Monte Carlo Smoothing with Application to Audio Signal Enhancement

William Fong, Simon Godsill, Arnaud Doucet and Mike West

Abstract— **We describe methods for applying Monte Carlo filtering and smoothing for estimation of unobserved states in a non-linear state space model. By exploiting the statistical structure of the model, we develop a Rao-Blackwellised Particle Smoother. Due to the lengthy nature of real signals, we suggest processing the data in blocks and a block-based smoother algorithm is developed for this purpose. All the algorithms suggested are tested with real speech and audio data and the results are shown and compared with those generated using the generic particle smoother and the extended Kalman filter. It is found that the proposed Rao-Blackwellised Particle Smoother improves on the standard particle smoother and the extended Kalman smoother. In addition, the proposed Block-based smoother algorithm enhance the efficiency of the proposed Rao-Blackwellised smoother by significantly reducing the storage capacity required for the particle information.**

I. Introduction

MANY problems in applied statistics, statistical signal processing and time series analysis can be stated in a state space form as follows,

$$\begin{array}{lcll} x_{t+1} & \sim & f(x_{t+1}|x_t) & \text{State evolution density} \\ y_{t+1} & \sim & g(y_{t+1}|x_{t+1}) & \text{Observation density} \end{array} \qquad (1)$$

where $\{x_t\}$ are unobserved states of the system and $\{y_t\}$ are observations made over some time, t. $f(.|.)$ and $g(.|.)$ are pre-specified state evolution and observation densities.

A primary concern in many state-space inference problems is sequential estimation of the filtering distribution $p(x_t|y_{1:t})$ and simulation of the entire smoothing distribution $p(x_{1:t}|y_{1:t})$, where $y_{1:t} \triangleq \{y_1, y_2, \ldots, y_t\}$ and $x_{1:t} \triangleq \{x_1, x_2, \ldots, x_t\}$. Updating of the filtering distribution can be achieved in principle using the standard filtering recursions

$$p(x_{t+1}|y_{1:t}) = \int p(x_t|y_{1:t}) f(x_{t+1}|x_t) dx_t$$

$$p(x_{t+1}|y_{1:t+1}) = \frac{g(y_{t+1}|x_{t+1}) p(x_{t+1}|y_{1:t})}{p(y_{t+1}|y_{1:t})}.$$

Similarly, smoothing can be performed recursively backwards in time using the smoothing formula

$$p(x_t|y_{1:T}) = \int p(x_{t+1}|y_{1:T}) \frac{p(x_t|y_{1:t}) f(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} dx_{t+1}.$$

William Fong and Simon Godsill are with the Signal Processing Group, University of Cambridge, Cambridge, CB2 1PZ, U.K. E-mail: wnwf2@eng.cam.ac.uk, sjg@eng.cam.ac.uk

Arnaud Doucet is with Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010 Australia E-mail: a.doucet@ee.mu.oz.au

Mike West is with the Institute of Statistics and Decision Sciences, Duke University, Durham N.C., U.S.A. 27708-0251 E-mail: mw@isds.duke.edu

In practice these filtering and smoothing computations can only be performed in closed form for linear Gaussian models using the Kalman filter / smoother [1], [2] and for finite state-space hidden Markov models.

For non-linear non-Gaussian models, there is no general analytic expression for the required density functions. The extended Kalman filter [1], [2] is a popular approach for non-linear models, which linearises the filtering distributions, so that the Kalman filter can be applied. The extended Kalman filter is computationally cheap but can fail in situations where the density function is highly non-Gaussian or multimodal. In addition to the extended Kalman Filter, Kitagawa *et al.* [3] propose various approximation strategies, which are found to work well for low state dimension models.

Another approximation strategy is that of sequential Monte Carlo methods, also known as Particle Filters [4], [5], [6], [7], [8], [9]. Within the particle filter framework, the filtering distribution is approximated with an empirical distribution formed from point masses, or particles,

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1, \; w_t^{(i)} \geq 0$$

where $\delta(.)$ is the Dirac delta function and $w_t^{(i)}$ is a weight attached to particle $x_t^{(i)}$. Given this particle approximation to the posterior distribution, we can easily estimate the expected value of any function f *w.r.t.* this distribution, $I(f_t) \triangleq \int f(x_t) p(x_t|y_{1:t}) dx_t$, using

$$I(f_t) \approx \sum_{i=1}^{N} f(x_t^{(i)}) w_t^{(i)} \qquad (2)$$

In addition to the particle filter, particle smoothers, which are a simple and efficient method for generating realisations from the entire smoothing density $p(x_{1:T}|y_{1:T})$ using the particulate approximation have been developed [10], [11]. These smoothers generate batched realisations of $p(x_{1:T}|y_{1:T})$ based on the forward particle filtering results.

The main objective of this article is to improve the standard particle filter / smoother with application to audio signal enhancement. In section II, we describe the model adopted for audio signals, which is a non-linear parameterisation of the Time-Varying Autoregressive (TVAR) model. In section III, we briefly review the generic Monte Carlo filtering and smoothing algorithm [10], [11]. In section IV, we develop and describe a variance reduction strategy, the Rao-Blackwellised Particle Smoother, which takes advantage of the "tractable substructure" within our model. Due to the lengthy nature of real signals, we suggest processing the data in blocks and a block-based adaptation of the smoothing algorithm is developed and described in section V. Finally, simulations results are presented in section VI.

II. Audio models

The autoregressive (AR) [12], [13], [14] model is widely used and popular for audio signals. This model exploits the local correlation in a time series by forming the prediction for the current sample as a linear combination of the immediately preceding samples. One shortcoming of using an AR model for audio signals is obvious: the AR coefficients associated with the signal are assumed to be fixed throughout the analysis interval. In reality, however, the AR coefficients are continuously changing. Therefore, any realistic representation should thus involve a model whose parameters evolve over time.

One such model is the time-varying autoregressive (TVAR) process [15], [3], [16]. Models of this class have been applied in the context of speech modelling and enhancement, for example [17], [18], [19], [20]. In [21], a TVAR model with stochastically evolving parameters has been suggested, which is shown to outperform the standard AR process in speech modelling.

The audio signal process $\{u_t\}$, is modelled as a p^{th} order TVAR process, *i.e.*

$$u_t = \sum_{i=1}^{p} a_{t,i} u_{t-i} + e_t$$

Here $a_t = [a_{t,1}, \ldots, a_{t,p}]'$ is the p^{th} order AR coefficient vector and e_t is a Gaussian excitation at time t having variance σ_{et}^2. A Gaussian random walk model is assumed for the log-variance $\phi_{e_t} = \log(\sigma_{e_t}^2)$,

$$f(\phi_{e_t}|\phi_{e_{t-1}}, \sigma_{\phi_e}^2) = \mathcal{N}(\mu_{\phi_t}, \sigma_{\phi_e}^2) \tag{3}$$

where $\mu_{\phi_t} = \log(\alpha\sigma_{e_{t-1}}^2)$ and α is a coefficient just less than 1.

For the time variation in a_t, the simplest choice of all is perhaps a Gaussian random walk directly on the coefficients, that is

$$f(a_t|a_{t-1}) = \mathcal{N}(a_{t-1}, \sigma_a^2 I) \tag{4}$$

A model of this form is not constrained to be stable, which is an undesirable feature for audio. For sufficiently slow parameter variation, the stability criterion of a TVAR system is the same as for time invariant one (see, for example [22]), that is, stability is enforced by ensuring that all the instantaneous poles of the TVAR model, or the roots of the polynomial $(1 - \sum_{i=1}^{p} a_{t,i} z^{-i})$ lie strictly within the unit circle. This form of model was employed in the fixed-lag filtering work of [23].

As an alternative to direct coefficient modeling, one can reparameterise the model in terms of time-varying reflection coefficients (or equivalently partial correlation (PARCOR) coefficients) [10], [24]. The standard Levinson recursion [24] (see Appendix A) is used to transform between a_t and the reflection coefficient ρ_t. We adopt such a TV-PARCOR model in our simulations. This model is distinct to that employed in [23], which applies a constrained random walk directly in the TVAR coefficient domain, because we believe that it provides a better physical representation of audio signals. This arises since the TV-PARCOR model can be regarded as a time-varying acoustical tube mechanism, which is a reasonable approximation for speech and many musical instruments (see, for example [25]).

For our suggested TV-PARCOR model, approximate stability can be achieved by ensuring each reflection coefficient, ρ_t, is within the interval (-1,+1). The constrained random walk model for the time variation of ρ_t is

$$f(\rho_t|\rho_{t-1}, \sigma_a^2) \propto \begin{cases} \mathcal{N}(\rho_{t-1}, \sigma_a^2 I) & \text{if } \max_i\{|\rho_{t,i}|\} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\rho_t = [\rho_{t,1}, \ldots, \rho_{t,p}]'$. Our TV-PARCOR model can be contrasted with that of Kitagawa *et al.* [3], in which an independent Gaussian random walk model is applied to both the forward and backward reflection coefficients. In our model, the backward reflection coefficients are constrained to equal the forward coefficients. This reflects a belief that the shape of the human vocal tract is slowly time-varying owing to physical constraints, and hence the forward reflection coefficient should approximately equal the backward reflection coefficient.

The full specification of the state space model is then as follows. The state vector x_t is partitioned as $[z_t, \theta_t]'$ with $z_t = u_{t-p+1:t}$ and θ_t being the signal state and the parameter state respectively. The signal is assumed to be submerged in white Gaussian noise (WGN) with known variance σ_v^2, *i.e.*

$$g(y_t|x_t) = \mathcal{N}(y_t; u_t, \sigma_v^2)$$

The parameter vector θ_t is further partitioned as $[a_t, \phi_{e_t}]'$. Hyperparameters σ_a^2 are $\sigma_{\phi_e}^2$ are assumed to be prespecified and fixed in all the simulations. The initial state probability is diffuse Gaussian over the stable region for the model.

III. Monte Carlo Filtering and Smoothing Algorithms

The earliest references to Monte Carlo filtering go back many decades (see [26] and citations in [6]). The 1990s has seen a dramatic resurgence of interest in these filters, owing largely to the massive increases in available computational power (see for example [4], [7], [27]). Refer to [5] for an up-to-date survey of the field. In this section, we will briefly review generic filtering and smoothing algorithms.

Consider the filtering distribution $p(x_t|y_{1:t})$, which can be rearranged as follows,

$$\begin{aligned} p(x_t|y_{1:t}) &\propto g(y_t|x_t) p(x_t|y_{1:t-1}) \\ &= \int g(y_t|x_t) f(x_t|x_{t-1}) p(x_{1:t-1}|y_{1:t-1}) dx_{1:t-1} \end{aligned} \tag{6}$$

Assuming that a particle approximation to $p(x_{1:t-1}|y_{1:t-1})$ has already been generated,

$$p(x_{1:t-1}|y_{1:t-1}) \approx \sum_{i=1}^{N} \delta(x_{1:t-1} - x_{1:t-1}^{(i)})$$

Then, assuming we can evaluate $f(x_t|x_{t-1})$ and $g(y_t|x_t)$ pointwise, we generate, for each state trajectory $x_{1:t-1}^{(i)}$, a random sample from a proposal distribution, $x_t^{(i)} \sim q(x_t|x_{1:t-1}^{(i)}, y_{1:t})$. The filtering distribution (6) can then be approximated as

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)})$$

with

$$w_t^{(i)} \propto \frac{g(y_t|x_t^{(i)}) f(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{1:t-1}^{(i)}, y_{1:t})}$$

For example, in some simple scenarios, the proposal distribution can be chosen as $q(x_t|x_{1:t-1}^{(i)}, y_{1:t}) = f(x_t|x_{t-1}^{(i)})$, so the importance weight reduces to $w_t^{(i)} \propto g(y_t|x_t^{(i)})$. If a resampling step [7] is performed, such that the probability that $x_t^{(i)}$ is selected is proportional to $w_t^{(i)}$, then the resulting samples are an unweighted approximate random draw from the filtering distribution $p(x_t|y_{1:t})$.

The generic particle filtering algorithm, with the proposal distribution $q(x_t|x_{1:t-1}^{(i)}, y_{1:t}) = f(x_t|x_{t-1}^{(i)})$, is given below,

Algorithm 1: **Generic particle filter with resampling** [5], [6], [7], [9]

Let $f(x_1|x_0) = f(x_1)$ be the state prior distribution. Then for $t = 1$ to T:

- For $i = 1, \ldots, N$, generate N samples from the proposal distribution $q(x_t|x_{1:t-1}^{(i)}, y_{1:t}) = f(x_t|x_{t-1}^{(i)})$,

$$x_t^{(i)} \sim f(x_t|x_{t-1}^{(i)})$$

- For $i = 1, \ldots, N$, evaluate the importance weights and normalise:

$$w_t^{(i)} \propto g(y_t|x_t^{(i)}), \qquad \sum_{i=1}^{N} w_t^{(i)} = 1$$

- Resample $\{x_t^{(i)}; i = 1, \ldots, N\}$ N times with replacement. The probability of $x_t^{(i)}$ being selected is equal to $w_t^{(i)}$.

Note that the final resampling step is in practice usually replaced with a part-deterministic selection step in order to improve the Monte Carlo efficiency [5].

In addition to the Monte Carlo filtering algorithm, Monte Carlo smoothing methods have been developed to generate realisations from the smoothing density. Earlier existing approaches include approximating the individual marginal smoothing distribution $p(x_t|y_{1:T})$, either using the two-filter formula [8] or forward filtering-backward smoothing [6], [28]. As an alternative, Tanizaki [29] has recently proposed methods for generating random draws from the the joint distribution $p(x_t, x_{t+1}|y_{1:T})$ or $p(x_{t-1}, x_t, x_{t+1}|y_{1:T})$. As with the methods of [6], [28] these techniques require the simulation of expensive normalising constants, and the focus is on *marginal* state distributions rather than joint distributions over time. In this paper, we propose improvements to a recently developed algorithm [10], [11] in which samples are drawn from the joint smoothing density $p(x_{1:T}|y_{1:T})$. In its original form, sample realisations are obtained using the following factorisation

$$p(x_{1:T}|y_{1:T}) = p(x_T|y_{1:T}) \prod_{t=1}^{T-1} p(x_t|x_{t+1:T}, y_{1:T}) \quad (7)$$

where, given the particle approximation to $p(x_t|y_{1:t})$ and using the Markovian assumptions of the model, we can write,

$$\begin{aligned} p(x_t|x_{t+1:T}, y_{1:T}) &\propto p(x_t|y_{1:t}) f(x_{t+1}|x_t) \quad (8) \\ &\approx \sum_{i=1}^{N} w_{t|t+1}^{(i)} \delta(x_t - x_t^{(i)}) \end{aligned}$$

with the modified weights

$$w_{t|t+1}^{(i)} = \frac{w_t^{(i)} f(x_{t+1}|x_t^{(i)})}{\sum_{j=1}^{N} w_t^{(j)} f(x_{t+1}|x_t^{(j)})}. \quad (9)$$

This revised particle distribution can now be used to generate states successively in the reverse-time direction, conditioning upon future states. It should be noted that, in contrast with [6], [28], [29] the method does not require the simulation of any additional normalising constants, since the modified weights are easily renormalised.

Having performed a forward sweep of particle filtering, generating weighted particles $\{x_t^{(i)}, w_t^{(i)}; i = 1, \ldots, N, t = 1, \ldots, T\}$ the smoothing algorithm proceeds as follows,

Algorithm 2: **Generic particle smoother** [10], [11]

- Choose $\widetilde{x}_T = x_T^{(i)}$ with probability $w_T^{(i)}$.
- For $t = T-1$ to 1 :
  - Calculate $w_{t|t+1}^{(i)} \propto w_t^{(i)} f(\widetilde{x}_{t+1}|x_t^{(i)})$ for each $i = 1, ..., N$;
  - Choose $\widetilde{x}_t = x_t^{(i)}$ with probability $w_{t|t+1}^{(i)}$.
- $\widetilde{\mathbf{x}}_{1:T} = (\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_T)$ is an approximate realisation from $p(x_{1:T}|y_{1:T})$.

Further independent realisations are obtained by repeating this procedure as many times as required.

IV. Rao-Blackwellised particle filtering and smoothing

One of the major drawbacks of any Monte Carlo filtering / smoothing strategy is that sampling in high-dimensional spaces can be inefficient. In some cases, however, the model has "tractable substructure" [30], which can be analytically marginalised out, conditional on other state variables. This applies for example to our noise reduction model in section II, in which the signal process $\{z_t\}$ can be integrated out conditional upon the values of the parameter process $\{\theta_t\}$. The advantage of this strategy is that it can drastically reduce the size of the space over which we need to sample, and the variance of the resulting estimates is then lower than that of the standard particle filtering approach [4], [6].

Marginalising out some of the variables is an example of a standard statistical variance reduction strategy known as Rao-Blackwellisation, see [31] for a general discussion on the topic. Rao-Blackwellised particle filters have been applied in specific contexts such as mixtures of Gaussians [4], [6], fixed parameter estimation [32] and Dirichlet process models [33]. Vermaak *et al.* [23] have used Rao-Blackwellisation for fixed-lag smoothing in speech signals. In their approach, a Rao-Blackwellised particle filter is applied forwards through time, and fixed lag smoothing is achieved by the use of Markov Chain Monte Carlo (MCMC) "resample-move" steps [34] over a small time lag of samples.

In this paper we focus on applying Rao-Blackwellisation to fixed-interval smoothing; that is, given $y_{1:T}$, we would like to simulate from the entire state density $p(x_{1:T}|y_{1:T})$. The reason for this is that a greater degree of smoothing can be important for the convincing reconstruction of audio signals. In section V we describe how to make our algorithm more practical for very large datasets by performing the smoothing sequentially in sub-blocks, or frames, rather than over the entire data set each time.

A. Rao-Blackwellised Particle Filter

First we review the standard Rao-Blackwellised particle filter [30], [6]. Assume that the state vector $x_{1:t}$ can be partitioned as $[z_{1:t}, \theta_{1:t}]$ and $z_{1:t}$ can be marginalised out analytically. For instance, if conditional on $\theta_{1:t}$, $z_{1:t}$ reduces to a linear Gaussian state-space system, then all the integration can be performed analytically on-line using the Kalman filter and the prediction error decomposition (for a general reference on the Kalman filter and prediction error decomposition, refer to [1], [2]).

Let us consider the marginal filtering distribution,

$$\begin{aligned} p(\theta_t|y_{1:t}) &= \int p(z_t, \theta_t|y_{1:t})dz_t \\ &\propto \int p(y_t|\theta_{1:t}, y_{1:t-1})f(\theta_t|\theta_{t-1})\times \\ &\quad \times p(\theta_{1:t-1}|y_{1:t-1})d\theta_{1:t-1} \end{aligned}$$

where

$$p(y_t|\theta_{1:t}, y_{1:t-1}) = \int p(z_t, y_t|\theta_{1:t}, y_{1:t-1})dz_t$$

Given the particle approximation to $p(\theta_{1:t-1}|y_{1:t-1})$, new particles $\theta_t^{(i)}$ are generated from $f(\theta_t|\theta_{t-1}^{(i)})$, and $p(\theta_t|y_{1:t})$ is approximated by

$$p(\theta_t|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)}\delta(\theta_t - \theta_t^{(i)})$$

with

$$w_t^{(i)} \propto p(y_t|\theta_{1:t}^{(i)}, y_{1:t-1})$$

Under the assumption of a conditionally linear Gaussian structure, $p(y_t|\theta_{1:t}, y_{1:t-1})$ can be evaluated efficiently using the Kalman filter and the prediction error decomposition, as outlined below.

Conditional on the parameter vector θ_t the signal z_t is linear and Gaussian, facilitating a conditionally Gaussian state-space representation (a special case of (1)),

$$z_t = \mathcal{T}(\theta_t)z_{t-1} + \mathcal{H}(\theta_t)\omega_t \tag{10a}$$
$$y_t = \mathcal{Z}(\theta_t)z_t + \mathcal{G}(\theta_t)\omega_t \tag{10b}$$

where $\mathcal{T}(\theta_t)$, $\mathcal{H}(\theta_t)$, $\mathcal{Z}(\theta_t)$ and $\mathcal{G}(\theta_t)$ are system matrices associated with the linear state space model taking appropriate forms (see [23], [35] for details in the TVAR case - the same results apply for our TV-PARCOR model once reflection coefficients have been transformed to TVAR coefficients via the Levinson-Durbin recursion). ω_t is defined as $\omega_t \triangleq [e_t, v_t]'$.

Combining the Kalman filter equations described in Appendix B with the ordinary particle filtering equations produces the following Rao-Blackwellised filtering algorithm:

Algorithm 3: **Rao-Blackwellised particle filter** [4], [6]

Let $f(\theta_1|\theta_0) = f(\theta_1)$ and initialise $\{\zeta_{1|0}, P_{1|0}\}$ with $\{\zeta_{1|1}, P_{1|1}\}$. For $t = 1$ to T

- For $i = 1, \ldots, N$, generate N parameter samples from the appropriate proposal distribution $\theta_t^{(i)} \sim f(\theta_t|\theta_{t-1}^{(i)})$.
- For each realisation of $\theta_{1:t}^{(i)}$, update the sufficient statistics for z_t using the Kalman filter:
  – plug $\theta_t^{(i)}$ into the system matrices, $\mathcal{T}(\theta_t^{(i)})$, $\mathcal{G}(\theta_t^{(i)})$, $\mathcal{Z}(\theta_t^{(i)})$, $\mathcal{H}(\theta_t^{(i)})$ and $\mathcal{Q}(\theta_t^{(i)})$.
  – compute the sufficient statistics, ζ and P, for the following distributions,

$$p(z_t|\theta_{1:t}, y_{1:t-1}) = \mathcal{N}(z_t; \zeta_{t|t-1}^{(i)}, P_{t|t-1}^{(i)})$$
$$p(z_t|\theta_{1:t}, y_{1:t}) = \mathcal{N}(z_t; \zeta_{t|t}^{(i)}, P_{t|t}^{(i)})$$

where $[\zeta_{t|t-1}^{(i)}, P_{t|t-1}^{(i)}]$ are given by one-step ahead prediction (16) and $[\zeta_{t|t}^{(i)}, P_{t|t}^{(i)}]$ are given by the estimation update (17).

- For $i = 1, \ldots, N$ evaluate the importance weight $w_t^{(i)}$ associated with $\theta_t^{(i)}$ using the prediction error decomposition as given by (18), according to [2]

$$w_t^{(i)} \propto \mathcal{N}(y_t|y_{t|t-1}^{(i)}, \mathcal{F}_t^{(i)})$$

- Resample $\{\theta_t^{(i)}, \zeta_{t|t}^{(i)}, P_{t|t}^{(i)}; i = 1, \ldots, N\}$ N times with replacement. The probability of $\{\theta_t^{(i)}, \zeta_{t|t}^{(i)}, P_{t|t}^{(i)}\}$ being selected is equal to $w_t^{(i)}$.

B. Rao-Blackwellised Particle Smoother

As previously discussed, the state estimate can be further improved by performing smoothing. We modify the generic particle smoothing algorithm to incorporate Rao-Blackwellisation to form a Rao-Blackwellised Particle Smoother.

We want to sample from the entire state density $p(z_{1:T}, \theta_{1:T}|y_{1:T})$ which can be factorised as follows

$$\begin{aligned} & p(z_{1:T}, \theta_{1:T}|y_{1:T}) \\ & = p(z_T, \theta_T|y_{1:T}) \prod_{t=1}^{T-1} p(z_t, \theta_t|z_{t+1:T}, \theta_{t+1:T}, y_{1:T}) \end{aligned}$$

and

$$\begin{aligned} & p(z_t, \theta_t|z_{t+1:T}, \theta_{t+1:T}, y_{1:T}) \\ & = \int p(z_t, \theta_{1:t}|z_{t+1:T}, \theta_{t+1:T}, y_{1:T}) d\theta_{1:t-1} \\ & = \int p(\theta_{1:t}|z_{t+1:T}, \theta_{t+1:T}, y_{1:T}) p(z_t|z_{t+1:T}, \theta_{1:T}, y_{1:T}) d\theta_{1:t-1} \end{aligned} \quad (11)$$

Using the particulate approximation for the parameter filtering distribution $p(\theta_{1:t}|y_{1:t})$, given by the forward sweep of the Rao-Blackwellised particle filter, the marginal smoothing distribution $p(\theta_{1:t}|z_{t+1:T}, \theta_{t+1:T}, y_{1:T})$ is approximated by

$$p(\theta_{1:t}|z_{t+1:T}, \theta_{t+1:T}, y_{1:T}) \approx \sum_{i=1}^{N} w_{t|t+1}^{(i)} \delta(\theta_{1:t} - \theta_{1:t}^{(i)}) \quad (12)$$

with modified weight $w_{t|t+1}^{(i)}$ as follows:

$$\begin{aligned} w_{t|t+1}^{(i)} & \propto w_t^{(i)} p(z_{t+1}, \theta_{t+1}|\theta_{1:t}^{(i)}, y_{1:t}) \quad (13) \\ & \propto w_t^{(i)} f(\theta_{t+1}|\theta_t^{(i)}) p(z_{t+1}|\theta_{t+1}, \theta_{1:t}^{(i)}, y_{1:t}) \end{aligned}$$

Using this particulate approximation for the parameter smoothing distribution, the joint smoothing density (11) can be approximated by,

$$\begin{aligned} & p(z_t, \theta_t|z_{t+1:T}, \theta_{t+1:T}, y_{1:T}) \\ & \approx \int p(z_t|z_{t+1:T}, \theta_{1:T}, y_{1:T}) \sum_{i=1}^{N} w_{t|t+1}^{(i)} \delta(\theta_{1:t} - \theta_{1:t}^{(i)}) d\theta_{1:t-1} \\ & \approx \sum_{i=1}^{N} w_{t|t+1}^{(i)} p(z_t|z_{t+1:T}, \theta_{1:t}^{(i)}, \theta_{t+1:T}, y_{1:T}) \delta(\theta_t - \theta_t^{(i)}) \quad (14) \end{aligned}$$

We now prove the correctness of the Rao-Blackwellised approximation for the parameter smoothing distribution.

Proof: Using the Markovian assumptions of the model, the marginal smoothing density can be rewritten as follows

$$\begin{aligned} & p(\theta_{1:t}|z_{t+1:T}, \theta_{t+1:T}, y_{1:T}) \\ & = p(\theta_{1:t}|z_{t+1}, \theta_{t+1}, y_{1:t}) \\ & = \frac{p(z_{t+1}, \theta_{t+1}|\theta_{1:t}, y_{1:t}) p(\theta_{1:t}|y_{1:t})}{\int p(z_{t+1}, \theta_{t+1}|\theta_{1:t}, y_{1:t}) p(\theta_{1:t}|y_{1:t}) d\theta_{1:t}} \\ & = \frac{p(z_{t+1}|\theta_{1:t+1}, y_{1:t}) f(\theta_{t+1}|\theta_t) p(\theta_{1:t}|y_{1:t})}{\int p(z_{t+1}|\theta_{1:t+1}, y_{1:t}) f(\theta_{t+1}|\theta_t) p(\theta_{1:t}|y_{1:t}) d\theta_{1:t}} \end{aligned}$$

Using the particle approximation from the forward sweep of the Rao-Blackwellised particle filter

$$p(\theta_{1:t}|y_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(\theta_{1:t} - \theta_{1:t}^{(i)})$$

we now have,

$$\begin{aligned} & p(\theta_{1:t}|z_{t+1:T}, \theta_{t+1:T}, y_{1:T}) \\ & \approx \sum_{i=1}^{N} \frac{w_t^{(i)} p(z_{t+1}|\theta_{1:t}^{(i)}, \theta_{t+1}, y_{1:t}) f(\theta_{t+1}|\theta_t^{(i)})}{\sum_{j=1}^{N} w_t^{(j)} p(z_{t+1}|\theta_{1:t}^{(j)}, \theta_{t+1}, y_{1:t}) f(\theta_{t+1}|\theta_t^{(j)})} \delta(\theta_{1:t} - \theta_{1:t}^{(i)}) \\ & = \sum_{i=1}^{N} w_{t|t+1}^{(i)} \delta(\theta_{1:t} - \theta_{1:t}^{(i)}) \end{aligned}$$

where the normalised weight is

$$w_{t|t+1}^{(i)} \propto w_t^{(i)} p(z_{t+1}|\theta_{1:t}^{(i)}, \theta_{t+1}, y_{1:t}) f(\theta_{t+1}|\theta_t^{(i)})$$

as required. ■

We now show the way to draw approximate smoothed realisations $\{\tilde{z}_t, \tilde{\theta}_t; 1, \ldots, T\}$ from (14). Given $\tilde{\theta}_{t+1:T}$ and $\tilde{z}_{t+1:T}$, $\tilde{\theta}_t$ can be drawn from

$$\tilde{\theta}_{1:t} \sim \sum_{i=1}^{N} w_{t|t+1}^{(i)} \delta(\theta_{1:t} - \theta_{1:t}^{(i)})$$

in a similar manner to the generic particle smoother (algorithm 2). And say, we obtain $\tilde{\theta}_{1:t} = \theta_{1:t}^{(j)}$, then the smoothed signal realisation $\tilde{z}_t$ is obtained by sampling from the conditional density function $p(z_t|\theta_{1:t}^{(j)}, \tilde{z}_{t+1:T}, \tilde{\theta}_{t+1:T}, y_{1:T})$.

Under the assumption of a conditionally Gaussian structure for the signal, the modified weight (13) takes the form

$$w_{t|t+1}^{(i)} \propto w_t^{(i)} f(\tilde{\theta}_{t+1}|\theta_t^{(i)}) \mathcal{N}(\tilde{z}_{t+1}; \zeta_{t+1|t}^{(i)}, P_{t+1|t}^{(i)})$$

where $\zeta_{t+1|t}^{(i)}$ and $P_{t+1|t}^{(i)}$ can be computed efficiently using the one-step ahead equation (16). In addition, $p(z_t|\theta_{1:t}^{(j)}, \tilde{z}_{t+1:T}, \tilde{\theta}_{t+1:T}, y_{1:T})$ is a Gaussian distribution

$$p(z_t|\theta_{1:t}^{(j)}, \tilde{\theta}_{t+1:T}, \tilde{z}_{t+1:T}, y_{1:T}) = \mathcal{N}(z_t; \tilde{\zeta}_{t|T}, \tilde{P}_{t|T})$$

where $\tilde{\zeta}_{t|T}$ and $\tilde{P}_{t|T}$ can be evaluated efficiently using one step of the Kalman smoother, which in turn use the sufficient statistics for $p(z_t|\theta_{1:t}^{(j)}, y_{1:t})$ ($\tilde{\zeta}_{t|t}^{(j)}$ and $\tilde{P}_{t|t}^{(j)}$) stored in the forward pass of the Rao-Blackwellised particle filter.

By repeating the sampling process

$$\begin{aligned} \tilde{\theta}_t & \sim \sum_{i=1}^{N} w_{t|t+1}^{(i)} \delta(\theta_t - \theta_t^{(i)}) \\ \tilde{z}_t & \sim \mathcal{N}(z_t; \tilde{\zeta}_{t|T}, \tilde{P}_{t|T}) \end{aligned}$$

recursively backward in time, approximate samples $[\tilde{z}_{1:T}, \tilde{\theta}_{1:T}]$ are drawn from $p(z_{1:T}, \theta_{1:T}|y_{1:T})$.

Assume that a forward sweep of Rao-Blackwellised particle filtering has been performed as shown in algorithm 3, generating weighted particles and sufficient statistics $\{w_t^{(i)}, \theta_t^{(i)}, \zeta_{t|t}^{(i)}, P_{t|t}^{(i)}; i = 1, \ldots, N, t = 1, \ldots, T\}$. Rao-Blackwellisation is then applied to generate state realisations:

Algorithm 4: **Rao-Blackwellised Particle Smoother**

• Choose $\tilde{\theta}_T = \theta_T^{(j)}$ with probability $w_T^{(j)}$ and set the signal statistics $[\tilde{\zeta}_{T|T}, \tilde{P}_{T|T}]$ as $[\zeta_{T|T}^{(j)}, P_{T|T}^{(j)}]$. $\tilde{z}_T$ is generated by sampling from the following distribution

$$\tilde{z}_T \sim \mathcal{N}(\tilde{\zeta}_{T|T}, \tilde{P}_{T|T})$$

• For $t = T-1$ to 1
 – Particle smoothing:
 ∗ given $\tilde{\theta}_{t+1}$ and $[\zeta_{t|t}^{(i)}, P_{t|t}^{(i)}]$ (sufficient statistics for $p(z_t|\theta_{1:t}^{(i)}, y_{1:t})$ stored in the forward pass of the Rao-Blackwellised particle filter), compute $[\zeta_{t+1|t}^{(i)}, P_{t+1|t}^{(i)}]$ using the one-step ahead prediction equation (16) for $i = 1, \ldots, N$;
 ∗ calculate $w_{t|t+1}^{(i)} \propto w_t^{(i)} \mathcal{N}(\tilde{z}_{t+1}; \zeta_{t+1|t}^{(i)}, P_{t+1|t}^{(i)}) f(\tilde{\theta}_{t+1}|\theta_t^{(i)})$ for $i = 1, \ldots, N$;
 ∗ sample the indicator $j = i$ with probability $w_{t|t+1}^{(i)}$ and set $\tilde{\theta}_t = \theta_t^{(j)}$.
 – Kalman smoothing:
 ∗ set $[\tilde{\zeta}_{t|t}, \tilde{P}_{t|t}, \tilde{P}_{t+1|t}] = [\zeta_{t|t}^{(j)}, P_{t|t}^{(j)}, P_{t+1|t}^{(j)}]$ and compute

$$\begin{aligned}
\tilde{\zeta}_{t|T} &= \tilde{\zeta}_{t|t} + \mathcal{S}_t \left(\tilde{\zeta}_{t+1|T} - \mathcal{T}(\tilde{\theta}_{t+1}) \tilde{\zeta}_{t|t} \right) \\
\tilde{P}_{t|T} &= \tilde{P}_{t|t} + \mathcal{S}_t (\tilde{P}_{t+1|T} - \tilde{P}_{t+1|t}) \mathcal{S}_t'
\end{aligned}$$

with $\mathcal{S}_t = \tilde{P}_{t|t} \mathcal{T}(\tilde{\theta}_{t+1})' \tilde{P}_{t+1|t}^{-1}$. $\tilde{z}_t$ is then drawn from the following distribution, $\tilde{z}_t \sim \mathcal{N}(\tilde{\zeta}_{t|T}, \tilde{P}_{t|T})$.
 – set $\tilde{x}_t = [\tilde{z}_t, \tilde{\theta}_t]$.

$\tilde{x}_{1:T} = [\tilde{z}_{1:T}, \tilde{\theta}_{1:T}]$ is an approximate realisation of $p(x_{1:T}|y_{1:T})$. Further independent realisations are obtained by repeating the procedure described above as many times as required.

V. Block-based Particle Smoother

The particle filter is a computationally efficient algorithm due to its potential for parallel processing. In addition, particles generated at the current time step depend on those of the previous time step only, so no particles further back in time need to be stored in a pure filtering application. The particle smoother, however, requires storage of information from the entire particle histories. It is thus inapplicable for lengthy datasets such as audio signals unless we compromise the number of particles used to approximate the filtering distribution, due to the ever-growing storage and processing capacity required.

In this section, we propose a modification to the general smoothing algorithm in order to process a lengthy dataset in blocks. The advantage of our proposed method is that it significantly reduces the memory capacity required and allows smoothing computations to be carried out 'on-line' in a block-by-block fashion. This is because only particle filter information regarding the current block needs to be stored for the smoothing operation, while the ordinary particle smoother requires particle information for the whole time series.

A lengthy time series is divided into two R non-overlapping blocks, with $T_1, T_2, \ldots, T_r, \ldots, T_R$ marking the end of each block (the block lengths can potentially be different from block to block). For each block of data, a forward sweep of particle filtering is performed, generating weighted particles $\{x_t^{(i)}, w_t^{(i)}; i = 1, \ldots, N, t = 1, \ldots, T\}$ (refer to algorithms 1 and 3 for details) with T being the number of data point in the block concerned. The particle filter is initialised at the start of each new block with the final weighted particles from the previous block. A smoothing analysis is then performed in which M realisations are generated from the smoothing density for that block of data. For example, after processing the first two blocks, $r = 1, 2$, we get smoothed realisations,

$$\begin{aligned}
\tilde{x}_{1:T_1}^{(i)} &\sim p(x_{1:T_1}|y_{1:T_1}) \\
\tilde{x}_{T_1+1:T_2}^{(j)} &\sim p(x_{T_1+1:T_2}|y_{1:T_2})
\end{aligned}$$

with $i, j \in \{1, \ldots, M\}$. We require, as before, realisations from the entire smoothing density, which can be factorised as follows

$$p(x_{1:T_2}|y_{1:T_2}) = p(x_{1:T_1}|x_{T_1+1:T_2}, y_{1:T_2}) p(x_{T_1+1:T_2}|y_{1:T_2})$$

where, using the Markovian assumptions of the dynamical model, we can write,

$$\begin{aligned}
p(x_{1:T_1}|x_{T_1+1:T_2}, y_{1:T_2}) &= \frac{p(x_{T_1+1}|x_{1:T_1}, y_{1:T_1}) p(x_{1:T_1}|y_{1:T_1})}{p(x_{T_1+1}|y_{1:T_1})} \\
&\propto f(x_{T_1+1}|x_{T_1}) p(x_{1:T_1}|y_{1:T_1})
\end{aligned}$$

The smoothing density $p(x_{1:T_1}|y_{1:T_1})$ can be approximated using the output from the particle smoother for $M >> 1$,

$$p(x_{1:T_1}|y_{1:T_1}) \approx \sum_{i=1}^{M} \delta(x_{1:T_1} - \tilde{x}_{1:T_1}^{(i)})$$

Hence, for each realisation from block 2, $\tilde{x}_{T_1+1:T_2}^{(j)}$, we can generate an approximate realisation from the entire smoothing sequence by drawing $x_{1:T_1}$ from the following discrete distribution:

$$p(x_{1:T_1}|\tilde{x}_{T_1+1:T_2}^{(j)}, y_{1:T_2}) \approx \sum_{i=1}^{M} w_{T_1|T_1+1}^{(i|j)} \delta(x_{1:T_1} - \tilde{x}_{1:T_1}^{(i)})$$

with the associated weight being

$$w_{T_1|T_1+1}^{(i|j)} \propto f(\tilde{x}_{T_1+1}^{(j)}|\tilde{x}_{T_1}^{(i)})$$

Thus particle $\tilde{x}^{(i)}_{1:T_1}$ is joined to $\tilde{x}^{(j)}_{T_1+1:T_2}$ with probability $w^{(i|j)}_{T_1|T_1+1}$, and an approximate realisation from the entire smoothing density $\{x^{(j)}_{1:T_2} = [\tilde{x}^{(i)}_{1:T_1}, \tilde{x}^{(j)}_{T_1+1:T_2}]; j = 1, \ldots, M\}$ is obtained. We have thus avoided the need to repeat the backwards smoothing pass over times 1 to T_1 (hence 'on-line' smoothing), and significant memory storage savings are achieved, since the particle filter output from times 1 to T_1 can now be discarded.

Algorithm 5: **Block-based smoother**

- Divide the lengthy times series into non-overlapping blocks, with $T_1, \ldots, T_R$ marking the end of each block.
- Perform particle filtering and smoothing analysis on the first block, and generate $\tilde{x}^{(i)}_{1:T_1}$ for $i = 1, \ldots, M$.
- For $r = 2$ to R:
  – Perform particle filtering and smoothing analysis on block r, generate M realisations of $\tilde{x}_{T_{r-1}+1:T_r} \sim p(x_{T_{r-1}+1:T_r}|y_{1:T_r})$.
  – For each $\tilde{x}^{(j)}_{T_{r-1}+1:T_r}$, calculate the transition probability $w^{(i|j)}_{T_{r-1}|T_{r-1}+1} \propto f(\tilde{x}^{(j)}_{T_{r-1}+1}|\tilde{x}^{(i)}_{T_{r-1}})$ for each $\tilde{x}^{(i)}_{T_1:T_{r-1}}$, $i, j = 1, \ldots, M$;
  – Choose $\tilde{x}^{(j)}_{1:T_{r-1}} = \tilde{x}^{(i)}_{1:T_{r-1}}$ with probability $w^{(i|j)}_{T_{r-1}|T_{r-1}+1}$ and set $\tilde{x}^{(j)}_{1:T_r} = [\tilde{x}^{(j)}_{1:T_{r-1}}, \tilde{x}^{(j)}_{T_{r-1}+1:T_r}]$.

In theory a very large value of M is likely to be required in order for the block-based smoother to approximate well an optimal smoother. However, for the application demonstrated later, a rather small value of M was found to give excellent results. This conclusion may not of course carry through into different applications.

As stated above, once each non-overlapping block has been processed, all information regarding the weighted filter particles can be discarded and thus the storage capacity required is reduced significantly when $M \ll N$. If we assume that any particle will take up two units of memory space (a unit for the particle value and a unit for its weight), the fraction of memory space saved is given by:

$$1 - \frac{1}{R}\left(\frac{(R-1)M}{2N} + 1\right) \tag{15}$$

on condition that the time series is divided into R equal size blocks.

VI. Experimental Results

Extensive tests have been carried out to investigate the effectiveness of the suggested algorithms using a variety of audio datasets (refer to *http://www-sigproc.eng.cam.ac.uk/∼wnwf2*). It is found that our proposed block-based Rao-Blackwellised (TV-PARCOR) particle smoother consistently outperforms the classical extended Kalman filter / smoother and the generic particle filter / smoother. Some representative examples of the tests conducted are described in detail here:

- Test 1: test the effectiveness of the block-based particle smoother using a short section of speech data.
- Test 2: compare the extended Kalman filter / smoother, the generic particle filter / smoother and the Rao-Blackwellised Particle Smoother, using a section of noisy speech and a section of vocal music data.
- Test 3: verify our suggestion that TV-PARCOR model gives improved performance for audio signals compared with a standard TVAR parameterisation, using two different pieces of high quality music.
- Test 4: compare the Rao-Blackwellised particle smoother with the extended Kalman filter / smoother using a long section of high quality music at various input signal-to-noise ratios (SNR).

In each case, differences in audio quality and changes in SNR are noted and compared, for which, the input SNR and output SNR are defined by,

$$SNR_{in} = 10 \log \frac{\sum_{t=1}^{T} u_t^2}{\sum_{t=1}^{T} (y_t - u_t)^2}$$

$$SNR_{out} = 10 \log \frac{\sum_{t=1}^{T} u_t^2}{\sum_{t=1}^{T} (\hat{u}_t - u_t)^2}$$

where $u_{1:T}$ and $y_{1:T}$ are the clean audio signal level and audio signal corrupted with white Gaussian noise respectively. $\hat{u}_{1:T}$ is the estimated audio signal waveform from different algorithms. For example, in the case of the Monte Carlo smoother, the estimated signal waveform is obtained as the arithmetic mean of the M smoothed realisations, which, for small M, we can think of as a rough-and-ready approximation to the true posterior mean estimate of $u_{1:T}$.

A. Test 1 — Block-based Particle Smoother

A small section of speech data representing the word "reward" is presented in Figure 1, which is used to test whether the block-based particle smoother will introduce any undesirable artifacts.

Due to the practical limitations in storage capacity available, only a small section of data could be used for this test, which involves the memory-expensive generic particle smoother. Hence, it is seen that the proposed block-based processing algorithm will be crucial in practice.

In this test, we use the Rao-Blackwellised particle smoother employing the TV-PARCOR random walk model (5) with order $p = 6$. The variance of the white Gaussian noise signal is assumed to be known and chosen to make the input SNR equals $3dB$. Other fixed hyperparameters used are $\sigma_a^2 = 0.001$, $\alpha = 0.955$ and $\sigma_{\phi_e}^2 = 0.01$. For the particle filter / smoother setup, different combinations of N, number of particles, and M, number of smoothing realisations generated, are tested.

For M fixed to 10, the output SNR for the ordinary Rao-Blackwellised particle smoother and the block-based Rao-Blackwellised particle smoother using different number of particles, N, is presented in Figure 2. As shown in Figure 2, the block-based Rao-Blackwellised particle smoother gives almost the same degree of enhancement as the ordinary Rao-Blackwellised particle smoother for N greater than 50. It was also found that no further improvement was observed for N greater 100. As a result, N is fixed to 100 for the rest of simulations.

For N fixed to 100, the effect of different M on the output SNR is presented in Figure 3. As shown in Figure 3, the block-based Rao-Blackwellised particle smoother gives similar SNR improvement to the ordinary Rao-Blackwellised particle smoother. In consideration of the sample realisations diversity and the processing time, M is chosen to be 10 for the rest of the simulation. We note that $M = 10$ is an extremely small sample size for general Monte Carlo estimation. However, since the performance in terms of SNR does not improve for larger M, we adopt $M = 10$ for computational reasons.

Comparing the Rao-Blackwellised particle filter output and the two smoother outputs, a substantial improvement in sound quality is observed for both smoothers over the filter. In addition, the differences between the ordinary and block-based Rao-Blackwellised particle smoother are hardly audible. The block-based Rao-Blackwellised particle smoother, however, requires far less memory capacity.

B. Test 2 — Rao-Blackwellised Particle Smoother

A long section of speech data representing the words "Good service should be rewarded by big tips" (as used in [21], [23]) is used to compare the performance of the extended Kalman filter / smoother, the generic particle smoother and the Rao-Blackwellised particle smoother. Figure 4 shows a typical frame of the speech data used and its corresponding noise corrupted version, with input SNR of 10 dB. The time-varying characteristics of the signal are clearly evident.

As in the previous test, the variance of the white Gaussian noise signal is assumed to be known and fixed, $\sigma_v^2 = 0.001$ and the remaining fixed hyperparameters used are $\sigma_a^2 = 10^{-4}$, $\alpha = 0.995$ and $\sigma_{\phi_e}^2 = 10^{-6}$. Finally, the TVAR model order is fixed to $p = 6$.

For the extended Kalman filter / smoother, a Gaussian random walk is assumed directly on the AR coefficients as in (4) and stability is not enforced. This model is used because the TV-PARCOR model is not straightforward to implement with the extended Kalman filter; indeed, this is a major justification for adopting computer intensive methods such as the particle filter / smoother for the TV-PARCOR model. For the generic particle smoother and the Rao-Blackwellised Particle Smoother, the TV-PARCOR random walk model is used as before. In addition, due to the number of data involved, the block-based smoothing algorithm is adopted, with block size being 3000. $N = 100$ particles are used and smoothing is applied to generate $M = 10$ realisations.

The input SNR of the noisy speech data is 10.2 dB and the output SNR of the three algorithms are summarised below:

Extended Kalman Smoother	12.1 dB
Generic Particle Smoother	10.8 dB
Rao-Blackwellised Particle Smoother	13.3 dB

Reconstructed signals using different algorithms are presented in Figure 4. For the audio quality the ordinary particle smoother gives the least audible improvement. For the extended Kalman filter's output, there are still some audible artifacts and sibilant sounds. The Rao-Blackwellised Particle smoother can eliminate almost all of the white Gaussian noise; however, it gives a slightly higher sibilant noise than the output using the extended Kalman filter. In terms of SNR, the Rao-Blackwellised particle smoother gives best performance.

We re-ran the test using a section of high quality vocal music. The experimental setup is the same as in Test 1. The input SNR of the noisy audio clip is 10.5 dB and the output SNR is summarised below:

Extended Kalman Smoother	13.6 dB
Generic Particle Smoother	11.5 dB
Rao-Blackwellised Particle Smoother	14.7 dB

In addition to comparing the improvement in SNR, we compare the efficiency of the ordinary particle smoother and the Rao-Blackwellised particle smoother using a small section of speech data, "reward" as in Test 1. In this simulation, the number of particle, N, is set to be 1000 while the smoothing trajectories of the reflection coefficients are found by repeated application of different smoothers independently.

Figure 5 shows the smoothing trajectories of the reflection coefficient ρ_1. We assess the efficiency of the smoother by measuring the number of distinct trajectories - a smoother which generates very few distinct trajectories is degenerate and likely to give a very poor approximation to the true smoothing density. As shown in Figure 6, for the same number of filtered particles in each case, the proposed Rao-Blackwellised particle smoother consistently gives more distinct trajectories than the generic particle smoother. The Rao-Blackwellised particle smoother is thus likely to give a better approximation to the posterior distribution concerned. Of course, increasing the number of filtering particles improves the performance of both smoothers; we have chosen a value of $N = 1000$ in which the differences between the two smoothers are most clearly seen.

As a result of this and other simulations on audio data, we conclude that the Rao-Blackwellised particle smoother outperforms the generic particle smoother and the Extended Kalman filter / smoother quite consistently. It confirms experimentally the theory that by marginalising out some of the state variables, the estimation performance and the efficiency will improve.

C. Test 3 — TV-PARCOR model

As the extended Kalman filter is a computationally cheap algorithm, the Rao-Blackwellised particle smoother has to show consistent improvement over the extended Kalman filter / smoother, in order to justify using the suggested algorithm in practice. Therefore we re-run the test using two pieces of high quality music. Meanwhile, we include the Rao-Blackwellised (TVAR) particle smoother with random walk operating directly on the AR coefficients (4) in the test in order to verify experimentally our suggestion that the TV-PARCOR model is a better physical representation of audio signals than the TVAR model.

The first piece of music is a section of violin playing. In our simulations, the model order is assumed to be $p = 6$, and the fixed hyperparameters used are $\sigma_a^2 = 10^{-4}$, $\alpha = 0.995$ and $\sigma_{\phi_e}^2 = 10^{-6}$. For the Rao-Blackwellised (RB TVAR / RB TV-PARCOR) Particle Smoother, the block size is set to be 2000, $N = 100$ particles are used and $M = 10$ smoothed trajectories are generated from each block. The input SNR of the noisy data is 5.8 dB and the output SNR of different algorithms are summarised below:

Extended Kalman Smoother	6.9 dB
RB TVAR Particle Smoother	10.0 dB
RB TV-PARCOR Particle Smoother	12.1 dB

The second piece of music used is a section of brass. The parameter settings are the same as in the previous test. The input SNR of the noisy data is 10.4 dB and the output SNR of different algorithms are summarised below:

Extended Kalman Smoother	6.3 dB
RB TVAR Particle Smoother	16.8 dB
RB TV-PARCOR Particle Smoother	17.0 dB

We conclude that the Rao-Blackwellised particle smoother outperforms the extended Kalman filter very dramatically in terms of SNR for these extracts, giving a significant noise reduction when the extended Kalman filter effectively fails. In addition, the TV-PARCOR model outperforms the standard TVAR model, with the amount of improvement depending on the type of input material.

D. Test 4 — Different input SNR

In our final test, we investigate the performance of the Rao-Blackwellised particle smoother algorithm at different input SNR levels and compare with those generated using the extended Kalman filter / smoother. A section of piano music is used for this purpose.

As in the previous test, the model order is assumed to be $p = 6$, and the fixed hyperparameters used are $\sigma_a^2 = 10^{-4}$, $\alpha = 0.995$ and $\sigma_{\phi_e}^2 = 10^{-6}$. The output SNR using the different algorithms given noisy signals at different input SNR levels are summarised below:

SNR in	**Rao-Blackwellised smoother**	**extended Kalman smoother**
0 dB	8.5 dB	4.2 dB
10 dB	13.9 dB	13.8 dB
20 dB	20.9 dB	20.9 dB

It is found that the Rao-blackwellised particle smoother performs significantly better than the extended Kalman filter at low SNR, while both algorithms perform equally well at high SNR. This is as expected: the particle smoother takes proper account of the uncertainty in state estimates, which is especially important at low SNRs, but much less significant at high SNR.

VII. Conclusions

In this article, we have applied sequential Monte Carlo smoothing methods to audio signal enhancement problems. A TV-PARCOR model was proposed for modelling the time variation of the AR coefficients, which had the advantage that stability could be enforced easily and has a more reasonable physical interpretation in terms of acoustical tube models for voice and musical instruments.

In cases where the models concerned have some “tractable substructure”, Rao-Blackwellisation can be applied to reduce the estimation variance. A Rao-Blackwellised smoothing algorithm was developed in order to generate more reliable state estimates than in the generic particle smoother. In addition, owing to the lengthy nature of audio signals, the generic Monte Carlo smoother was adapted for block-based processing, so that storage and computation were reduced and the algorithm can be run in a pseudo-sequential, block-based mode.

Extensive tests have been carried out to investigate the effectiveness of the suggested algorithms applied to a variety of audio data. It is found that the Block-based Rao-Blackwellised (TV-PARCOR) particle smoother outperforms classical approaches such as the extended Kalman filter / smoother and also the generic particle smoother. Some representative examples of the test results are presented in this article, confirming the expected improvements in performance over both the extended Kalman filter / smoother and the generic particle smoother.

Appendix

I. Levinson Recursion

The Levinson recursions [24] define a non-linear invertible mapping between the AR, a_i, and reflection coefficients, ρ_i. The algorithms transforming the AR coefficients from the reflection coefficients, and vice versa are called the the step-up and step-down recursions respectively.

We define the $a_j(i)$ as the i^{th} element of an j^{th} order AR parameter. The step-up and step-down recursions proceed as follows,

Algorithm 6: Step-up Recursion

1. Initialise the recursion: $a_0(0) = 1$
2. For $j = 0, \ldots, p-1$

- For $i = 1, \ldots, j$, $a_{j+1}(i) = a_j(i) + \rho_{j+1} a_j(j-i+1)$
- $a_{j+1}(j+1) = \rho_{j+1}$

Algorithm 7: Step-down Recursion

1. Set $\rho_p = a_p(p)$
2. For $j = p-1, \ldots, 1$

- For $i = 1, \ldots, j$,

$$a_j(i) = \frac{1}{1-\rho_{j+1}^2}\left(a_{j+1}(i) - \rho_{j+1} a_{j+1}(j-i+1)\right)$$

- set $\rho_j = a_j(j)$

II. Kalman Filter

Given a conditional Gaussian state space model (equation 10) and assuming that the parameter $\theta_{1:t}$ is known, the Kalman filter equations are as follows.

Let $\zeta_{t|t-1} = \mathrm{E}(z_t|y_{1:t-1}, \theta_{1:t})$ denote the minimum variance estimate of z_t based on observations up to and including y_{t-1}, and $P_{t|t-1} = \mathrm{cov}(z_t - \zeta_{t|t-1}, z_t - \zeta_{t|t-1}|y_{1:t-1}, \theta_{1:t})$

denote the corresponding covariance matrix. From the Kalman filter,

$$\zeta_{t|t-1} = \mathcal{T}(\theta_t)\zeta_{t-1|t-1} \tag{16a}$$

$$P_{t|t-1} = \mathcal{T}(\theta_t)P_{t-1|t-1}\mathcal{T}(\theta_t)' + \mathcal{H}(\theta_t)\mathcal{Q}(\theta_t)\mathcal{H}(\theta_t)' \tag{16b}$$

with $\mathcal{Q}(\theta_t) = \mathrm{cov}(w_t, w_t)$. When y_t is available, the update equation will refine the estimate,

$$\zeta_{t|t} = \zeta_{t|t-1} + P_{t|t-1}\mathcal{Z}(\theta_t)'\mathcal{F}_t^{-1}(y_t - y_{t|t-1}) \tag{17a}$$

$$P_{t|t} = P_{t|t-1}(I - \mathcal{Z}(\theta_t)'\mathcal{F}_t^{-1}\mathcal{Z}(\theta_t)P_{t|t-1}) \tag{17b}$$

with $\mathcal{F}_t = \mathrm{cov}(y_t, y_t|y_{1:t-1}, \theta_{1:t})$ and $y_{t|t-1} = \mathrm{E}(y_t|y_{1:t-1}, \theta_{1:t})$ are given by the prediction error decomposition.

$$y_{t|t-1} = \mathcal{Z}(\theta_t)\zeta_{t|t-1} \tag{18a}$$

$$\mathcal{F}_t = \mathcal{Z}(\theta_t)P_{t|t-1}\mathcal{Z}(\theta_t)' + \mathcal{G}(\theta_t)\mathcal{Q}(\theta_t)\mathcal{G}(\theta_t)' \tag{18b}$$

References

[1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Englewood Cliffs, 1979.

[2] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*, Cambridge University Press, 1989.

[3] G. Kitagawa and W. Gersch, *Smoothness Priors Analysis of Time Series, Lectures Notes in Statistics #116*, New York: Springer-Verlag, 1996.

[4] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Tech. Rep. CUED/F-INFENG/TR310, Department of Engineering, Cambridge University, 1998.

[5] A. Doucet, N. de Freitas and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.

[6] A. Doucet, S. J. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[7] N. J. Gordon, D. J. Salmond and A. F. M. Smith, "Novel approach to nonlinear / non-Gaussian Bayesian state estimation," in *IEE Proceedings-F*, April 1993, vol. 140, pp. 107–113.

[8] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphics Statistics*, vol. 5, pp. 1–25, 1996.

[9] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[10] A. Doucet, S. J. Godsill and M. West, "Monte Carlo filtering and smoothing with application to time-varying spectral estimation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000, vol. 2, pp. 701–704.

[11] S. J. Godsill, A Doucet and M West, "Monte Carlo smoothing for non-linear time series," Tech. Rep. CUED/F-INFENG/TR, Department of Engineering, Cambridge University, 2001, Submitted for publication.

[12] S. J. Godsill and P. J. W. Rayner, *Digital audio restoration - A statistical model-based approach*, Berlin: Springer-Verlag, 1998.

[13] J. Makhoul, "Linear prediction: A tutorial review," in *Proceedings of the IEEE*, 1975, vol. 63, pp. 561–580.

[14] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, 1978.

[15] G. Kitagawa, "Changing spectrum estimation," *Journal of Sound and Vibration*, vol. 89, no. 4, pp. 443–445, 1983.

[16] R. Prado, M. West and A. Krystal, "Evaluation and comparation of eeg traces: Latent structure in non-stationary time series," *Journal of the American Statistical Association*, vol. 94, pp. 1083–1095, 1999.

[17] A. Dembo and O. Zeitouni, "Maximum a posteriori estimation of time-varying ARMA processes from noisy observations," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 4, pp. 471–476, 1988.

[18] Y. Grenier, "Time-dependent ARMA modeling of nonstationary signals," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 4, pp. 899–911, 1983.

[19] P. Ha and S. Ann, "Robust time-varying parametric modelling of voiced speech," *Signal Processing*, vol. 42, pp. 311–317, 1995.

[20] M. G. Hall, A. V. Oppenheim and A. S. Willsky, "Time-varying parametric modelling of speech," *Signal Processing*, vol. 5, pp. 267–285, 1983.

[21] J. Vermaak, C. Andrieu, A. Doucet and S. J. Godsill, "Non-stationary Bayesian modelling and enhancement of speech signals," Tech. Rep. CUED/F-INFENG/TR351, Department of Engineering, Cambridge University, 1999.

[22] C. A. Desoer and M. Vidyasagar, *Feedback systems: Input-output properties*, Academic Press, 1976.

[23] J. Vermaak, C. Andrieu, A. Doucet and S. J. Godsill, "On-line Bayesian modelling and enhancement of speech signals," Tech. Rep. CUED/F-INFENG/TR361, Department of Engineering, Cambridge University, 2000.

[24] B. Friedlander, "Lattice filters for adaptive processesing," *Proc.IEEE*, vol. 70, no. 8, pp. 829–867, August 1982.

[25] J. P. Proakis, J. R. Deller and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*, New York: Macmillan, 1993.

[26] J. E. Handschin and D. Q. Mayne, "Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering," *International Journal of Control*, vol. 9, no. 5, pp. 547–559, 1969.

[27] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxillary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[28] M. Hürzeler and H. R. Künsch, "Monte Carlo approximation for general state space models," *Journal of Computational and Graphics Statistics*, vol. 7, pp. 175–193, 2000.

[29] H. Tanizaki, "Nonlinear and non-Gaussian state space modeling using sampling techniques," *Annals of the Institute of Statistical Mathematics*, vol. 53, no. 1, pp. 63–81, 2001.

[30] A. Doucet, N. de Freitas, K. Murphy and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Stanford, California: Morgan Kaufmann, 2000.

[31] G. Casella and C. P. Robert, "Rao-Blackwellisation of sampling schemes," *Biometrika*, vol. 83, pp. 81–94, 1996.

[32] A. Kong, J. S. Liu and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–288, 1994.

[33] S. N. MacEachern, M. Clyde and J. S. Liu, "Sequential importance sampling for nonparametric Bayes models: the next generation," *Canadian Journal of Statistics*, vol. 27, pp. 251–267, 1999.

[34] C. Berzuini and W. Gilks, "Following a moving target - Monte Carlo inference of dynamic Bayesian models," *The Journal of Royal Statistical Society, Series B*, vol. 63, pp. 1–20, 2001.

[35] J. Vermaak and M. Niranjan, "Markov chain Monte Carlo methods for speech enhancement," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, 1998, vol. 2, pp. 1013–1016.

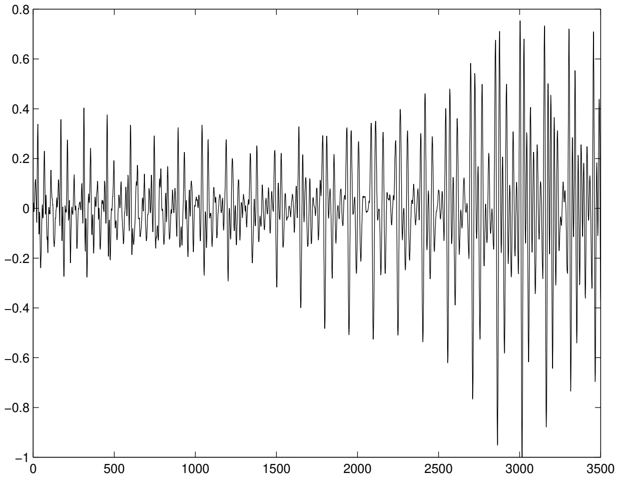

Fig. 1. Audio signal showing the word “reward”

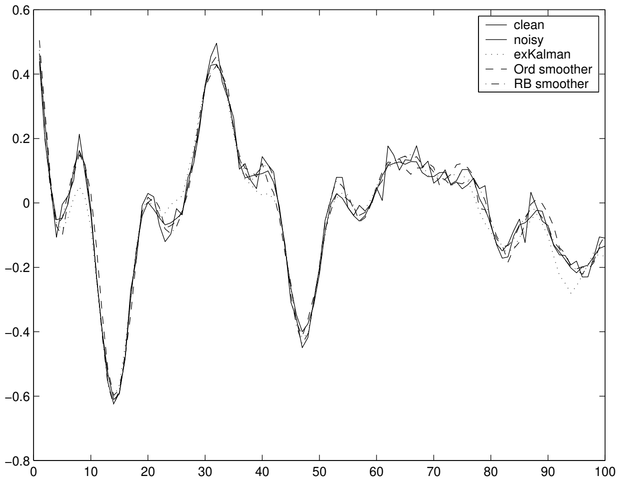

Fig. 4. A typical frame showing the clean, noise corrupted signals and reconstructed signal using different algorithms

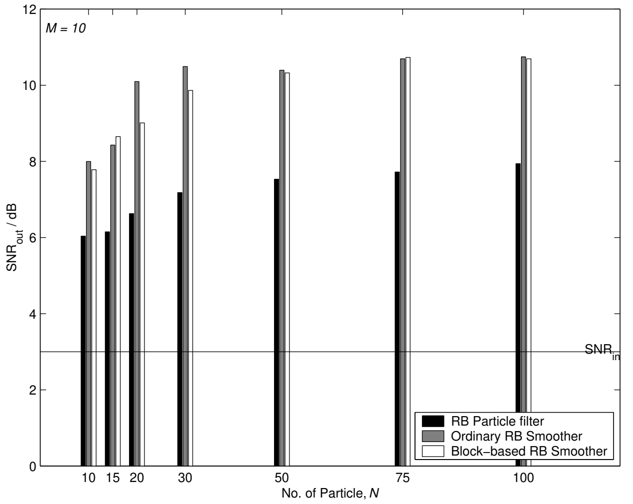

Fig. 2. For $M = 10$, output SNR for different N

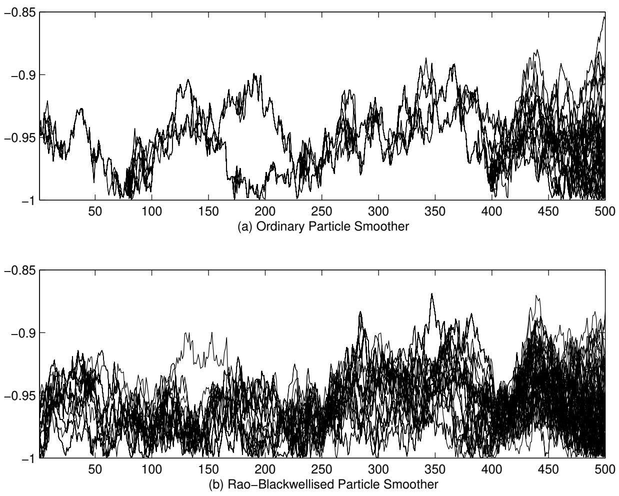

Fig. 5. For $N = 1000$ and $M = 100$, plots of smoothing trajectories of ρ_1 using different smoothers

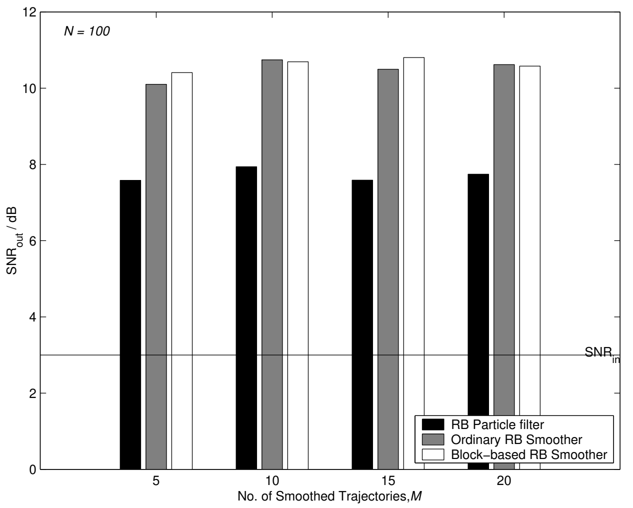

Fig. 3. For $N = 100$, output SNR for different M

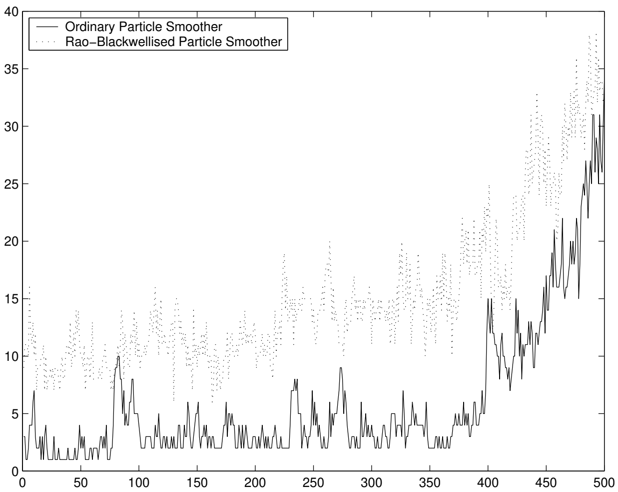

Fig. 6. For $N = 1000$ and $M = 100$, the variability of the smoothing trajectories are measured by number of distinct trajectories