

Notes on Testing and Extensions to the Rao-Blackwellised Particle Smoother

Pete Bunch

April 18, 2012

1 Comparison of RB Particle Smoothers

We have four RB smoothing algorithms for comparison: the filter-smoother (result of the final frame of the RB particle filter), the Kim's approximation smoother (which ignores the linear state contribution), the linear sampling smoother (of Fong et al.), and the full smoother (of Sarkka et al.). Previous tests have shown negligible difference between these.

1.1 The Model

The following model has been used for comparing the algorithms. It has been invented purely for this test and has limited physical meaning. A particle moves in one dimensions, and experiences a stochastic excitation of its $(d-1)$ th derivative. All the derivatives are subject to a mean reversion effect to ensure stability.

$$dz^{(d-1)}(t) = -\lambda_{d-1}z^{(d-1)}(t)dt + \sigma d\beta(t) \quad (1)$$

$$dz^{(n)}(t) = (z^{(n+1)} - \lambda_n z^{(n)})dt \quad 0 \leq n < d-1 \quad (2)$$

Stack this lot up into a matrix equation.

$$x(t) = \begin{bmatrix} z(t) \\ z'(t) \\ \vdots \\ z^{(d-1)} \end{bmatrix} \quad (3)$$

$$dx(t) = \begin{bmatrix} -\lambda_0 & 1 & \dots & 0 & 0 \\ 0 & -\lambda_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -\lambda_{d-2} & 1 \\ 0 & 0 & \dots & 0 & -\lambda_{d-1} \end{bmatrix} x(t)dt + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \sigma \end{bmatrix} d\beta(t) \quad (4)$$

This is easily discretised by matrix fraction decomposition.

$$x_k = Ax_{k-1} + w_k \quad (5)$$

w_k is a Gaussian random variable with covariance $Q_n = \sigma^2 Q$ where Q is constant (for fixed size time step).

Now to make it only conditionally linear-Gaussian. We'll make σ^2 variable over time. The “nonlinear” part of the state will be modelled as a first order AR process driven by Gaussian noise.

$$u_k = \alpha u_{k-1} + \epsilon_k \quad (6)$$

ϵ_k is a Gaussian random variable with covariance Ψ (constant for fixed size time step).

Now make $\sigma_k^2 = u_k^2$. Finally, we make linear-Gaussian observations of the z_k only.

$$y_k = C z_k + v_k \quad (7)$$

$$C = [1 \quad 0 \quad \dots \quad 0] \quad (8)$$

$$v_k \sim \mathcal{N}(v_k | 0, R) \quad (9)$$

To summarise, our dynamics are specified by the following equations.

$$u_k = \alpha u_{k-1} + \epsilon_k \quad (10)$$

$$\epsilon_k \sim \mathcal{N}(\epsilon_k | 0, \Psi) \quad (11)$$

$$\sigma_k^2 = u_k^2 \quad (12)$$

$$x_k = A x_{k-1} + w_k \quad (13)$$

$$w_k \sim \mathcal{N}(w_k | 0, \sigma_k^2 Q) \quad (14)$$

$$y_k = C x_k + v_k \quad (15)$$

$$v_k \sim \mathcal{N}(v_k | 0, R) \quad (16)$$

This nicely fits the conditionally linear-Gaussian framework required for RB filtering and smoothing. It is also easy to vary the number of state dimensions, d to test the effect on algorithm performance.

2 Testing

Here are some results, using 100/10 filter particle and 100/10 smoother trajectories on 5000 time steps, with 1 – 10 dimensions.

It looks like we can conclude that the Kim's approximation smoother deteriorates with the dimension of the linear state, and that all of them outperform the filter-smoother (i.e. Kitagawa smoother). However, there's no significant difference between the linear-sampling (Fong et al.) and full (Sarkka et al.) smoothers on this test.

Additionally, a second model has been tried in which there is an independent driving Brownian motion on every derivative, all with the same standard deviation, σ_k , so the dimension of w_k is also d . RMSE results closely resemble those above with a single Brownian motion.

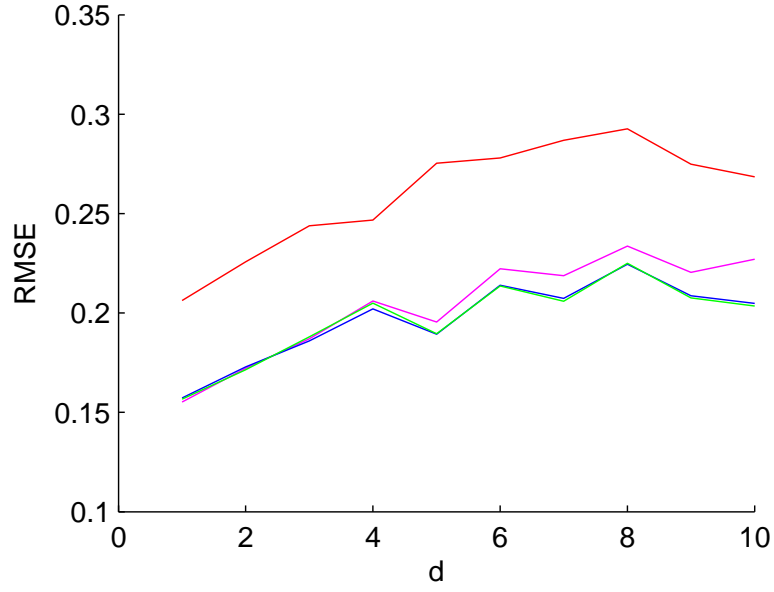


Figure 1: RMSEs for filter-smoother (red), Kim's approximation smoother (pink), linear-sampling smoother (blue), full smoother (green) using 100 filter particle and 100 smoother trajectories.

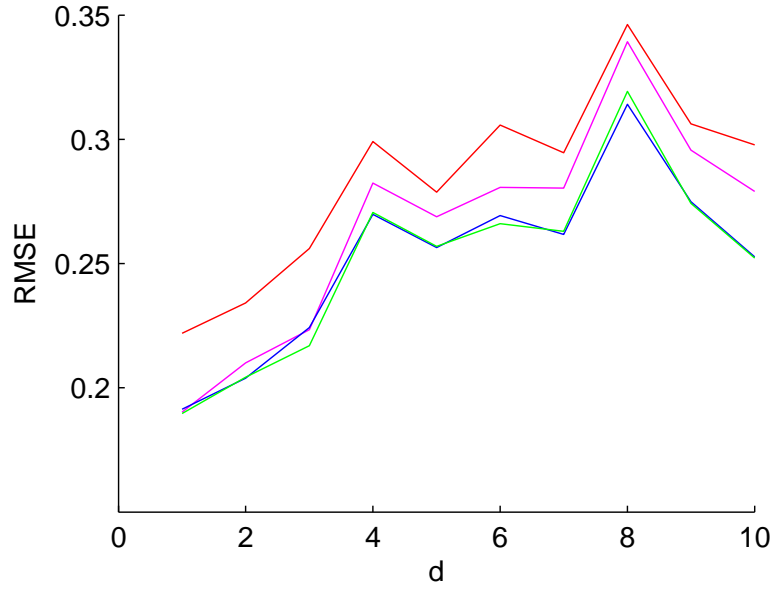


Figure 2: RMSEs for filter-smoother (red), Kim's approximation smoother (pink), linear-sampling smoother (blue), full smoother (green) using 10 filter particle and 10 smoother trajectories.

3 Other Testing

Tests have also been conducted using a different model, in which a number of Brownian motion or OrnsteinUhlenbeck processes are run in parallel, conditional upon a common, time-varying diffusion constant (the “nonlinear” part of the state) which evolves according to an AR process. Testing using a wide variety of model and algorithm parameters yielded no combination for which the full smoother outperformed the linear-sampling smoother.

4 MCMC smoothing

The new MCMC smoothing algorithms can be applied to conditionally linear-Gaussian models.

At the moment, we’re sampling the conditional distribution (or an approximation thereof), $p(u_{1:k}|u_{k+1:K}, y_{1:K})$, directly. This means that we have to calculate a sampling weight for each filtering particle at every time step for every smoothing trajectory, so the complexity is $\mathcal{O}(N_F \times N_S \times K)$. We could bring this down using MCMC with a short chain, instead of direct sampling. Briefly, a new nonlinear sequence, $u_{1:k}^*$, is proposed from the particle approximation of $p(u_{1:k}|y_{1:k})$ which we already have from the filter. This is accepted with probability,

$$\alpha = \left[1, \frac{p(u_{1:k}^*|u_{k+1:K}, y_{1:K})p(u_{1:k}|y_{1:k})}{p(u_{1:k}|u_{k+1:K}, y_{1:K})p(u_{1:k}^*|y_{1:k})} \right] \quad (17)$$

$$= \left[1, \frac{\int p(y_{k+1:K}|x_k, u_{1:k}^*, u_{k+1:K})p(x_k|u_{1:k}^*, y_{1:k})dx_k p(u_{k+1}|u_k^*)}{\int p(y_{k+1:K}|x_k, u_{1:k}, u_{k+1:K})p(x_k|u_{1:k}, y_{1:k})dx_k p(u_{k+1}|u_k)} \right]. \quad (18)$$

Using this strategy, we only calculate “weights” of the particles we actually visit. Accuracy approaches that of the direct sampler as the length of the chain, M , increases. As long as $M < N_F$, then this is quicker than direct sampling.

We can extend this framework to construct a smoother which is not limited to the support of the filtering approximation. We can expand the target distribution as,

$$p(u_{1:k}|u_{k+1:K}, y_{1:K}) \propto \int p(y_{k:K}|x_k, u_{1:k}, u_{k+1:K})p(x_k|u_{1:k}, y_{1:k-1})dx_k \\ \times p(u_{k+1}|u_k)p(u_k|u_{k-1})p(u_{1:k-1}|y_{1:k-1}). \quad (19)$$

Now we can change the MCMC proposal to $p(u_{1:k-1}|y_{1:k-1})q(u_k|u_{k-1}, u_{k+1}, y_{1:K})$. The corresponding acceptance probability is,

$$\alpha = \left[1, \frac{p(u_{1:k}^*|u_{k+1:K}, y_{1:K})p(u_{1:k-1}|y_{1:k-1})q(u_k|u_{k-1}, u_{k+1}, y_{1:K})}{p(u_{1:k}|u_{k+1:K}, y_{1:K})p(u_{1:k-1}^*|y_{1:k-1})q(u_k^*|u_{k-1}, u_{k+1}, y_{1:K})} \right] \quad (20)$$

$$= \left[1, \frac{\int p(y_{k:K}|x_k, u_{1:k}^*, u_{k+1:K})p(x_k|u_{1:k}^*, y_{1:k-1})dx_k}{\int p(y_{k:K}|x_k, u_{1:k}, u_{k+1:K})p(x_k|u_{1:k}, y_{1:k-1})dx_k} \frac{p(u_{k+1}|u_k^*)p(u_k^*|u_{k-1})q(u_k|u_{k-1}, u_{k+1}, y_{1:K})}{p(u_{k+1}|u_k)p(u_k|u_{k-1})q(u_k^*|u_{k-1}, u_{k+1}, y_{1:K})} \right]. \quad (21)$$

The difficulty here would be devising good choices for $q(u_k|u_{k-1}, u_{k+1}, y_{1:K})$, which will be significantly more complex than the standard (i.e. not RB) case.