Monte Carlo approximations for general state space models

by

Markus Hürzeler and
Hans R. Künsch

Research Report No. 73
April 1995

Seminar für Statistik

Eidgenössische Technische Hochschule (ETH)

CH-8092 Zürich

Switzerland

Monte Carlo approximations for general state space models

Markus Hürzeler and
Hans R. Künsch

Seminar für Statistik
ETH Zentrum
CH-8092 Zürich, Switzerland

April 1995

Abstract

Non-linear and non-Gaussian state space models form a very large and flexible model class in time series analysis. Two methods for generating iteratively samples from filter densities and smoother densities by simple rejection algorithms are introduced. We illustrate the behavior of our methods in several non-linear and non-Gaussian examples and compare them with other well-known methods.

Keywords: State space models, Kalman filter, Kalman smoother, Monte Carlo methods, Robustness for time series, Non-linear time series analysis.

1 Introduction

Nonlinear and nonGaussian state space models are very flexible and suitable for a variety of time series occurring in practice, see West, Harrison and Migon (1986), Kitagawa (1987, 1991), Carlin, Polson and Stoffer (1992). But in order to make use of this potential and to apply these models to data, one has to solve the hard problem of performing the necessary computations. Although the relevant formulae can be written down easily, they cannot be calculated in closed form except in a few special situations. So one has to rely on approximations. Basically we can distinguish three kinds of approximations: analytical, numerical and Monte Carlo approximations. The standard analytical approximation is the well known extended Kalman filter. More recent proposals are for instance the procedure of Harrison and Stevens (1976) for the multiprocess model and Schick and Mitter (1994) for contaminated mormal errors. Typically the error in analytical approximations is not known and reducing it by using higher order approximations is prohibitive. Numerical approximations started with Kitagawa (1987). They are extremely accurate, but their complexity makes them not feasible in higher dimensions. Monte Carlo approximations have been proposed by Carlin et al. (1992), Kitagawa (1993) and Tanizaki (1993), and also our approach falls in this category.

Carlin et al. (1992) use Markov chain Monte Carlo which is quite different from what we propose. Since the Markov chain Monte Carlo method does not make use of the special structure of the state space model, recursive computation of the filter distribution is not possible. Moreover parameters of the model have to be treated as hyper-parameters in a Bayesian framework.

The closest approaches to ours are the simulation based density estimator of Tanizaki (1993) and the method of Kitagawa (1993). Tanizaki relies on an analytical approximation to the filter densities and then computes recursively importance weights. Kitagawa (1993) generates recursively samples from the prediction and the filter distributions, and so do we. But our method which has been developed independently avoids a problem of discreteness inherent in Kitagawa's (1993) approach because his filter sample is a subset of the prediction sample. This creates problems when the filter density is much larger than the prediction density in some region, see Fig. 1, (i) and (j) of Kitagawa(1993) where the filter distribution is not well approximated in the right tail. This is because his filter sample is a subset of the prediction sample whereas ours is not. In fact, one of our algorithms does not even need a prediction sample. In our approach the filter sample is constructed from an absolutely continuous distribution. So we need fewer replicates for our sample. This is an advantage even with todays computer hardware because one has to store all filter samples if one wants to compute later on the smoothing distributions or likelihood ratios.

Our paper is organized as follows: In Section 2 we describe our model and the formulae for prediction filtering and smoothing. Section 3 presents two variants for constructing the filter samples. Section 4 deals with smoothing, and in Section 5 we discuss problems occurring when state transitions are not absolutely continuous. Finally Section 6 presents 4 examples.

2 The general state space model and the Kalman filter

We assume that the observed time series (Y_t) derives from an unobservable k-dimensional state process (X_t). The joint distribution of the observation and the state process is specified as follows:

(M1) $X_0, X_1, X_2, \ldots$ is a Markov chain with $X_0 \sim p_0(x)dx$ and

$$X_t | X_{t-1} = x_{t-1} \quad \sim \quad p_t(x|x_{t-1})dx.$$

(M2) The Y_t's are conditionally independent given $X_0, X_1, X_2, \ldots$ and each Y_t is conditionally independent of $X_s, s \neq t$ given X_t with

$$Y_t | X_t = x_t \quad \sim \quad q_t(y|x_t)dy.$$

For simplicity we assume Y_t to be real valued although this can easily be generalized. Frequently this model takes the special form

$$X_t = g(X_{t-1}) + V_t \tag{2.1}$$

$$Y_t = h(X_t) + W_t \tag{2.2}$$

where $g : \mathbb{R}^k \to \mathbb{R}^k$ and $h : \mathbb{R}^k \to \mathbb{R}$ are given functions and (V_t), (W_t) are two independent i.i.d. sequences with

$$V_t \sim p(v)dv, \quad W_t \sim q(w)dw. \tag{2.3}$$

If X_0 is independent of $(V_t)_{t>0}, (W_t)_{t>0}$ (M1) and (M2) are satisfied with

$$p_t(x|x_{t-1}) = p(x - g(x_{t-1})) \tag{2.4}$$

$$q_t(y|x_t) = q(y - f(x_t)). \tag{2.5}$$

In Section 5 we will briefly discuss the case where the transition probabilities for (X_t) are not absolutely continuous.

Let Y_1^s be the observations until time s, i.e. $Y_1^s = (Y_1, \dots, Y_s)$ and denote the conditional density of X_t given Y_1^s by $f_{t|s}(x|y_1^s)$. Depending on whether $t > s$, $t = s$, $t < s$ $f_{t|s}$ is called the prediction, filter and smoother density respectively.

In principle we can obtain $f_{t|t}$ and $f_{t|t-1}$ according to the following general Kalman recursions:

$$f_{t|t-1}(x|y_1^{t-1}) = \int p_t(x|x_{t-1}) f_{t-1|t-1}(x_{t-1}|y_1^{t-1}) dx_{t-1} \tag{2.6}$$

$$f_{t|t}(x|y_1^t) = \frac{q_t(y_t|x) f_{t|t-1}(x|y_1^{t-1})}{\int q_t(y_t|x_t) f_{t|t-1}(x_t|y_1^{t-1}) dx_t} \tag{2.7}$$

The recursion is started with $f_{0|0}(x) = p_0(x)$. In practice exact computations are not possible except when (2.1) – (2.2) hold, g and h are linear and V_t, W_t and X_0 are all Gaussian. Then all $f_{t|s}$ are also Gaussian and their moments are obtained from the usual Kalman filter. In practically all other cases we have to rely on some approximations.

Since we are also going to propose approximations for the smoother densities $f_{t|T}(t \le T)$, we briefly recall here the relevant formulae. It is well known that conditional on Y_1^T $(X_1, X_2, \dots, X_T)$ is a time inhomogeneous Markov chain backward in time. Its distribution is given as follows

$$\begin{aligned} X_T|Y_1^T &\sim f_{T|T}(x|y_1^T)dx, \\ X_t|X_{t+1}, Y_1^T &\sim \frac{p_{t+1}(x_{t+1}|x) f_{t|t}(x|y_1^t)}{f_{t+1|t}(x_{t+1}|y_1^t)} dx. \end{aligned}$$

Hence we have

$$f_{t|T}(x|y_1^T) = \int \frac{p_{t+1}(x_{t+1}|x) f_{t|t}(x|y_1^t)}{f_{t+1|t}(x_{t+1}|y_1^t)} f_{t+1|T}(x_{t+1}|y_1^T) dx_{t+1}. \tag{2.8}$$

3 Recursive sampling according to the filter density

Our aim is to generate recursively a sample $(x_{t|t}^{(j)})_{j=1,\dots,N}$ distributed according to $f_{t|t}(x|y_1^t)dx$. (We take the observations as fixed and suppress the dependence of $x_{t|t}^{(j)}$ on y_1^t). With the help of such a sample we can then approximate moments and quantiles by using averages instead of integrals. Two different ways of constructing these samples will be discussed.

3.1 The direct method

Assume that we have constructed already the sample at time $t-1$. Then (2.6) and (2.7) suggest the following approximations of $f_{t|t-1}$ and $f_{t|t}$:

$$\hat{f}_{t|t-1}(x|y_1^{t-1}) = \frac{1}{N}\sum_{j=1}^{N} p_t(x|x_{t-1|t-1}^{(j)}) \tag{3.1}$$

$$\hat{f}_{t|t}(x|y_1^{t}) \propto \sum_{j=1}^{N} q_t(y_t|x)p_t(x|x_{t-1|t-1}^{(j)}) \tag{3.2}$$

The direct method proceeds by generating $(x_{t|t}^{(j)})_{j=1,\dots,N}$ according to (3.2). This can be done by the following rejection algorithm. Assume that c_t is an upper bound of $q_t(y_t|.)$:

$$c_t \geq \sup_x q_t(y_t|x)$$

1. Put $j=1$.
2. If $j \leq N$, generate I uniform on $\{1,\dots,N\}$, otherwise stop.
3. Generate $X \sim p_t(x|x_{t-1|t-1}^{(I)})dx$ and U uniform on $[0, c_t]$.
4. If $q_t(y_t|X) \geq U$, put $x_{t|t}^{(j)} = X$ and $j = j+1$.
5. Return to 2.

It is easily verified that this leads to the required distribution. The only requirements are that $q_t(y_t|.)$ is bounded and that we can simulate easily according to $p_t(x|x_{t-1})dx$ for any x_{t-1}. These conditions are usually satisfied.

3.2 The kernel method

Here we proceed in two steps. First we generate a prediction sample according to the approximate prediction density (3.1). Denote this sample by $(x_{t|t-1}^{(j)})_{j=1,\dots,N}$. Then we estimate $f_{t|t-1}$ by a kernel density estimator and plug the result in formula (2.7) for $f_{t|t}$. This means that we generate $(x_{t|t}^{(j)})_{j=1,\dots,N}$ according to

$$\hat{f}_{t|t}(x|y_1^{t}) \propto \sum_{j=1}^{N} q_t(y_t|x)K(\sigma^{-1}A_t(x - x_{t|t-1}^{(j)})). \tag{3.3}$$

Here A_t is a square root of the inverse covariance matrix of $(x_{t|t-1}^{(j)})_{j=1,\ldots,N}$:

$$(A_t' A_t)^{-1} = (N-1)^{-1} \sum_j (x_{t|t-1}^{(j)} - \overline{x}_{t|t-1})(x_{t|t-1}^{(j)} - \overline{x}_{t|t-1})',$$

σ is a bandwidth and K is a radially symmetric density in $\mathbb{R}^k$ with support in $\{x| \, ||x|| \leq 1\}$.

Formally, simulating from (3.3) looks the same as simulating from (3.2). However the compact support of the terms on the right hand side of (3.3) allows us to speed up the previous algorithm. Assume that we have coefficients $c_{t,j}$ satisfying

$$c_{t,j} \geq \sup\{q_t(y_t|x)| \, ||A_t(x - x_{t|t-1}^{(j)})|| \leq \sigma\}.$$

Then the following algorithm generates the filter sample

1. Put $j = 1$.
2. If $j \leq N$, generate I according to $P[I = i] \propto c_{t,i}$, otherwise stop.
3. Generate $Z \sim K(x)dx$ and U uniform on $[0, c_{t,I}]$. Compute $X = x_{t|t-1}^{(I)} + \sigma A_t^{-1} Z$.
4. If $q_t(y_t|X) \geq U$, put $x_{t|t}^{(j)} = X$ and $j = j + 1$.
5. Return to 2.

It is easy to see that this algorithm does generate a sample according to (3.3) and that it leads on average to fewer rejections when not all $c_{t,j}$ are equal. However for a practical procedure we have to be able to choose the bandwidth and to compute bounds $c_{t,j}$. For the choice of the bandwidth we propose the following method. If $f_{t|t-1}$ were a Gaussian density, then the optimal bandwidth with respect to asymptotic mean integrated square error would be

$$\sigma = A(K) N^{-1/(k+4)} \tag{3.4}$$

where $A(K)$ is given in formula (4.15) of Silverman (1986). However the interesting and crucial situations occur when $f_{t|t-1}$ is multimodal. The discussion in Section 3.4.2 of Silverman (1986) suggests to choose in this case a somewhat smaller value than (3.4). In our implementation we therefore used

$$\sigma = A(K) N^{-1/(k+4)}/2.5. \tag{3.5}$$

Whether we are able to compute good bounds $c_{t,j}$ depends of course on the form of q and thus on the model considered. A closed form for $c_{t,j}$ can be given if Y_t derives from X_t according to (2.2) with h linear:

$$Y_t = h' X_t + W_t \quad (h \in \mathbb{R}^k).$$

Assume in addition that the density of the observation noise W_t is increasing on $(-\infty, 0]$ and decreasing on $[0, \infty)$. If we put

$$\alpha_{t,j}^{\pm} = h' x_{t|t-1}^{(j)} \pm \sigma (h'(A_t' A_t)^{-1} h)^{1/2},$$

then it is easily checked that

$$c_{t,j} = \begin{cases} q(\alpha^+_{t,j}) & \text{if } \alpha^+_{t,j} < 0 \\ q(\alpha^-_{t,j}) & \text{if } \alpha^-_{t,j} > 0 \\ q(0) & \text{otherwise.} \end{cases}$$

So in this case we can compute the best possible bounds in closed form.

Finally we should briefly mention the role of the dimension k. It is known, that in high dimensions, say $k \geq 4$, kernel density estimation is virtually impossible. However our aim is not to estimate the density by itself, but to sample from the density (3.3). In particular even the bandwidth $\sigma = 0$ is not totally unreasonable, and some smoothing of the empirical distribution should improve the results. So we expect that our procedure does not suffer too much from the curse of dimension.

4 Smoothing

We assume that we have stored the filter samples $(x^{(j)}_{t|t})_{j=1,\ldots,N}$ for $t = 1, \ldots, T$. We now proceed to obtain approximations of (2.8) using these filter samples. As for constructing the filter samples there are two ways to do it. Both ways replace the integral in (2.8) by an average and estimate $f_{t|t}$. This gives

$$\hat{f}_{t|T}(x|y_1^T) = \frac{1}{N} \sum_{j=1}^{N} \frac{p_{t+1}(x^{(j)}_{t+1|T}|x)\hat{f}_{t|t}(x|y_1^t)}{\beta_j} \tag{4.1}$$

where β_j is a normalizing constant so that each summand is a probability density. The difference between the two methods lies in $\hat{f}_{t|t}$. The direct method uses equation (3.2). We assume that we have bounds

$$c_{t,j} \geq \sup_x q_t(y_t|x)p_{t+1}(x^{(j)}_{t+1|T}|x).$$

Then we propose the following algorithm:

1. Put $j = 1$.
2. If $j \leq N$, generate I uniform on $\{1, \ldots, N\}$, otherwise stop.
3. Generate X according to $p_t(x|x^{(I)}_{t-1|t-1})dx$ and U uniform on $[0, c_{t,j}]$.
4. If $q_t(y_t|X)p_{t+1}(x^{(j)}_{t+1|T}|X) \geq U$, put $x^{(j)}_{t|T} = X$ and $j = j + 1$.
5. Return to 2.

This algorithm does not produce an i.i.d. sample from (4.1), but rather one observation from each of the mixture densities in (4.1). Still we can estimate expectations w.r. to $f_{t|T}$ by the corresponding means over the sample.

The kernel method uses instead of (3.2)

$$\hat{f}_{t|t}(x|y_1^t) = \det(A_t)/(\sigma N)\sum_{i=1}^{N} K(\sigma^{-1}A_t(x - x_{t|t}^{(i)})) \tag{4.2}$$

where A_t is a square root of the inverse of the empirical covariance matrix of $(x_{t|t}^{(i)})_{i=1,\ldots,N}$ and σ is a bandwidth. We choose σ as in the filter algorithm. Note however that A_t is now a different quantity. For generating $x_{t|T}^{(j)}$ we assume that we have bounds

$$c_{t,ij} \geq \sup_x \{p_{t+1}(x_{t+1|T}^{(j)}|x)| \; ||A_t(x - x_{t|t}^{(i)})|| \leq \sigma\}.$$

Then the following algorithm generates the smoothing sample (again one observation for each density in the mixture (4.1)).

1. Put $j = 1$.
2. If $j \leq N$, generate I according to $P[I = i] \propto c_{t,ij}$, otherwise stop.
3. Generate $Z \sim K(x)dx$ and U uniform on $[0, c_{t,Ij}]$. Compute $X = x_{t|t}^{(I)} + \sigma A_t^{-1} Z$.
4. If $p_{t+1}(x_{t+1|T}^{(j)}|X) \geq U$, put $x_{t|T}^{(j)} = x$ and $j = j + 1$.
5. Return to 2.

There is however an approximate smoothing algorithm which does not require any sampling. Namely if $f_{t|t}(x|y_1^t)$ approaches the discrete distribution giving each $x_{t|t}^{(i)}$ probablility N^{-1} (i.e. we let σ tend to zero in (4.2)), then in the limit the smoother sample is concentrated on $(x_{t|t}^{(i)})_{i=1,\ldots,N}$, with the following probabilities:

$$P[X_t = x_{t|t}^{(i)}|Y_1^T] = \sum_{j=1}^{N} P[X_{t+1} = x_{t+1|T}^{(j)}|Y_1^T]\pi(j,i) \tag{4.3}$$

where

$$\pi(j,i) = p_{t+1}(x_{t+1|T}^{(j)}|x_{t|T}^{(i)})/\sum_{\ell} p_{t+1}(x_{t+1|T}^{(j)}|x_{t|T}^{(\ell)}).$$

Hence we can compute $P[X_t = x_{t|t}^{(i)}|Y_1^T]$ recursively for $t = T,\ T-1, \ldots, 1$, starting with uniform probabilities N^{-1}.

Obviously if both N and T are large all these algorithms have serious storage problems. A straightforward remedy is to be satisfied with $f_{t|t+L}$ for L fixed. Then we have to store only $L+1$ (or $L+2$) filter samples. However this saving is at least partially outweighted by additional computations. For each t we compute $f_{t|t+1}, f_{t|t+2}, \ldots, f_{t|t+L}$. Another approach is systematic thinning of the filter samples. If $k = 1$ we might for instance retain only every m-th order statistic. For $k > 1$ it seems more difficult to do such a balanced thinning.

5 State transitions which are not absolutely continuous

In many applications with $k > 1$, transition densities for (X_t) do not exist because certain components of X_t are deterministically determined by X_{t-1}. For the kernel method this does not matter. Typically simulation from the conditional distribution of X_t given X_{t-1} is straightforward, and this is all we need.

For the direct method however some problems occur. If we use the analogue of (3.2) we obtain a distribution which is discrete in some components and absolutely continuous in the others. The reason for using the mixture (3.1) instead of the empirical distribution of a sample from $f_{t|t-1}$ was precisely to avoid discrete distributions. We can of course nevertheless use the analogue of (3.2). Since only some components will be discrete we expect that the effects of this discretenes will be less visible.

In many cases transition densities do not exist because we have to enlarge the state vector in order to obtain a first order Markovian behavior. In these cases we can avoid the discreteness completely by updating with several observations at the time. To state this precisely we denote $(X_s, X_{s+1}, \ldots, X_t)$ by X_s^t and replace the model assumption (M1) by

(M1') $X_{1-m}, X_{2-m}, \ldots,$ is an m-th order Markov chain with

$$X_{1-m}^0 \quad \sim \quad p_0(x_{1-m}^0)dx_{1-m}\ldots dx_0$$

and

$$X_t | X_{t-m}^{t-1} = x_{t-m}^{t-1} \quad \sim \quad p_t(x|x_{t-m}^{t-1})dx.$$

(M2) remains unchanged. We denote the conditional density of X_{t-m+1}^t given Y_1^s by $f_{t|s}$.

Then the recursions (2.6) and (2.7) are replaced by

$$\begin{aligned} f_{t|t-m}(x_{t-m+1}^t | y_1^{t-m}) &= \int \prod_{j=1}^m p_{t-m+j}(x_{t-m+j} | x_{t-2m+j}^{t-m+j-1}) \\ &\quad f_{t-m|t-m}(x_{t-2m+1}^{t-m} | y_1^{t-m}) dx_{t-2m+1} \ldots dx_{t-m}, \qquad (5.1) \\ f_{t|t}(x_{t-m+1}^t | y_1^t) &\propto \prod_{j=1}^m q_{t-m+j}(y_{t-m+j} | x_{t-m+j}) f_{t|t-m}(x_{t-m+1}^t | y_1^{t-m}). \qquad (5.2) \end{aligned}$$

Based on these recursions we can immediately generalize the direct method of Section 3.1 to the more general model (M1'), (M2). Details are left to the reader.

6 Examples

In this section we illustrate the performance of our methods in some linear and nonlinear state space models. We also show in two simulation studies the influence of the sample size and the random effects of the methods and the data.

6.1 Shifted-mean model

As our first example we consider the shifted-mean model from Kitagawa (1987). To a piecewise constant mean Gaussian white noise is added as shown in Figure 1. In order to estimate the unknown mean, i.e. to smooth the observations, the following state space model is used:

$$\begin{aligned} x_t &= x_{t-1} + v_t \\ y_t &= x_t + w_t \end{aligned}$$

where v_t is distributed with density $p(v) \propto (2.2 \cdot 10^{-7} + v^2)^{-0.75}$ and $w_t \sim N(0, 1.022)$. These parameters are the same as in Kitagawa (1987). The 10%-, 50%- and 90%-quantiles of the filter densities in Figure 2 are obtained with the direct method. For comparison the median of the filter density obtained by the numerical integration method is also drawn. For better visibility only a part of the series near a jump is shown. It can be seen that our direct method gives nearly the same results as the numerical integration method which is considered to be practically without error.

Figure 3 shows the results of the direct smoother. The shift in the state is detected quite well, and again our results are very close to the result from numerical integration. Note that we used a reasonable Monte Carlo sample size $N = 1000$.

6.2 Comparison of the Monte Carlo errors

We consider here the first 150 time points of the shifted mean model. This subset includes one shift at $t = 100$. The data are the same as in Figure 1. We then computed m replicates of the filter and smoother samples with different sample sizes N. For the filter m was 200 and for the smoother 50.

Since the numerical integration method is very precise, the error of the i-th replicate is

$$\mathrm{e}_{t|t;i}(N) = \hat{x}_{t|t;i} - \tilde{x}_{t|t} \tag{6.1}$$

where $\tilde{x}_{t|t}$ denotes the median of the numerical integration method, N the sample size of our Monte Carlo algorithm and $\hat{x}_{t|t;i}$ the filter median obtained by our method in the ith run.

Figure 4 gives box plots of $\mathrm{e}_{t|t;i}(N)$ for the direct, the kernel and the stochastic algorithm by Kitagawa (1993) for different sample sizes N. The bars at the righthand side indicate a unit length. As expected the spread of the errors decreases with increasing sample size N. The kernel filter and Kitagawa's stochastic filter behave quite similarly whereas the direct filter performs better, both with respect to bias and variability. The bias and a bigger variability of the errors occurs from $t = 104$ on. At this time-point the median of the numerical integration method detects the jump. Also the stochastic method detects the jump at $t = 104$, but the different new mean levels lead to the enlarged variability.

In analogy the errors $\mathrm{e}_{t|T;i}(N)$ with $T = 150$ for the direct and kernel smoother were calculated and are shown in Figure 5. For the smoother methods a small bias occurs at the time of the jump ($t = 101$), whereas in the other time-points the variability of the errors are small and around zero.

6.3 Nonlinear model

We consider the following nonlinear state and observation equation

$$
\begin{aligned}
x_t &= \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + v_t \\
y_t &= \frac{x_t^2}{20} + w_t
\end{aligned}
$$

with $v_t \sim N(0,10)$ and $w_t \sim N(0,1)$ independent of each other. This model was originally introduced by Andrade Netto et. al (1978) and is mentioned in the rejoinder of Kitagawa (1987).

A simulation of the state and observation variables is shown in Figure 6. We applied the kernel filter and smoother to this data. The estimated 10%-, 50%- and 90%-quantiles are shown in Figure 7 for the filter and in Figure 8 for the smoother. Due to the quadratic observation equation some of the filter densities are bimodal as can be seen in Figure 9. This shows well how the smoother $\hat{x}_{t|T}$ corrects the uncertainty of the filter $\hat{x}_{t|t}$. Moreover this quite complex model illustrates that we obtain good results with a reasonable sample size $N = 1000$.

6.4 AR-models contaminated by observation noise

In this example we study the robust performance of a non-Gaussian Kalman filter computed by our method for several observation noises. The outline of the study is the same as in Schick and Mitter (1994). The data are simulated as follows

$$
\begin{aligned}
x_t &= \alpha x_{t-1} + v_t \\
y_t &= x_t + w_t
\end{aligned}
$$

with $\alpha = 0.1$ and 0.5 respectively and $v_t \sim N(0,1)$. The observation noise w_t is $N(0,1)$-distributed for $t \neq 20$. At $t = 20$ there are 5 different distributions: $N(0,1)$ (no contamination), $N(0,9)$, Laplace, Cauchy and Slash. For each simulation we had a time series length T=50 and 200 replications of the data.

For all datasets generated in this way we compute the filter mean based on the model

$$
\begin{aligned}
x_t &= \alpha x_{t-1} + v_t \\
y_t &= x_t + w_t
\end{aligned}
$$

with $\alpha = 0.1$ and 0.5 respectively, $v_t \sim N(0,1)$ and $w_t \sim q(w)dw$ where $q(w)$ is Huber's density

$$
q(w) = \begin{cases} (1-\epsilon)/\sqrt{2\pi}\exp(-w^2/2) & |w| \leq k = k(\epsilon) \\ (1-\epsilon)/\sqrt{2\pi}\exp(k^2/2 - k\,|w|) & |w| > k \end{cases} \tag{6.2}
$$

which is least favorable for the location problem. Here three different values $\epsilon = 0.01, 0.05$ and 0.10 with corresponding values $k = 1.945$, 1.399 and 1.140 (see Huber (1981)) were used. The Monte Carlo sample size was $N = 1000$.

For comparison we calculated the linear Kalman filter with $\mathrm{Var}(w_t) \equiv 1$ and an approximate conditional mean robust filter (acm.filt) implemented in S-PLUS, see Martin (1981).

For the Hampel two-part redescending psi-function in acm.filt we used the proposed default values $a = 2.5$ and $b = 5$. This was a substitute for the other robust methods considered in Schick and Mitter (1994) which were not easily available.

Table 1 shows the mean square errors averaged over time and replications when no outliers are present. The loss of efficiency is quite small and increases as expected with the assumed contamination rate ϵ. In Table 2 of Schick and Mitter (1994), where the results of the same simulation study are given, only the Guttman-Peña estimator with a small $\tilde{R}_{out}$ is slightly better, whereas the Masreliez-Martin estimator and the first-order approximation are both worse.

Table 2 illustrates the performance in the presence of an outlier. We computed the mean square errors averaged over replications at that time-point where the outlier occurs. For economy of space only the case $\alpha = 0.1$ is given. For the normal and Laplace contamination, each with variance 9, we get with contamination rate 0.05 or 0.1 the same or better results than with acm.filt. For the Cauchy and the Slash contamination the direct and kernel method perform for $\epsilon = 0.1$ as well as acm.filt. This can be explained by noting that for the location estimation the first two contaminations have a bounded ψ-function whereas the last two have a redescending ψ-function. In all cases the direct method is slightly better than the kernel. Even though Huber's density is no longer least favorable in our setting it seems to have a good performance over a wide range of observation noise distributions.
Our table 2 can not be compared directly to table 4 of Schick and Mitter (1994) because of the small size of the study. Especially in the case of Cauchy and Slash contamination the MSE of the linear Kalman filter at one time-point depends strongly on the simulated errors and the x_t value of the AR(1)-process. In the no-contamination case the MSE's are averaged over the whole series, so there we can better compare the two results.

6.5 AR(2)-Process with outliers

As a two-dimensional problem let us consider the following AR(2)-process

$$z_t = z_{t-1} - 0.4 z_{t-2} + v_t \qquad (6.3)$$

with $v_t \sim N(0, 1)$ and the observational equation

$$y_t = z_t + w_t \qquad (6.4)$$

where $w_t \sim N(0, 0.25)$ contaminated with some outliers.

It is well known that this model can be written as a two-dimensional state space model with $x_t = (z_t, z_{t-1})'$.

For our filter and smoother algorithms we assume $v_t \sim N(0, 1)$ and $w_t \sim 2q(2w)$ with $q(w)$ the Huber density (6.2).

Figure 10 shows the observations and the mean of the filter densities by the direct method according to Section 5, whereas Figure 11 gives the mean of the direct smoother. It can be seen that the three outliers were well recognized and filtered out. Figure 12 compares the estimated observation noise obtained by the direct filter and the acm.filt, and shows that the two methods are very close. Likewise, the results for the direct smoother and for the acm.smo of S-PLUS are almost identical.

References:

Andrade Netto, M. L., Gimeno, L. and Mendes, M.J. (1978), *On the Optimal and Sub-optimal Nonlinear Filtering Problem for Discrete-Time Systems.* IEEE Transactions on Automatic Control, AC-23 No. 6, 1062-1067.

Carlin, B.P., Polson, N.G. and Stoffer, D.S. (1992), *A Monte Carlo approach to non-normal and nonlinear state-space modeling.* J. Amer Statist Assoc, **87**, 439-500.

Harrison, P.J. and Stevens, C.F. (1976), *Bayesian Forecasting, (with discussion).* Journal of the Royal Statistical Society, Ser. B, **34**, 1-41.

Huber, P.J. (1981), *Robust Statistics.* John Wiley & Sons, New York.

Kitagawa, G. (1987), *Non-Gaussian state-space modeling of nonstationary time series, (with discussion).* J. Amer Statist Assoc, **82,** 1032-1063.

Kitagawa, G. (1993), *A Monte Carlo filtering and smoothing method for Non-Gaussian nonlinear state space models.* Research Memorandum, No. 462, The Institute of Statistical Mathematics, Tokyo.

Martin, R. D. (1981). *Robust methods for time series.* In Applied Time Series Analysis II. D. F. Findley, ed., Academic Press, New York. 683-759.

Schick, I.C. and Mitter, S.K. (1994), *Robust recursive estimation in the presence of heavy-tailed observation noise.* Ann Statist., **22**, 1045-1080.

Silverman, B.W. (1986), *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London.

Tanizaki, H. (1993), *Nonlinear Filters.* Lecture Notes in Economics and Mathematical systems, No. 400, Springer, Berlin.

West, M., Harrison, P.J. and Migon, H.S. (1985), *Dynamic generalized linear models and Bayesian forecasting, (with discussion).* J. Amer Statist Assoc, **80**, 73-97.

		AR(1)-Parameter $\alpha = 0.1$	AR(1)-Parameter $\alpha = 0.5$
Kernel	$\epsilon = 0.01$	0.68	0.67
	$\epsilon = 0.05$	2.99	3.34
	$\epsilon = 0.1$	6.39	7.20
Direct	$\epsilon = 0.01$	0.59	0.71
	$\epsilon = 0.05$	2.85	3.30
	$\epsilon = 0.1$	6.35	7.34
acm.filt (a=2.5,b=5)		1.52	1.33

Table 1: Percentage by which the MSE's of the filters exceed that of the linear Kalman filter for the no-contamination case.

		Contamination: Normal ($\sigma = 3$)	Laplace ($\sigma = 3$)	Cauchy	Slash
Kernel	$\epsilon = 0.01$	68.63	54.52	0.69	0.77
	$\epsilon = 0.05$	54.21	46.06	0.54	0.61
	$\epsilon = 0.1$	50.07	41.20	0.49	0.53
Direct	$\epsilon = 0.01$	68.30	55.47	0.67	0.77
	$\epsilon = 0.05$	54.66	44.95	0.53	0.60
	$\epsilon = 0.1$	48.50	40.72	0.49	0.53
acm.filt (a=2.5,b=5)		59.11	45.58	0.49	0.53

Table 2: MSE's of the filters as percentage of that of the linear Kalman filter at time 20 ($\alpha = 0.1$.)

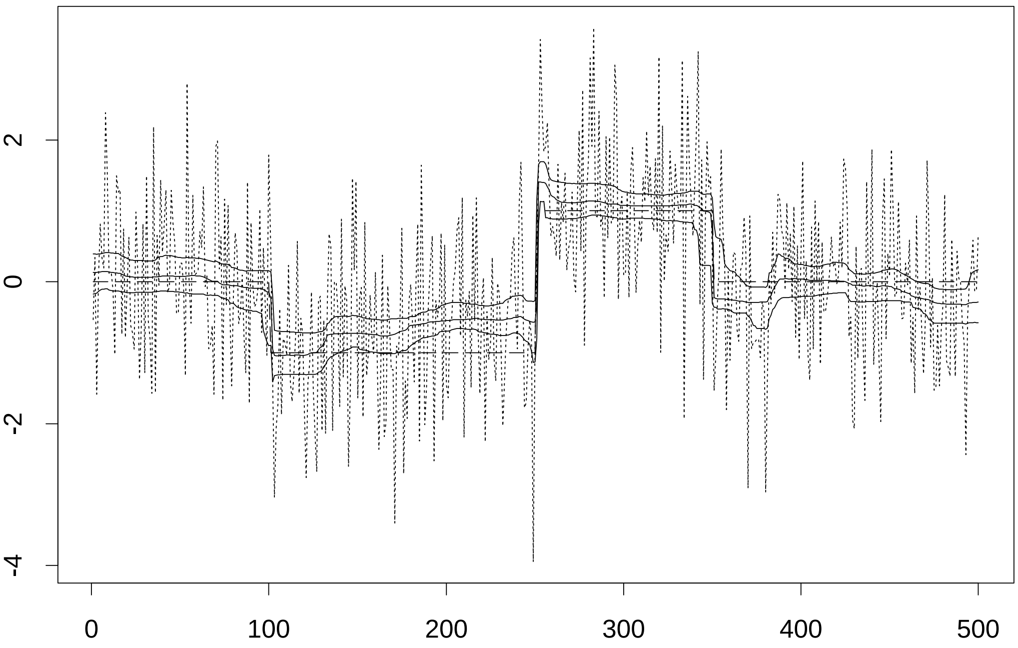

Figure 1: Shifted-mean model: Observations (dotted line) and state variable (dashed line) and the 10%-, 50%- and 90%-quantiles of the Monte Carlo smoother by the direct method (solid lines).

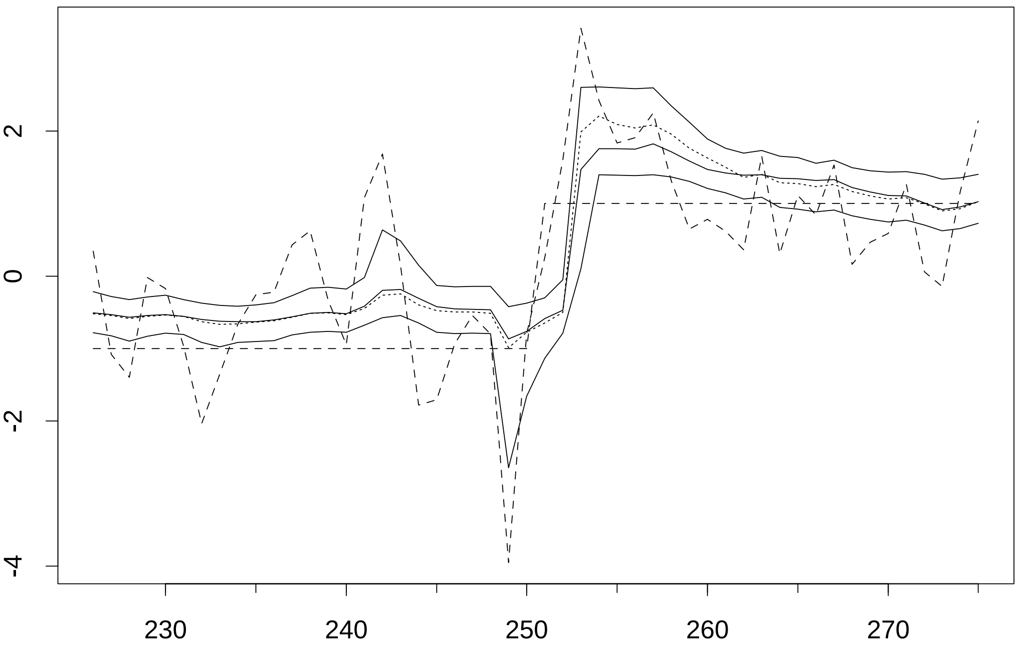

Figure 2: Behavior of Monte Carlo filter (direct method) near the jump at $t = 250$: Observations and state variable (dashed lines), 10%-, 50%- and 90%-quantiles of Monte Carlo filter (solid lines) and 50%-quantiles of filter with numerical integration (dotted line).

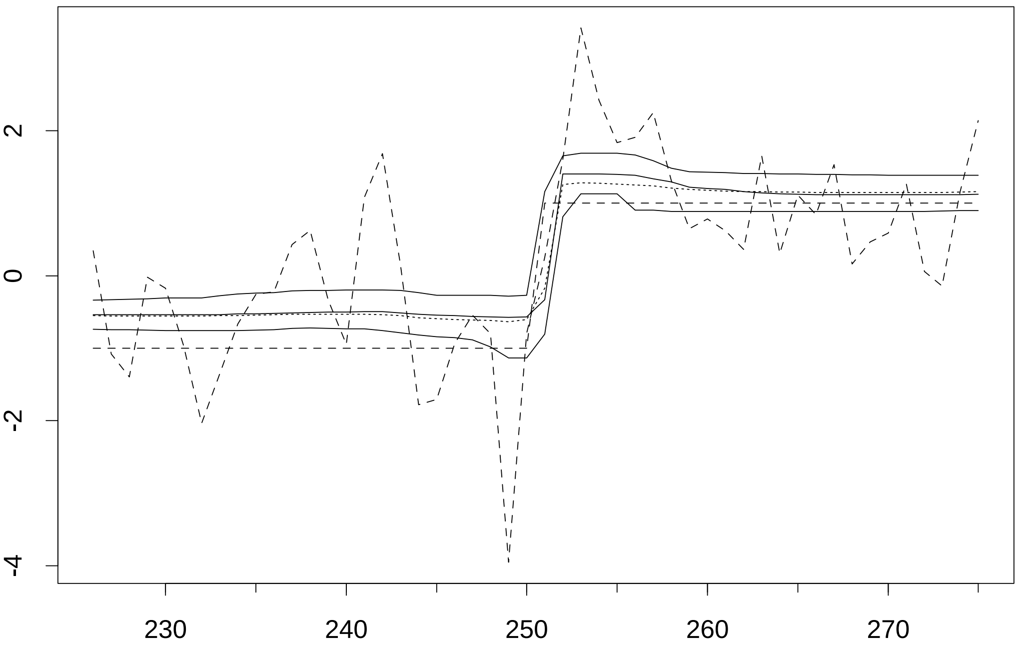

Figure 3: Behavior of Monte Carlo smoother (direct method) near the jump at $t = 250$: Observations and state variable (dashed lines), 10%-, 50%- and 90%-quantiles of Monte Carlo smoother (solid lines) and 50%-quantiles of smoother with numerical integration (dotted line).

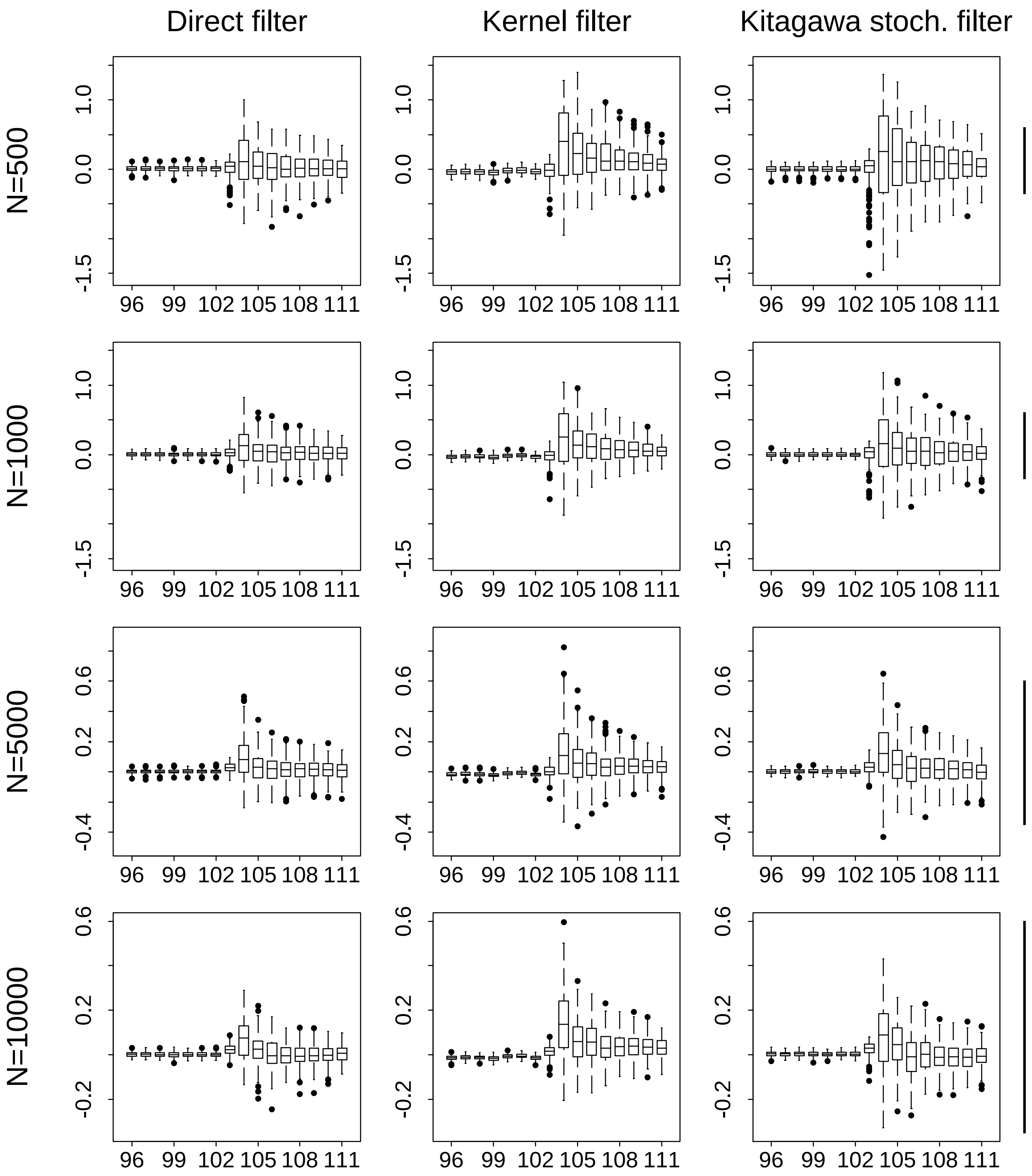

Figure 4: Shifted-mean model: Monte Carlo errors of different filters.

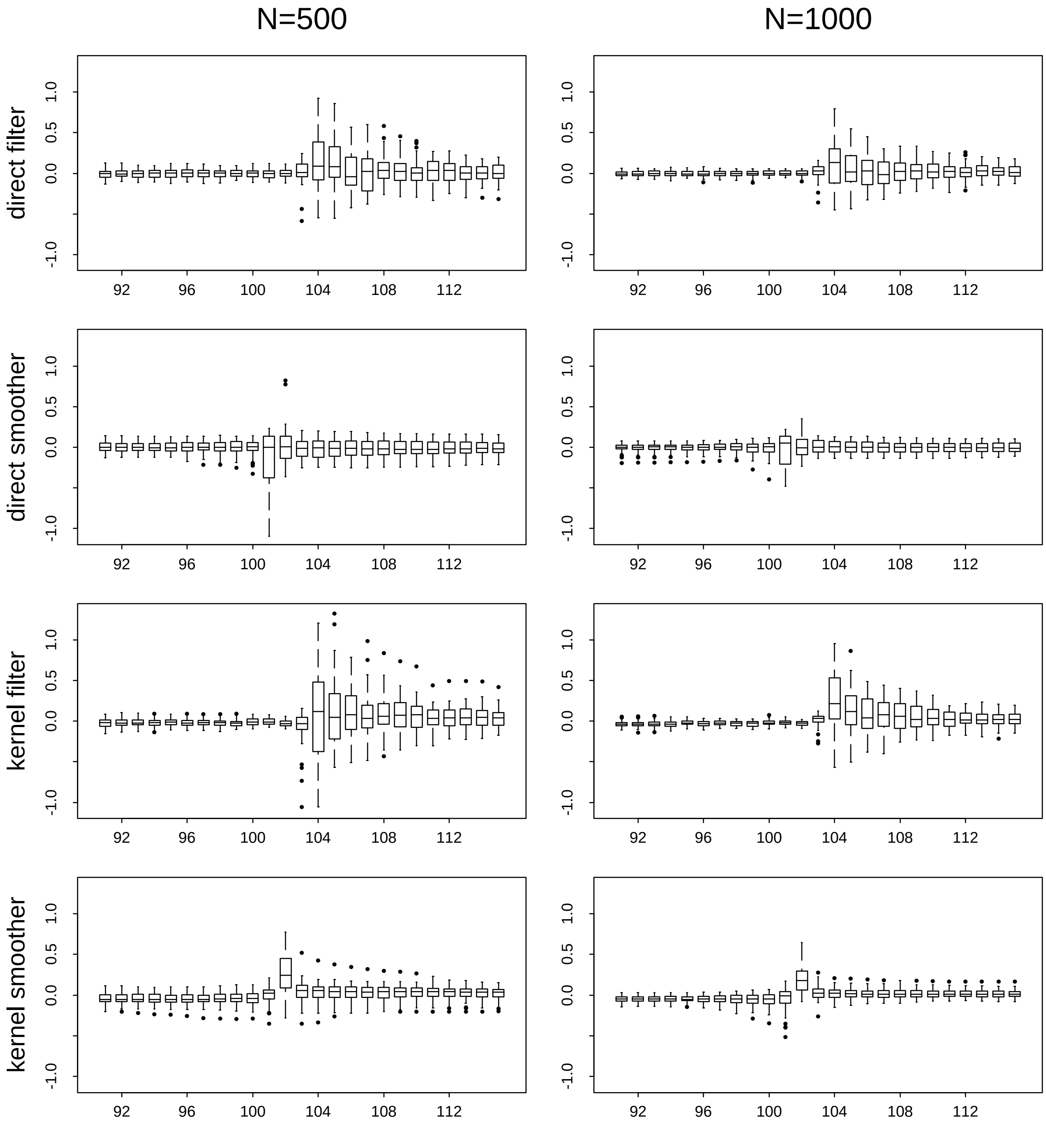

Figure 5: Shifted-mean model: Monte Carlo errors of filters and smoothers.

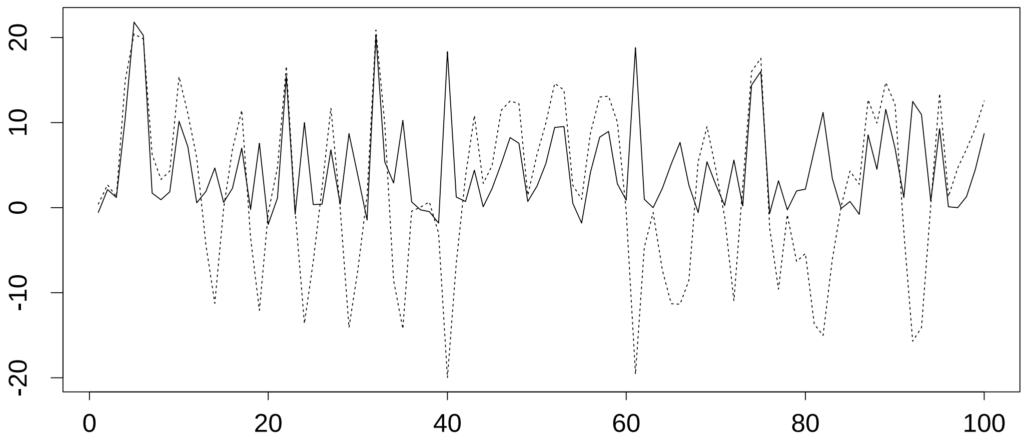

Figure 6: Nonlinear model: Simulated series of observation (solid line) and state variable (dotted line).

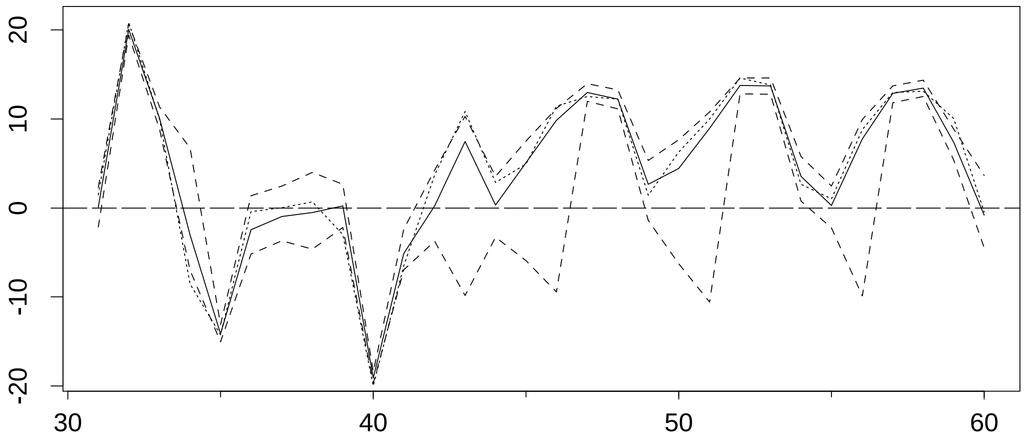

Figure 7: Behavior of Monte Carlo filter (kernel method) for $t = 31, ..., 60$: State variable (dotted line), 50%-quantiles of Monte Carlo filter (solid line) and 10%-, 90%-quantiles of Monte Carlo filter (dashed lines).

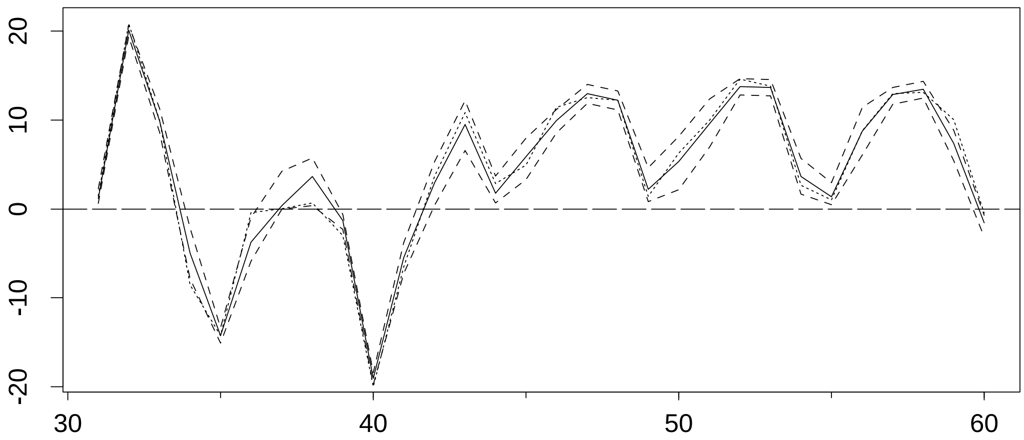

Figure 8: Behavior of Monte Carlo smoother (kernel method) for $t = 31, ..., 60$: State variable (dotted line), 50%-quantiles of Monte Carlo smoother (solid line) and 10%-, 90%-quantiles of Monte Carlo smoother (dashed lines).

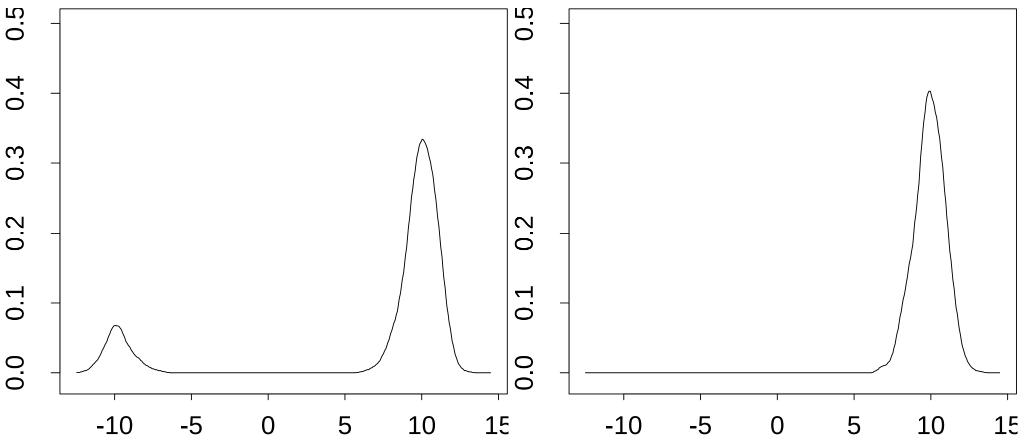

Figure 9: Density Esimation: Filter density (left figure) and smoother density (right figure, T=100) at $t = 46$ obtained by the kernel method.

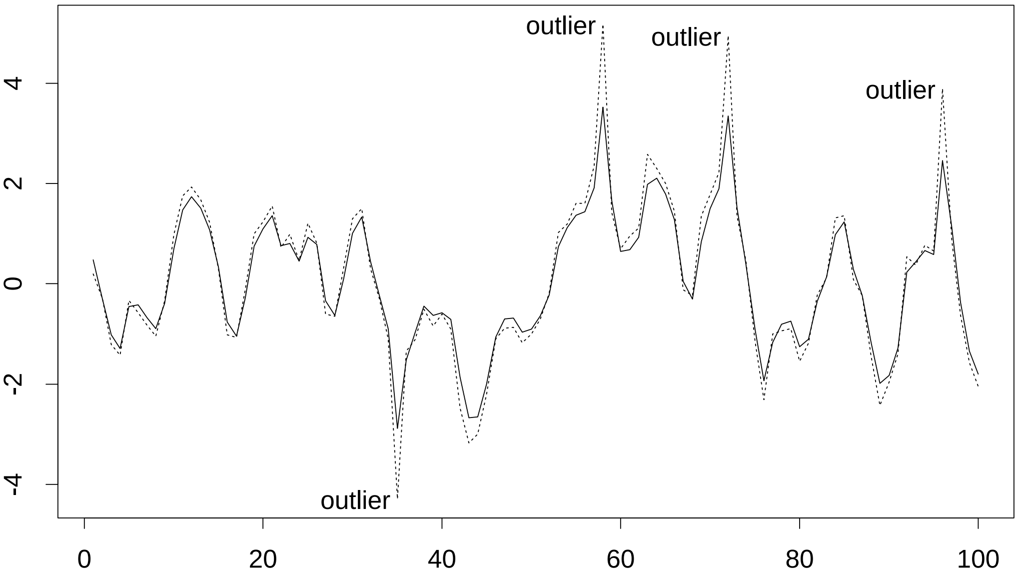

Figure 10: AR(2) with outliers: Mean of the direct filter (solid line) for sample size N=1000 and the observations y_t (dotted line). The four outliers are indicated.

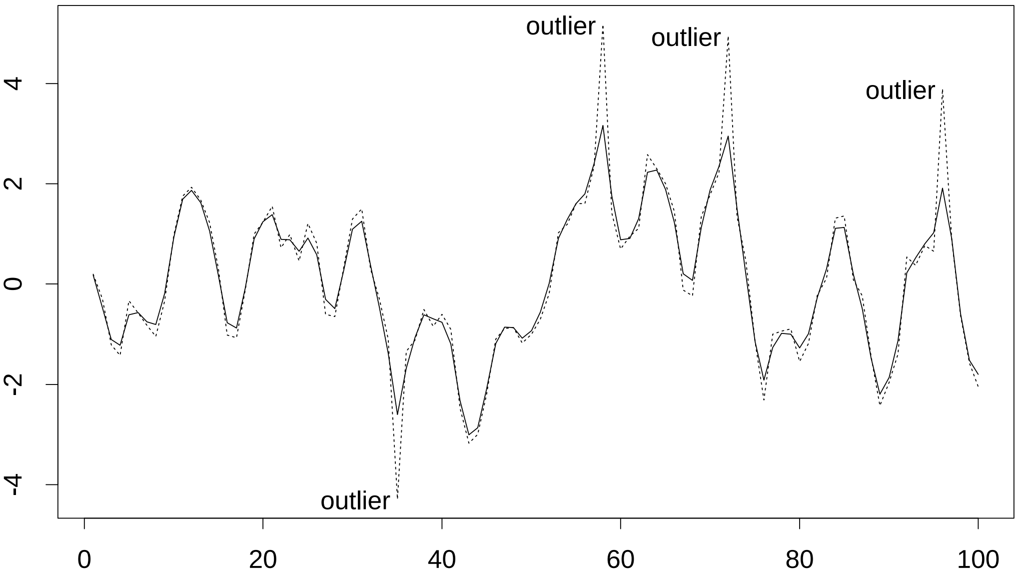

Figure 11: AR(2) with outliers: Mean of the direct smoother (solid line) for sample size N=1000 and the observations y_t (dotted line). The four outliers are indicated.

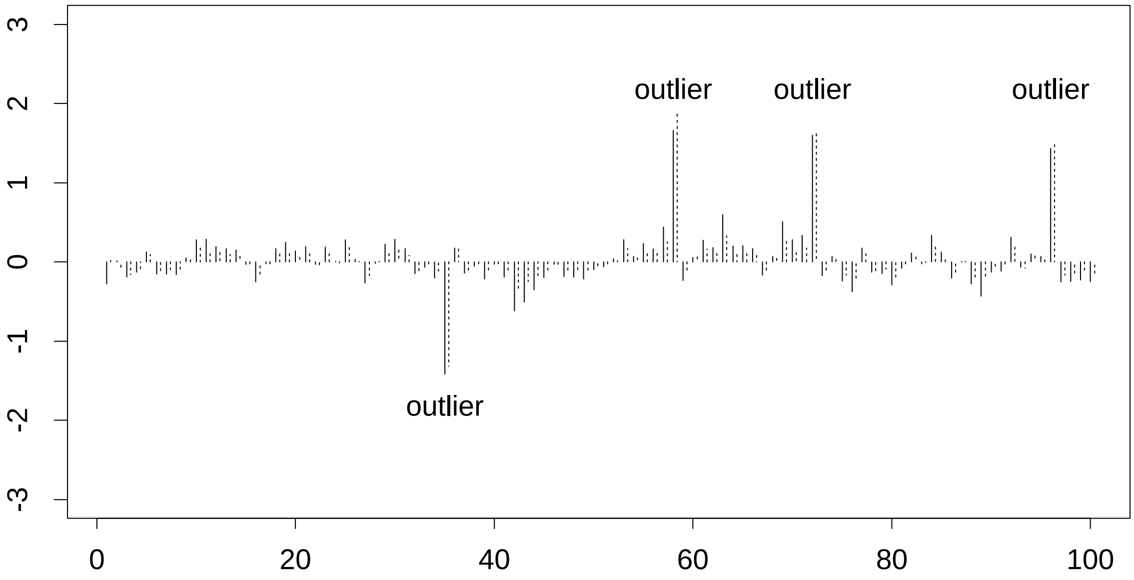

Figure 12: AR(2) with outliers: Observation noise estimated by the direct filter (solid) and the acm.filt (dotted).