m)}, \boldsymbol{\mu}_t^{(k^m)})}.$$

Finally, the set of weighted particles $\{(\mathbf{z}_{t+1}, \boldsymbol{\gamma}_{t+1}, \omega_{t+1})^{(m)}; m = 1 : M\}$ can be used to obtain an approximation to $p(\mathbf{z}_{t+1}, \boldsymbol{\gamma} | y_{1:(t+1)})$. Resampling can be done at each time t or only when the effective sample size given by $1 / \sum_{m=1}^M (\omega_t^{(m)})^2$ is below a certain threshold M_0 (Liu 1996). The value of a above is set to $a = (3\delta - 1)/2\delta$, with $\delta \in (0, 1]$ interpreted as a discount factor (see Liu and West 2001). Values of $\delta > 0.9$ are typically used in practice.

In the case of AR plus noise models with uniform or truncated normal priors on the moduli and wavelengths of the AR reciprocal roots, the algorithm above can be applied to transformations of the parameters so that the normal kernels in Step 2 above are appropriate. For example, assume that uniform priors are considered, i.e., assume that $g_j^c(r_j) = U(r_j | l_j^r, u_j^r)$ and $g_j^r(r_j) = U(r_j | l_j^r, u_j^r)$ for $j = 1 : (C+R)$, and that $h_j(\lambda_j) = U(\lambda_j | l_j^\lambda, u_j^\lambda)$ for $j = 1 : C$. Then, the algorithm can be applied to the transformed parameters $\log((r_j - l_j^r)/(u_j^r - r_j))$ for $j = 1 : (C+R)$, $\log((\lambda_j - l_j^\lambda)/(u_j^\lambda - \lambda_j))$ for $j = 1 : C$, $\log(w)$, and $\log(v)$, so that

$$\boldsymbol{\gamma} = \left(\log \frac{(r_1 - l_1^r)}{(u_1^r - r_1)}, \ldots, \log \frac{(r_{C+R} - l_{C+R}^r)}{(u_{C+R}^r - r_{C+R})}, \right.$$
$$\left. \log \frac{(\lambda_1 - l_1^\lambda)}{(u_1^\lambda - \lambda_1)}, \ldots, \log \frac{(\lambda_C - l_C^\lambda)}{(u_C^\lambda - \lambda_C)}, \log(w), \log(v) \right).$$

Section 4 illustrates the performance of this algorithm in simulated data sets.

3.3 Storvik algorithm

Similar to the approaches of Fearnhead (2002) and Carvalho et al. (2010), the algorithms of Storvik (2002) assume that the posterior distribution of $\boldsymbol{\theta}$ given $\mathbf{z}_{0:t}$ and $y_{1:t}$ depends on a sufficient statistic $\mathbf{t}_t$ that can be updated recursively. When applied to the structured AR plus noise model the general sequential importance sampling resampling algorithm of Storvik (2002) can be summarized as follows.

Storvik Algorithm

Importance sampling: for $m = 1 : M$

Step 1. Sample $\boldsymbol{\theta}^{(m)}$ from $g_1(\boldsymbol{\theta}|\mathbf{z}_{0:t}^{(m)}, y_{1:(t+1)})$,

Step 2. Sample $\tilde{\mathbf{z}}_{t+1}^{(m)} \sim g_2(\mathbf{z}_{t+1}|\mathbf{z}_{0:t}^{(m)}, y_{t+1}, \boldsymbol{\theta}^{(m)})$ and set $\tilde{\mathbf{z}}_{0:(t+1)}^{(m)} = (\mathbf{z}_{0:t}, \tilde{\mathbf{z}}_{t+1}^{(m)})$.

Step 3. Evaluate the weights

$$\tilde{\omega}_{t+1}^{(m)} = \omega_t^{(m)} \frac{p(\boldsymbol{\theta}^{(m)}|\mathbf{t}_t^{(m)})\, p(\tilde{\mathbf{z}}_{t+1}^{(m)}|\mathbf{z}_t^{(m)}, \boldsymbol{\theta}^{(m)})\, p(y_{t+1}|\tilde{\mathbf{z}}_{t+1}^{(m)}, \boldsymbol{\theta}^{(m)})}{g_1(\boldsymbol{\theta}^{(m)}|\mathbf{z}_{0:t}, y_{1:(t+1)})\, g_2(\tilde{\mathbf{z}}_{t+1}^{(m)}|\mathbf{z}_{0:t}^{(m)}, y_{t+1}, \boldsymbol{\theta}^{(m)})}.$$

Resampling: for $m = 1 : M$

Step 1. Sample an index k^m from $\{1, \ldots, M\}$ with probabilities proportional to $\tilde{\omega}_{t+1}^{(k^m)}$.

Step 2. Set $\mathbf{z}_{0:(t+1)}^{(m)} = \tilde{\mathbf{z}}_{0:(t+1)}^{(k^m)}$ and $\omega_{t+1}^{(m)} = 1/M$. Compute the sufficient statistics $\mathbf{t}_{t+1}^{(m)} = \mathcal{T}(\mathbf{t}_t^{(k^m)}, \mathbf{z}_{t+1}^{(m)}, y_{t+1})$.

We set $g_1(\boldsymbol{\theta}|\mathbf{z}_{0:t}^{(m)}, y_{1:(t+1)}) = p(\boldsymbol{\theta}|\mathbf{t}_t^{(m)})$ in Step 1 in order to make computation and simulation fast. We also set $g_2(\mathbf{z}_{t+1}|\mathbf{z}_{0:t}^{(m)}, y_{t+1}, \boldsymbol{\theta}^{(m)}) = p(\mathbf{z}_{t+1}|(\mathbf{z}_t, \boldsymbol{\theta})^{(m)}, y_{t+1})$ and so Step 2 of this algorithm uses the same equations of Step 2 in the PL algorithm. Finally, the sufficient statistics for the parameters are updated as follows. As described in Sect. 3.1, assuming truncated normal or uniform priors on the AR reciprocal roots, it can be shown that

$$(\boldsymbol{\phi}^{(j),m}|y_{1:t}, \mathbf{z}_{0:t}, \boldsymbol{\alpha}_{-j}, w) \sim TN(\mathbf{m}_{j,t}^{\phi,m}, \mathbf{C}_{j,t}^{\phi,m}, \mathcal{R}_j), \quad j = 1 : C,$$

$$(r_j^{(m)}|y_{1:t}, \mathbf{z}_{0:t}, \boldsymbol{\alpha}_{(-j)}, w) \sim TN(m_{j,t}^{(m)}, t, C_{j,t}^{(m)}, \mathcal{R}_j), \quad j = (C+1) : (C+R),$$

$$(w|y_{1:t}, \mathbf{z}_{0:t}, \boldsymbol{\alpha}) \sim IG\left(\frac{\nu_t^w}{2}, \frac{d_t^w}{2}\right),$$

$$(v|y_{1:t}, \mathbf{z}_{0:t}, \boldsymbol{\alpha}) \sim IG\left(\frac{\nu_t^v}{2}, \frac{d_t^v}{2}\right),$$

and so, the sufficient statistic $\mathbf{t}_t$ for $(\boldsymbol{\alpha}, w, v)$ is a function of $\sum_{l=1}^t x_{l-i}^2$ and $\sum_{l=1}^t x_{l-i}x_{l-j}$ for $i, j = 1 : (p-3)$, $\sum_{l=1}^t y_l^2$ and $\sum_{l=1}^t y_l x_l$. However, direct simulation from $p(\boldsymbol{\theta}|\mathbf{t}_t)$, as required in Step 1 of the algorithm, is not possible. Following Storvik (2002) a SISR algorithm that samples $\boldsymbol{\theta}$ approximately from $p(\boldsymbol{\theta}|\mathbf{t}_t)$ using a few Gibbs steps can be applied. We illustrate the performance of this algorithm in Sect. 4.

4 Examples

4.1 Simulation studies

4.1.1 Structured AR(1) plus noise model

A total of $T = 300$ observations were simulated from an AR(1) plus noise model with AR coefficient $\phi = 0.95$—or equivalently, real reciprocal root 0.95—observational variance $v = 0.02$, and system variance $w = 0.1$. We assumed *a priori* that $r \sim U(0, 1)$, $v \sim IG(\alpha_v, \beta_v)$, and $w \sim IG(\alpha_w, \beta_w)$. Non-informative priors on the variances are obtained when $\alpha_v, \alpha_w, \beta_v, \beta_w \to 0$, and so we set $\alpha_v = \alpha_w = 0.01$, and $\beta_v = \beta_w = 0.01$. The three SMC algorithms described in Sect. 3 were applied to the simulated data to obtain particle based approximations of $p(r, w, v, x_t|y_{1:t})$ sequentially over time. Each of the algorithms was independently run 4 times using 4 different random seeds, and all the algorithms used $M = 2000$ particles. A discount factor of $\delta = 0.95$ and was used for the Liu and West algorithm. A resampling step was also performed in the Liu and West algorithm at each time t.

Figure 1 compares parameter learning in the AR(1) plus noise model for the three algorithms. PL and Storvik algorithms show similar performances while the algorithm based on the approach of Liu and West (2001) shows particle degeneracy. Figure 2 displays plots of the effective sample sizes (as percentages based on a total number of $M = 2000$ particles) for each of the algorithms. These plots show that PL and Storvik algorithms have similar effective sample sizes, while the algorithm of Liu and West has much smaller effective sample sizes, with percentages below 80% threshold after $t = 100$ and often below 50% after $t = 150$.

More informative inverse-gamma priors on v and w with $\alpha_v, \alpha_w \in (3, 100)$ and $\beta_v = (\alpha_v - 1) \times 0.02$ and $\beta_w = (\alpha_w - 1) \times 0.1$ were also considered. Such priors led to similar results to those summarized above, i.e., PL and Storvik algorithm showed similar performances and the algorithm of Liu and West showed particle degeneracy issues when $M = 2000$ or less were used.

4.1.2 Structured AR(2) plus noise model

We simulated $T = 400$ observations from the AR(2) plus noise model with autoregressive coefficients $\phi_1 = 0.1$ and $\phi_2 = 0.8075$ and implied real reciprocal roots $r_1 = 0.95$ and $r_2 = -0.85$. The observational and system variances were set at $v = 0.02$ and $w = 0.1$, respectively. The time series plot of the simulated x_t appears in Fig. 3 (black line). We

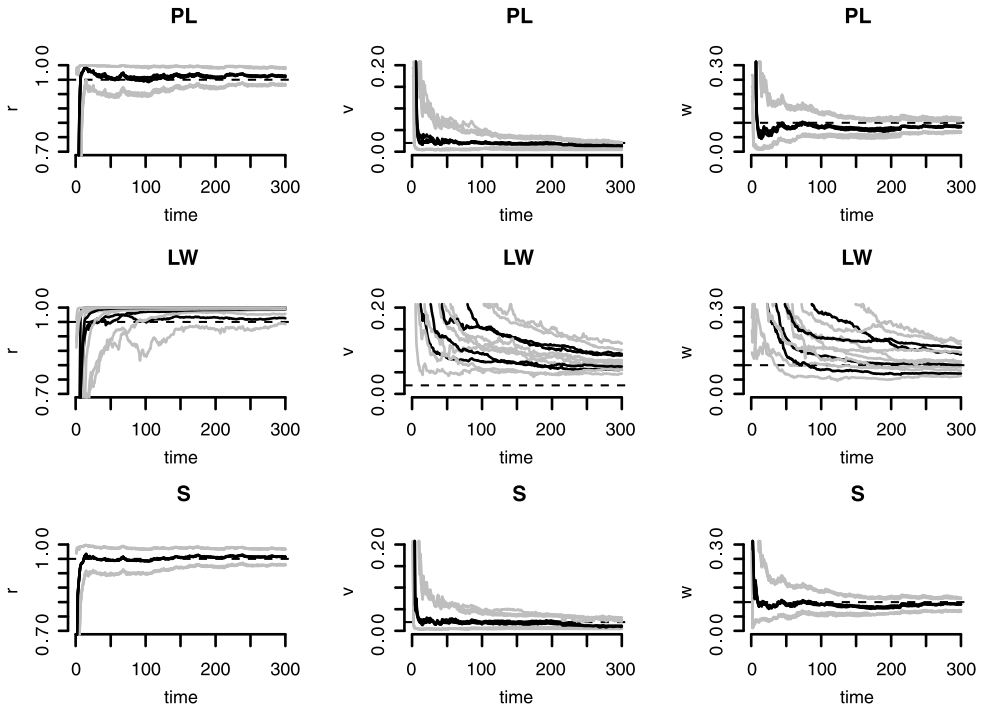

Fig. 1 AR(1) plus noise: Sequential parameter learning. The plots show the posterior means (*black*) and 95% credibility intervals (*gray*) of $(r|y_{1:t})$ (*first column*), $(v|y_{1:t})$ (*second column*), and $(w|y_{1:t})$ (*third column*) for 4 independent runs—i.e., runs with different random seeds—of the PL algorithm (*first row*), the algorithm of Liu and West (*second row*), and Storvik algorithm (*third row*)

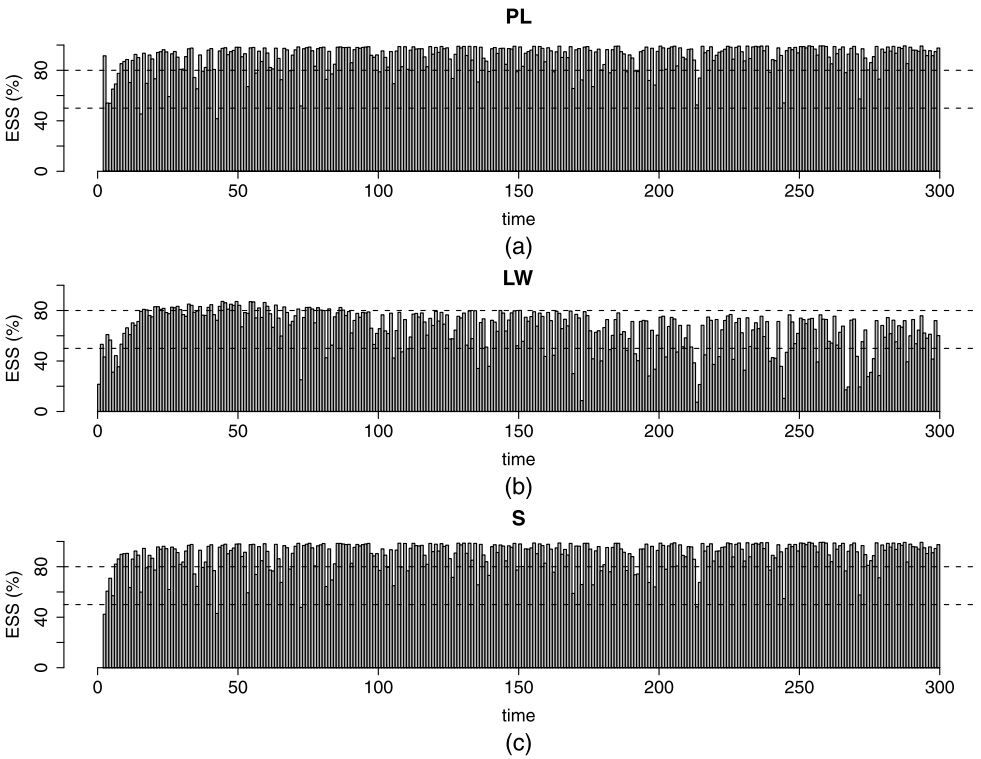

Fig. 2 AR(1) plus noise: Effective sample sizes. The plots show the effective sample sizes (as percentages) for the PL algorithm (*top plot*), the Liu and West algorithm (*middle plot*) and Storvik algorithm (*bottom plot*) based on $M = 2000$ particles. *Dotted lines* appear at 80% and 50%

applied the PL algorithm for on-line filtering and parameter learning described in Sect. 3. Uniform priors were assumed on r_1 and r_2, with $r_1 \sim U(0, 1)$ and $r_2 \sim U(-1, 0)$. Difuse inverse-gamma priors were assumed on v and w, with $v \sim IG(\alpha_v, \beta_v)$ and $w \sim IG(\alpha_w, \beta_w)$, where $\alpha_v = \alpha_w = 0.01$, $\beta_v = \beta_w = 0.01$. Figure 3 also shows the posterior mean of the estimated latent process x_t at each time t and 99% credibility intervals for $(x_t|y_{1:t})$ obtained from running the PL algorithm with $M = 6000$ particles (gray lines). The algorithm of Liu and West (2001) was also applied to these data using $\delta = 0.95$ and $M = 6000$ particles. A resampling step was done at each time t.

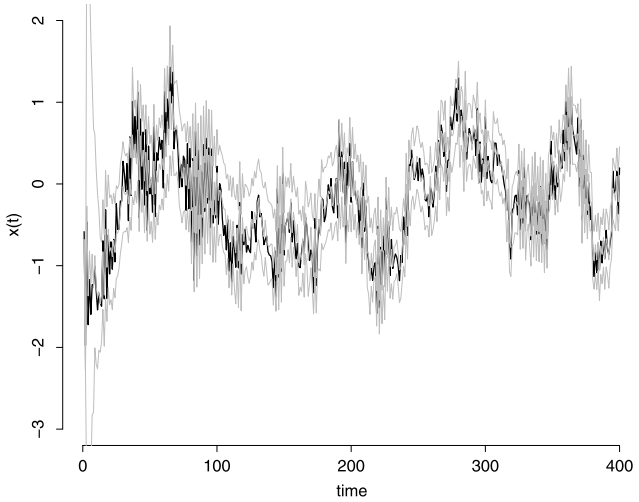

Fig. 3 AR(2) plus noise—Sequential state learning. Sequential posterior mean and 99% credibility interval (*gray lines*) of $(x_t|y_{1:t})$. The *black line* is the true simulated x_t

Figure 4 displays 99% posterior credibility intervals and posterior means of $(r_1|y_{1:t})$, $(r_2|y_{1:t})$, $(w|y_{1:t})$, and $(v|y_{1:t})$, respectively, for $t = 1:400$. The left plots correspond to results obtained by applying the PL algorithm with $M = 6000$ particles and the right plots correspond to posterior summaries obtained from running the algorithm of Liu and West (2001) with $M = 6000$ particles and $\delta = 0.95$. The plots show that the LW algorithm suffers from particle degeneracy much faster than the PL algorithm when the same number of particles is used in both algorithms. Figure 5 shows the effective sample sizes in % for both algorithms, which once again confirms that PL has a better performance than LW in this example.

4.1.3 *Structured AR(3) plus noise model*

In this example, the PL algorithm described in Sect. 3 was applied to $T = 250$ observations simulated from an AR(3) plus noise model with three characteristic reciprocal roots: one real root with modulus $r_1 = -0.95$ and a pair

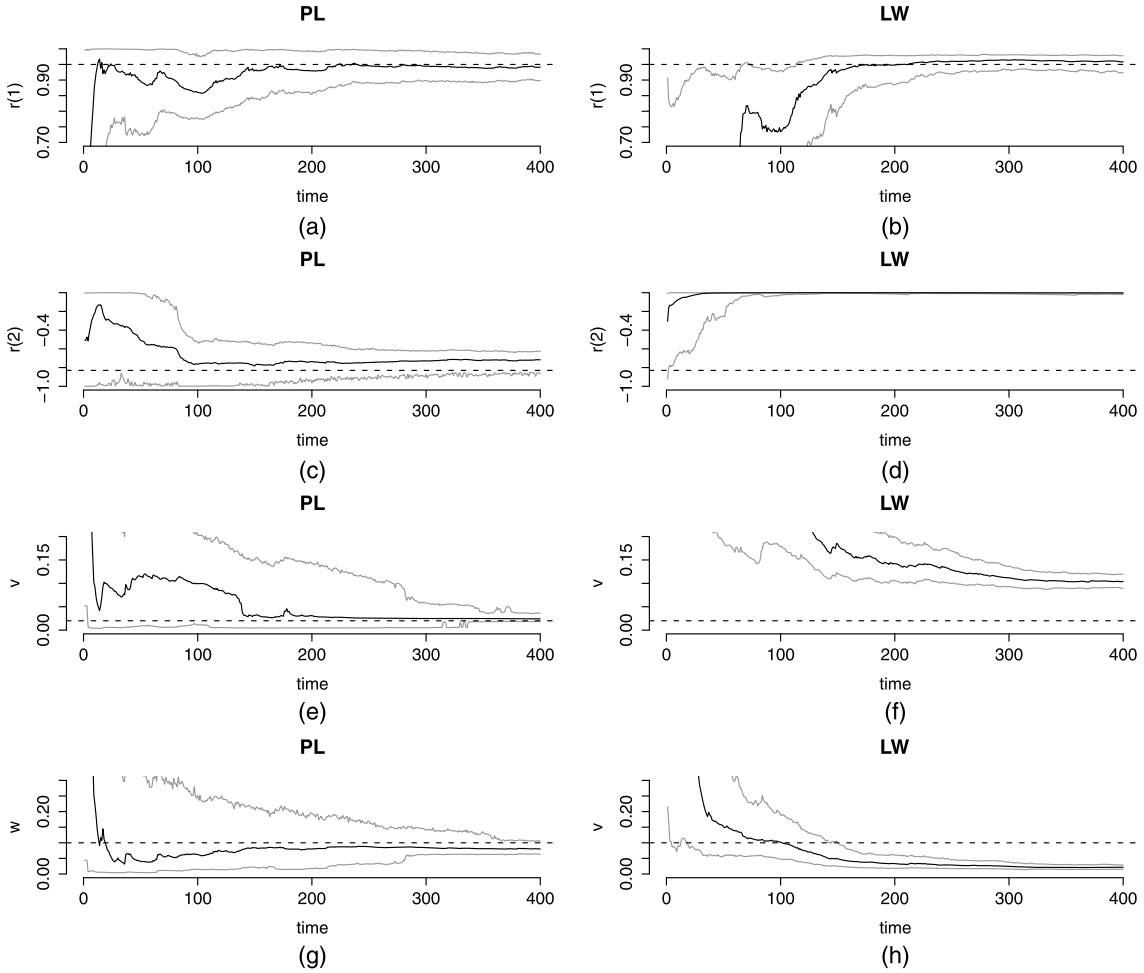

Fig. 4 AR(2) plus noise—Comparing PL (*left*) and LW (*right*). The *left plots* show the sequential posterior means (*black*) and 99% credibility intervals (*gray*). Posterior summaries obtained by running the PL and LW algorithms are based on $M = 6000$ particles

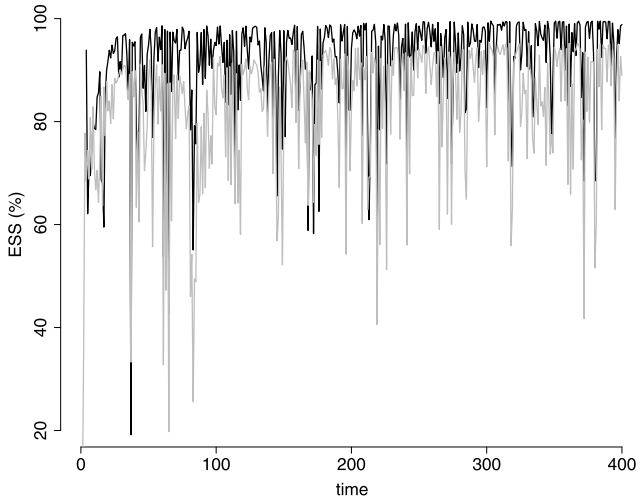

Fig. 5 AR(2) plus noise—Comparing PL and LW. Effective sample sizes (in %) for PL (*black line*) and LW (*gray line*) based on $M = 6000$ particles

of complex roots with modulus $r_2 = 0.95$ and wavelength $\lambda_2 = 16$. The observation and system variances were set to $v = 0.25$ and $w = 1$. The prior distributions were truncated normal priors on the reciprocal roots given by $r_1 \sim TN(r_1|-0.5, 1, (-1, 0))$, and $r_2 \sim TN(r_2|0.8, 1, (0.5, 1))$, for the moduli of the real and complex characteristic roots, and $\lambda_2 \sim TN(\lambda_2|16, 2, (12, 20))$ for the wavelength of the complex characteristic roots. Inverse-gamma priors were assumed on v and w, with $v \sim IG(\alpha_v, \beta_v)$ and $w \sim IG(\alpha_w, \beta_w)$, where $\alpha_v = 2$, $\beta_v = (\alpha_v - 1) \times 0.25$, $\alpha_w = 2$, and $\beta_w = (\alpha_w - 1) \times 1$. These priors correspond to approximate 99% prior credibility intervals of $(0.03, 2.42)$ for v and $(0.13, 9.68)$ for w.

Figure 6 shows the results obtained when the PL algorithm is applied to the simulated data with $M = 2000$ particles. Plots (a), (b), (c), (d) and (e) display, respectively, the distributions of $(r_1|y_{1:t})$, $(r_2|y_{1:t})$, $(\lambda_2|y_{1:t})$, $p(w|y_{1:t})$ and $p(v|y_{1:t})$. The dotted lines in pictures (a)–(e) represent the true values of r_1, r_2, λ_2, w and v. Plot (f) shows the true x_t (solid line) and the posterior mean of $(x_t|y_{1:t})$ (dotted line) for each $t = 1 : T$. From these figures it seems that the PL algorithm successfully learns about the fixed parameters in the model, however, in order to provide a more accurate assessment of its performance we proceed to compare PL-based results to results obtained using a MCMC algorithm.

Figures 7 and 8 illustrate and compare the performance of the PL algorithm with that of an MCMC algorithm that uses the forward filtering backward sampling (FFBS) scheme of Carter and Kohn (1994) and Frühwirth-Schnatter (1994). These figures show summaries of $p(r_1|y_{1:t})$, $p(r_2, \lambda_2|y_{1:t})$ and $p(v, w|y_{1:t})$ for $t = 20$ and $t = 250$ obtained from the PL algorithm with 2000 particles (Fig. 7), and posterior summaries obtained from 10000 MCMC iterations taken after 3000 burn-in iterations (Fig. 8). As shown in the figures, the PL algorithm performs well when compared to MCMC. Particle-based summaries of the various posterior distributions show that the PL scheme allows for the same type of learning that occurs with MCMC when additional data are received (compare summaries at $t = 20$ and $t = 250$ in both cases). Even though more accurate particle approximations (not shown) can be obtained when the number of particles are increased, the PL algorithm successfully achieves parameter learning and filtering in this example with a relatively small number of particles ($M = 2000$).

4.2 Analysis of EEG data

Signals such as EEGs can be thought as a superposition of unobserved latent components that represent brain activity in various frequency bands (Prado 2010a, 2010b). The solid black line in Fig. 9(a) displays a portion of a human EEG. The time series is part of a large data set that contains EEG traces recorded at 19 scalp locations on a patient who received electroconvulsive therapy as a treatment for major depression (for a description of the complete data see Prado et al. 2001 and references therein). The series was modelled in Prado and West (2010) using autoregressive and autoregressive moving average (ARMA) models. Specifically, it was shown that AR models of orders 8–12 could be used describe the main features of the series, which shows a persistent latent quasiperiodic component of period in the 11.5–13.5 range with corresponding modulus in the 0.90–0.99 range, and several latent components with lower period and moduli. These relatively long order AR models act as approximations to more complex, but also more parsimonious, ARMA models. An exploratory search across ARMA(p, q) models with $p, q \in \{1, \ldots, 8\}$ that uses the conditional and approximate log-likelihoods, and chooses the model orders via the Akaike information criterion, indicates that an ARMA$(2, 2)$ is the optimal model within this class to describe the EEG series. Prado and West (2010) also present an analysis in which some of the components of an AR(8) are inverted to approximate ARMA$(2, q)$ models, and find evidence supporting the choice of an ARMA$(2, 2)$.

The AR plus noise models presented here provide a way to describe time series with an ARMA(p, p) structure. As shown in Granger and Morris (1976), the sum of an AR(p) plus white noise results in an ARMA(p, p). The AR plus noise representation is easier to interpret than the ARMA representation in that it separates the sources of error into two: the observational error, that captures measurement error, and the system level error. In addition, the structured priors presented here allow researchers to include scientifically meaningful information, and the SMC algorithms lead to on-line filtering and parameter learning within this class of flexible models.

We now show how the structured AR plus noise models and the SMC algorithms can be used to analyze the EEG

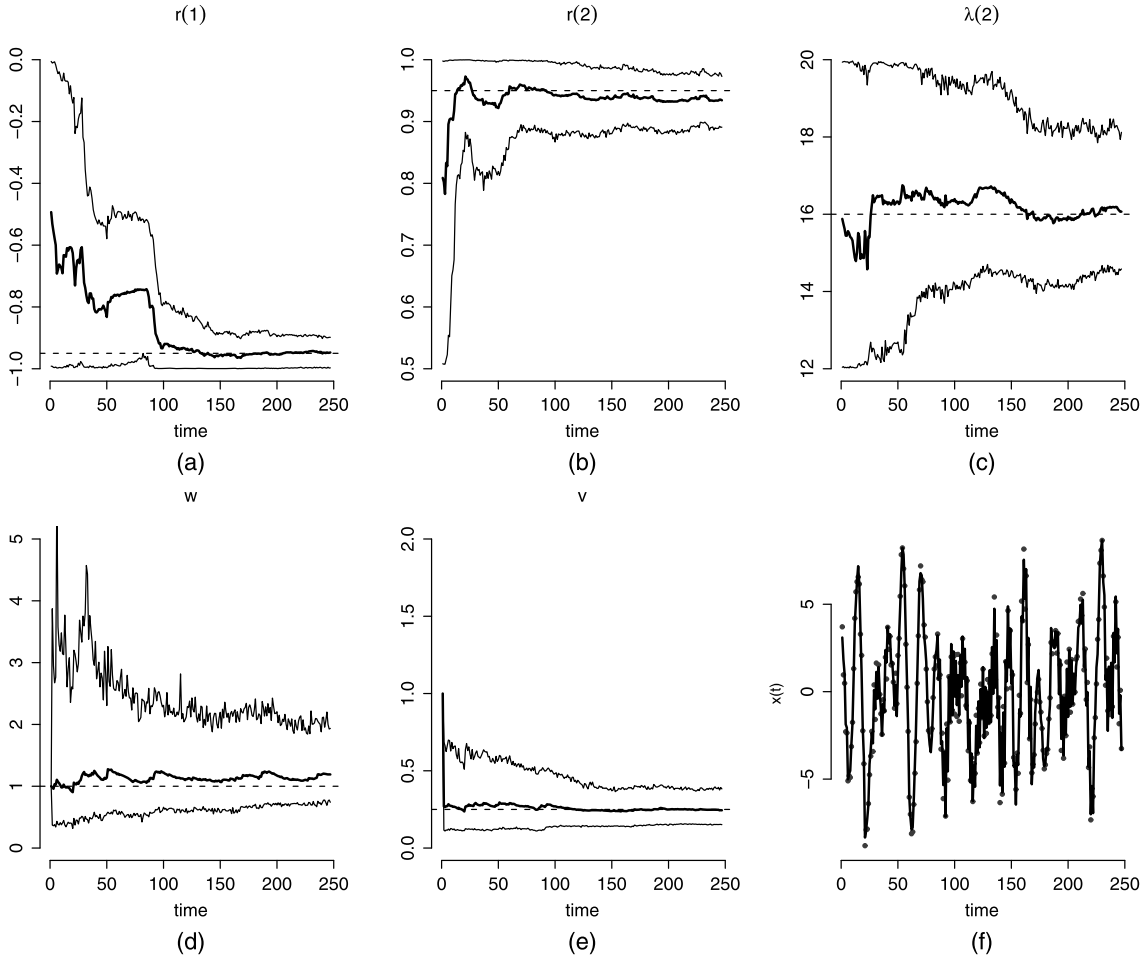

Fig. 6 AR(3) plus noise. Estimates of the AR plus noise model parameters obtained by applying the PL algorithm described in Sect. 3 to the simulated data y_t from an AR(3) plus noise model with one real reciprocal root $r_1 = -0.95$ and a pair of complex reciprocal roots with moduli $r_2 = 0.95$ and wavelength $\lambda_2 = 16$. Plots (**a**), (**b**), (**c**), (**d**), and (**e**) show, respectively, estimates of $p(r_1|y_{1:t})$, $p(r_2|y_{1:t})$, $p(\lambda_2|y_{1:t})$, $p(w|y_{1:t})$ and $p(v|y_{1:t})$ based on $M = 2000$ particles. Plot (**f**) displays x_t (*dots*) and estimates of the posterior mean of $(x_t|y_{1:t})$ (*solid line*) based on 2000 particles at each time $t = 1:250$

series in Fig. 9(a). We fitted an AR(3) plus noise model to the data. We assumed that the AR(3) process x_t had two pairs of complex reciprocal roots with modulus r_1 and period λ_1, and one real root with modulus r_2. Furthermore, we assumed the following prior structure: $r_1 \sim U(0.5, 1)$, $\lambda_1 \sim U(3, 20)$, $r_2 \sim U(-1, 1)$, $v \sim IG(\alpha_v, \beta_v)$ with $\alpha_v = 50$ and $\beta_v = (\alpha_v - 1) \times 3500$, and $w \sim IG(\alpha_w, \beta_w)$ with $\alpha_w = 2$ and $\beta_w = (\alpha_w - 1) \times 1$. The priors for r_1 and λ_1 were chosen to reflect the fact that it is known that the data will show at least one rather persistent quasiperiodic component (i.e., $r_1 > 0.5$), but there is a fair amount of uncertainty about the period of that component. A non-informative prior was chosen on r_2. For the variances, based on previous analyses of similar data, we assume that the standard deviation at the observational level $\sqrt{v}$ lies in the 50–70 microvolts range, while the system standard deviation $\sqrt{w}$ was assumed to be centered at 1.0 with a relatively large dispersion a priori.

Plots (a), (b), and (c) in Fig. 10 show estimates of $p(r_1|y_{1:t})$, $p(\lambda_1|y_{1:t})$, and $p(r_2|y_{1:t})$ for $t = 1:400$ obtained by applying the PL algorithm described in Sect. 3 with $M = 2000$ particles. These plots display the posterior means and 95% posterior intervals. The histograms in (d), (e), and (f), correspond to the PL-based distributions of $(r_1|y_{1:400})$, $(\lambda_1|y_{1:400})$, and $(r_2|y_{1:400})$. The plots show that there is strong evidence of a persistent quasiperiodic component in x_t with a modulus in the 0.9–1.0 range (posterior mean of 0.957), and period in the 11–14 range (posterior mean of 12.72). The 95% posterior interval for r_2 based on $M = 2000$ particles is $(-0.523, -0.121)$, and its posterior mean is -0.372, indicating that this is not a very persistent component. Under this model y_t is assumed to have an ARMA$(3, 3)$ structure, however, the posterior distribution of r_2 suggests that an AR(2) plus noise model, which leads to an ARMA$(2, 2)$ representation of y_t, may be a better model. We fitted such model and obtained similar poste-

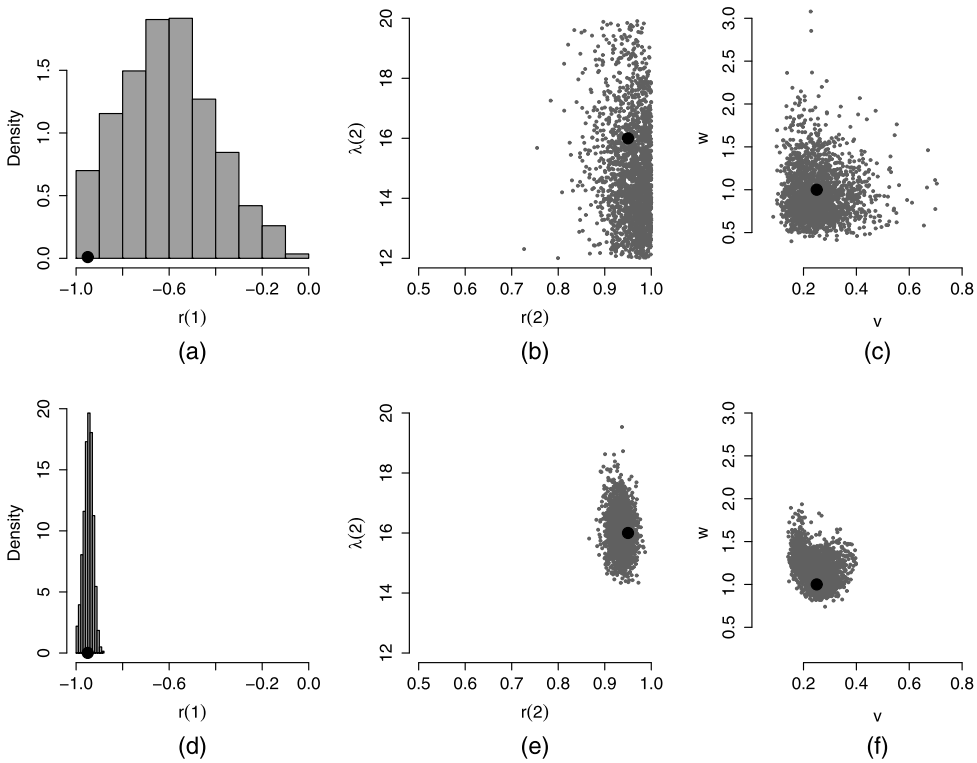

Fig. 7 AR(3) plus noise—PL. Plots (**a**), (**b**) and (**c**) are based on draws from the PL approximations to $p(r_1|y_{1:20})$, $p(r_2, \lambda_2|y_{1:20})$, and $p(v, w|y_{1:20})$, respectively, while the plots (**d**), (**e**) and (**f**) are based on draws from the PL approximations to $p(r_1|y_{1:250})$, $p(r_2, \lambda_2|y_{1:250})$ and $p(v, w|y_{1:250})$. These approximations are based on $M = 2000$ particles

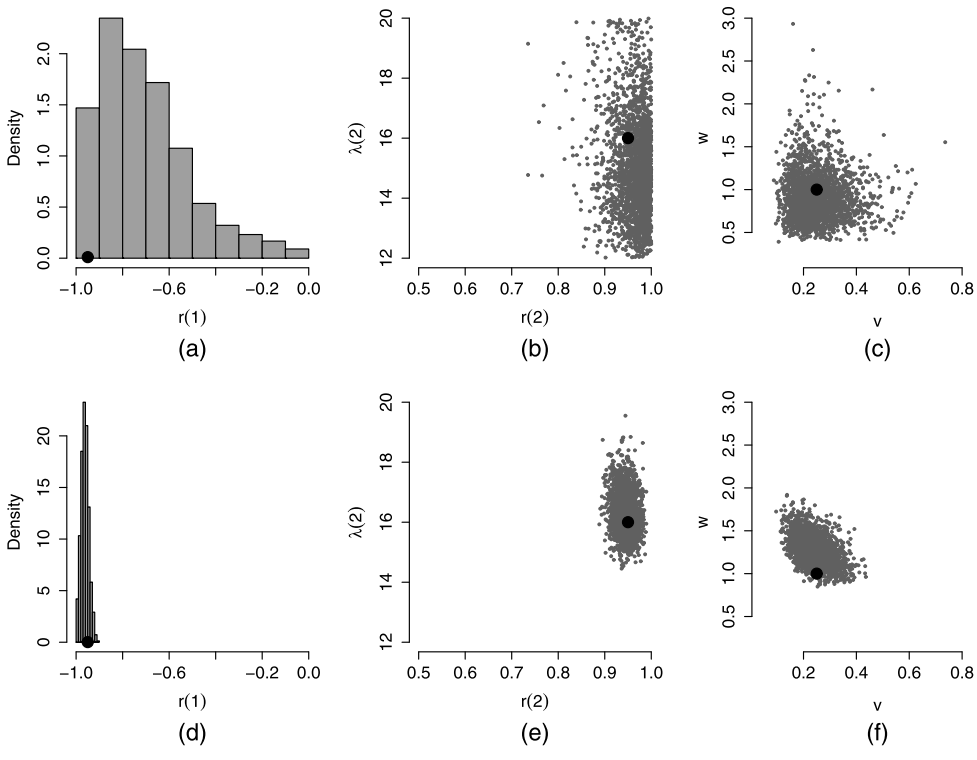

Fig. 8 AR(3) plus noise—MCMC. Plots (**a**), (**b**) and (**c**) are based on draws from the FFBS (MCMC) approximations to $p(r_1|y_{1:20})$, $p(r_2, \lambda_2|y_{1:20})$, and $p(v, w|y_{1:20})$, respectively, while the plots (**d**), (**e**) and (**f**) are based on draws from the FFBS (MCMC) approximations to $p(r_1|y_{1:250})$, $p(r_2, \lambda_2|y_{1:250})$ and $p(v, w|y_{1:250})$. These approximations are based on 10000 MCMC iterations obtained after a burn-in period of 3000 iterations

rior inference on (r_1, λ_1). Formal model comparison can be performed to choose between the AR(3) plus noise and the AR(2) plus noise models. We do not present such comparisons since the focus of the paper is on the SMC algorithms for filtering and parameter learning.

Plots (a)–(c) in Fig. 9 show various aspects of the fit of the AR(3) plus noise model to the EEG data. In particular, plot (a) shows the data, y_t (solid black line), along with 5 simulated traces, $y_t^{(*,m)}$ (gray lines), from the AR(3) plus noise

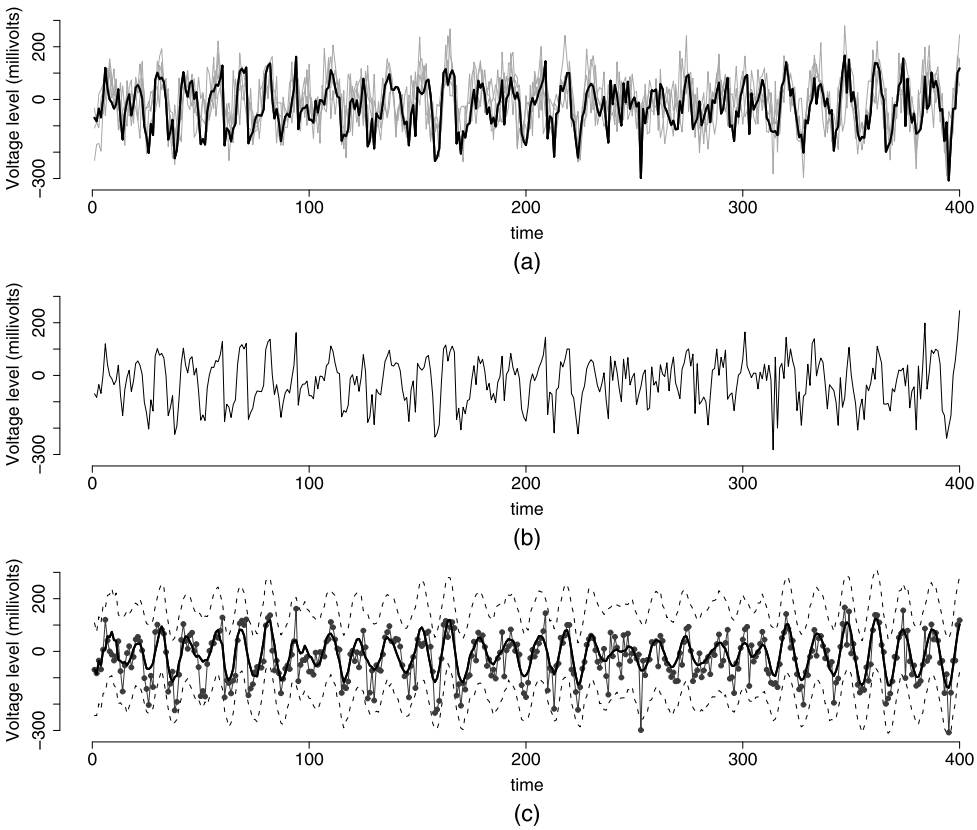

Fig. 9 Analysis of EEG data. (**a**) A section of a human EEG trace (*black line*) and 5 simulated y_t traces (*gray lines*) from the AR(3) plus noise model fitted with the PL algorithm with $M = 2000$ particles. (**b**) The time series shows the observed EEG trace from $t = 1 : 250$ and data simulated from the AR(3) plus noise model fitted with the PL algorithm for $t = 251 : 400$. (**c**) Estimates of the posterior mean of $(x_t|y_{1:t})$ based on the PL approximation with $M = 2000$ particles (*black line*). The graph also shows 95% intervals for $(y_t^*|y_{1:t})$ obtained from the model based on the PL approximation (*dotted lines*); the actual EEG data are displayed in *gray*

model. The traces were obtained via $y_t^{(*,m)} = x_t^{(m)} + \epsilon_t^{(m)}$, with $\epsilon_t^{(m)} \sim N(0, v_t^{(m)})$, and with $x_t^{(m)}$ and $v_t^{(m)}$, randomly chosen values from the $M = 2000$ particle-based representation of $(x_t|y_{1:t})$ and $(v|y_{1:t})$, respectively. Plot (b) displays the actual EEG data for $t = 1 : 250$ and a simulated trace for $t = 251 : 400$. These picture shows that the AR(3) plus noise model does well predictively since it is not easy to determine when the real data ends and when the simulated data begins. Figure 9(c) shows the PL-based posterior mean of $(x_t|y_{1:t})$ (black solid line), 95% posterior intervals for $(y_t^*|y_{1:t})$ (black dotted lines), and the EEG data (dark gray dots and lines). All the observations except two (one around $t = 250$ and another right before $t = 400$) lie within the 95% posterior bands indicating a good model fit. The model can be easily extended to handle outlying observations by changing the structure of the observational noise (e.g., Student-t, mixtures). See Lopes and Polson (2010) for PL applied to stochastic volatility models with Student-t errors.

The results based on the structured AR plus noise models summarized above are consistent with those obtained by Prado and West (2010) via long order AR models. The models presented here are more flexible since they capture additional structure (e.g. ARMA structure) that is at best approximated by long order AR models, and provide a way to incorporate prior information that is scientifically interpretable.

5 Discussion

We consider sequential Monte Carlo approaches to obtain simultaneous filtering and parameter estimation in state-space autoregressive models with structured priors on the AR coefficients. Specifically, algorithms based on the schemes proposed by Liu and West (2001), Storvik (2002), and Carvalho et al. (2010) are detailed and illustrated.

Similar to the approach of Huerta and West (1999b), priors are imposed on the moduli and wavelengths that define the reciprocal characteristic roots of the AR process at the system level. Such priors are important in practical scenarios since they allow for the incorporation of scientifically meaningful information (see examples in Huerta and West 1999a, 1999b, and Prado 2010b). Furthermore, structured AR plus noise models can be used to analyze data with an ARMA structure, as illustrated in the EEG data analysis of Sect. 4.2, providing a more interpretable representation of the ARMA structure and the capability of including informative priors.

Note that if $\epsilon_t = 0$ for all t, the state-space AR model becomes a standard autoregressive model and in such case, the particle-based approaches provide alternative sequential methods for parameter learning in structured autoregressive models that can be used instead of MCMC approaches that are not suitable for real-time inference. In particular, one may choose to use a standard long order autoregressive

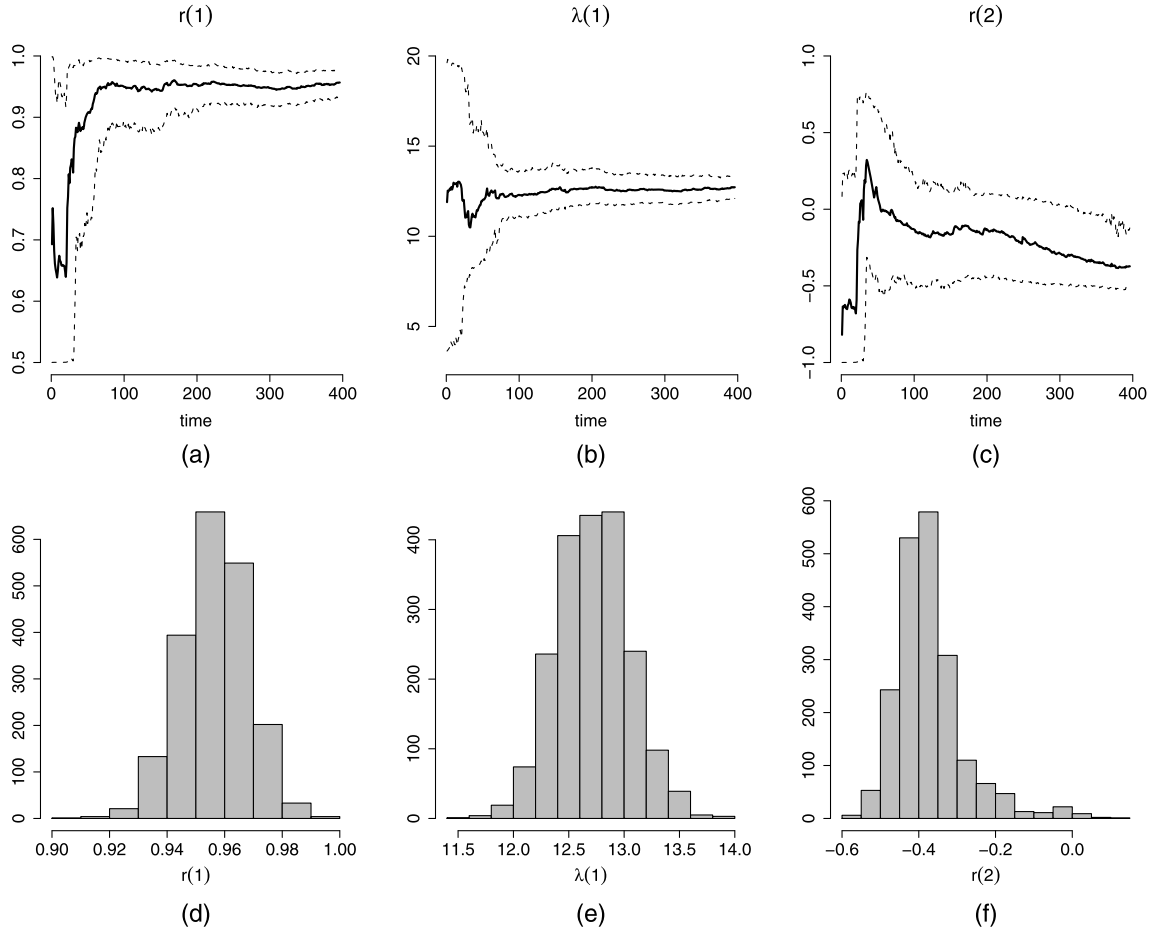

Fig. 10 Analysis of EEG data. Results obtained from applying the PL algorithm to fit a state-space AR(3) model to the EEG data shown in Fig. 9

model with structured priors instead of a low order AR plus noise model with structured priors to approximate a more complex ARMA model. Fitting AR models with no noise at the observational level has some advantages over fitting AR plus noise models when SMC approaches are used, since models with $\epsilon_t = 0$ and constant parameters over time do not show the particle degeneracy issues that appear in state-space models with unknown parameters. On the other hand, prior elicitation and interpretation in high order AR models are much more cumbersome than prior elicitation and interpretation in low order AR plus noise models. In general, we suggest the use of low order AR plus noise models for relatively short time series—say, no more than 1000 observations—and high order standard AR models for long time series.

We applied the proposed algorithms to data simulated from various AR state-space models. In these simulation studies we found that algorithms based on the schemes of Storvik (2002) and Carvalho et al. (2010) had similar performances, leading to good particle-based approximations to the posterior distributions of the parameters and the states. Algorithms based on the approach of Liu and West (2001) are generally easier to implement in the structured AR modeling framework considered here, however, particle degeneracy was evident in many of the examples considered even when relatively large numbers of particles were used. Particle degeneracy is a problem all filters eventually have to deal with, including PL and Storvik filters. However, PL, Storvik and similar filters that replenish particles based on low dimensional vectors of model-specific sufficient statistics will generally delay particle degeneracy (see our discussion on Sect. 3.1).

Acknowledgements The authors would like to thank the Statistical and Applied Mathematical Sciences Institute (SAMSI) for partially supporting his research via the 2008-09 Program on Sequential Monte Carlo Methods.

References

Cappé, O., Godsill, S., Moulines, E.: An overview of existing methods and recent advances in sequential Monte Carlo. IEEE Proc. Signal Process. **95**, 899–924 (2007)

Carter, C., Kohn, R.: Gibbs sampling for state space models. Biometrika **81**, 541–553 (1994)

Carvalho, C., Johannes, M., Lopes, H., Polson, N.: Particle learning and smoothing. Stat. Sci. **25**, 88–106 (2010)

Chopin, N., Jacob, P.E., Papaspiliopoulos, O.: SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates. Technical report, ENSAE-CREST (2011)

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc. B **68**, 411–436 (2006)

Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: Fifteen years later. In: Crisan, D., Rozovsky, B. (eds.) Oxford Handbook of Nonlinear Filtering. Oxford University Press, London (2011)

Fearnhead, P.: MCMC, sufficient statistics and particle filter. J. Comput. Graph. Stat. **11**, 848–862 (2002)

Flury, T., Shephard, N.: Learning and filtering via simulation: smoothly jittered particle filters. Technical report, Oxford-Man Institute (2009)

Frühwirth-Schnatter, S.: Data augmentation and dynamic linear models. J. Time Ser. Anal. **15**, 183–202 (1994)

Granger, C., Morris, M.J.: Time series modelling and interpretation. J. R. Stat. Soc. A **139**(2), 246–257 (1976)

Huerta, G., West, M.: Bayesian inference on periodicities and component spectral structure in time series. J. Time Ser. Anal. **20**, 401–416 (1999a)

Huerta, G., West, M.: Priors and component structures in autoregressive time series models. J. R. Stat. Soc. B **61**, 881–899 (1999b)

Liu, J., West, M.: Combined parameter and state estimation in simulation-based filtering. In: Doucet, A., de Freitas, N., Gordon, N. (eds.) Sequential Monte Carlo Methods in Practice, pp. 197–223. Springer, Berlin (2001)

Liu, J.S.: Metropolized independent sampling with comparisons to rejection sampling and importance sampling. Stat. Comput. **6**, 113–119 (1996)

Lopes, H.F., Polson, N.G.: Particle learning for fat-tailed distributions. Technical report, The University of Chicago Booth School of Business (2010)

Lopes, H.F., Tsay, R.S.: Particle filters and Bayesian inference in financial econometrics. J. Forecast. **30**, 168–209 (2011)

Lopes, H.F., Carvalho, C.M., Johannes, M., Polson, N.G.: Particle learning for sequential Bayesian computation (with discussion). In: Bernardo, J.M., Bayarri, M.J., Berger, J.O., Dawid, A.P., Heckerman, D., Smith, A.F.M., West, M. (eds.) Bayesian Statistics 9, pp. 317–360. Oxford University Press, Oxford (2010)

Pitt, M., Shephard, N.: Filtering via simulation: Auxiliary variable particle filters. J. Am. Stat. Assoc. **94**, 590–599 (1999)

Poyiadjis, G., Doucet, A., Singh, S.S.: Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. Biometrika **98**, 65–80 (2011)

Prado, R.: Characterization of latent structure in brain signals. In: Chow, S., Ferrer, E., Hsieh, F. (eds.) Statistical Methods for Modeling Human Dynamics, pp. 123–153. Routledge, Taylor and Francis, New York (2010a)

Prado, R.: Multi-state models for mental fatigue. In: O'Hagan, A., West, M. (eds.) The Handbook of Applied Bayesian Analysis. Oxford University Press, Oxford (2010b)

Prado, R., West, M.: Time Series: Modeling, Computation, and Inference. CRC Press, Chapman & Hall, Boca Raton (2010)

Prado, R., West, M., Krystal, A.: Multi-channel EEG analyses via dynamic regression models with time-varying lag/lead structure. J. R. Stat. Soc., Ser. C, Appl. Stat. **50**, 95–109 (2001)

Storvik, G.: Particle filters for state-space models with the presence of unknown static parameters. IEEE Trans. Signal Process. **50**, 281–289 (2002)

West, M.: Approximating posterior distributions by mixtures. J. R. Stat. Soc. B **55**, 409–422 (1993)

West, M.: Time series decomposition. Biometrika **84**, 489–494 (1997)