Deterministic and stochastic particle filters in state space models

Erik Bølviken[1,2] *and Geir Storvik*[1,2]

[1]University of Oslo, [2]The Norwegian Computing Center

1 Introduction.

Optimal or Bayesian filtering in state space models is a question of computing series of linked numerical integrals where output from one is input to the other (Bucy and Senne 1971). Particle filtering can be regarded as techniques where these integrals are solved by replacing the complicated posterior densities involved by *discrete* approximations, based on *particles* (Kitagawa 1996). There is evidence that the numerical errors as the process is iterated often stabilize or at least do not accumulate sharply (see section 2.5). Such filters can be constructed in many ways. Most of the contributions to this volume employ Monte Carlo designs (see also Doucet (1998) and the references therein). Particles are then random drawings of state vectors under the current posterior. This amounts to Monte Carlo evaluations of integrals. Numerically inaccurate, but often easy to implement and practical, general methods to run the sampling have been developed. Alternatively, particles can be laid out through a deterministic plan, using more sophisticated and more accurate numerical integration techniques. This approach has been discussed in Kitagawa (1987), Pole and West (1988) and Pole and West (1990), but recently most work has been based on Monte Carlo methods. To some extent Monte Carlo and deterministic particle filters are complementary approaches, and one may also wonder whether they may be usefully combined (see Monahan and Genz (1997) for such a combination in a non-dynamic setting). Emphasis in this paper is on deterministic filtering. A general framework can be found in the above mentioned references and in West and Harrison (1997)[Section 13.5]. We shall present a common perspective in the next section, where our contribution will be on design issues.

Deterministic filtering could be the method of choice when the state process has low dimension and high numerical accuracy is wanted. Our applications, which are a group of statistical problems from population biology, is of that type. The biological signals in some of the examples in Section 3 are extremely weak, yet governed by well-defined stochastic models coming from basic theory on fluctuations of animal populations. The scientific objective is understanding the underlying mechanisms. This means model identification,

and in a frequency context, computation and maximization of likelihood functions. Filtering is then a mean to compute such quantities rather than an end in itself, but the computational problems remain much the same as in engineering where the interest is usually the filtering estimates. Even though our statistical applications are off-line, computer speed is still highly important, since the computations are often repeated many times to estimate uncertainty through bootstrapping. One difference from engineering, however, may be the attitude towards Monte Carlo error. In ad-hoc situations where a filtered estimate is used for, say guidance or control, added uncertainty that is a small fraction of the total may not matter much. In a context of basic science, as in Section 3, this could be different, and Monte Carlo error in likelihood evaluations can be awkward if the function is to be maximized. Various types of computing likelihoods will be compared in Section 3.

2 General issues.

2.1 Model and exact filter.

We are dealing with a discrete time vectorial process $\{\mathbf{x}_t\}$ in $\mathcal{R}^{n_x}$, observed indirectly through another process $\{\mathbf{y}_t\}$ in $\mathcal{R}^{n_y}$. The framework is a Markov model for $\{\mathbf{x}_t\}$ based on (possibly time-varying) transition densities $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and conditionally independent observations $\{\mathbf{y}_t\}$ given $\{\mathbf{x}_t\}$. The latter usually means that the conditional density of the vector $\mathbf{y}_{1:t} \stackrel{\Delta}{=} (\mathbf{y}_1, \ldots, \mathbf{y}_t)$ given the corresponding $\mathbf{x}_{1:t} \stackrel{\Delta}{=} (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ factors into $p(\mathbf{y}_{1:t}|\mathbf{x}_{1:t}) = \prod_{s=1}^{t} p(\mathbf{y}_s|\mathbf{x}_s)$, where $p(\mathbf{y}_s|\mathbf{x}_s)$ is the density of $\mathbf{y}_s$ given $\mathbf{x}_s$. The letter p will throughout be used to designate density functions of various kinds, for example in addition to those above, $p(\mathbf{y}_{1:t})$ for the density function of $\mathbf{y}_{1:t}$. These functions may depend on t themselves, say $p_t(\mathbf{y}_{1:t})$ or $p_s(\mathbf{y}_s|\mathbf{x}_s)$ rather than $p(\mathbf{y}_{1:t})$ or $p(\mathbf{y}_s|\mathbf{x}_s)$, but it is not necessary to include this in the notation.

The exact filter for $\{\mathbf{x}_t\}$ can be written as a set of recursive integration equations (Bucy and Senne 1971). Start with $p(\mathbf{x}_0) \stackrel{\Delta}{=} p(\mathbf{x}_0|\mathbf{y}_0)$ as prior for $\mathbf{x}_0$ and calculate recursively

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \tag{2.1}$$

$$C_t = \int p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) d\mathbf{x}_t \tag{2.2}$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = C_t^{-1} p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t-1}), \tag{2.3}$$

for $t = 1, \ldots, T$. The normalisation constants $\{C_t\}$ then produce the log-likelihood function of observations $\mathbf{y}_{1:T}$ through

$$\log\{p(\mathbf{y}_{1:T})\} = \sum_{t=1}^{T} \log(C_t). \tag{2.4}$$

The relationships (2.1-2.4) are well-known and are consequences of $\{\mathbf{x}_t\}$ being a Markov process and $\{\mathbf{y}_t\}$ being conditionally independent given $\{\mathbf{x}_t\}$. The proof is an elementary application of Bayes' formula.

2.2 Particle filters.

Computation of the posterior densities $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ and the log likelihood function $p(\mathbf{y}_{1:T})$ is an exercise in high-dimensional numerical integration. General purpose integration methods are bound to be inefficient, and it seems better to design iterative schemes imitating the exact one (2.1-2.3). Particle filters proceed in this manner through *discrete* approximations to the exact posterior distributions. Let $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ be some discrete analogue to the exact density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. The points $\mathbf{x}_t^{(i)}$ on which $\hat{p}(\mathbf{x}_t)$ assigns positive probabilities are known as the *particles*. Their number N_t may vary. Suppose a reasonable approximation $\hat{p}_{t-1}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ is available at time $t-1$. When inserted for the exact density $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ on the right in (2.1), we obtain

$$\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \sum_{j=1}^{N_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(j)})\hat{p}(\mathbf{x}_{t-1}^{(j)}|\mathbf{y}_{1:t-1}). \tag{2.5}$$

The main point of the design is to ensure that this is a good approximation to the exact predictive density $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. When (2.5) replaces its exact counterpart in (2.3), we immediately have

$$\tilde{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \widetilde{C}_t^{-1} p(\mathbf{y}_t|\mathbf{x}_t)\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}), \qquad (\widetilde{C}_t \text{ a constant}), \tag{2.6}$$

as approximate update density. To complete the recursion, (2.6) must be replaced by a particle approximation $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$. Filters proposed in the literature vary in how this step is carried out. Most of the papers of this volume use Monte Carlo sampling. If the particles $\mathbf{x}_t^{(i)}$ are drawn randomly from (2.6), the probabilities become $\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) = N_t^{-1}$. We may enhance numerical accuracy by importance sampling (Doucet 1998), the probabilities are then the importance weights. Many tricks have in recent years been invented to obtain computationally fast sampling.

Stochastic generation of particles corresponds to Monte Carlo evaluations of integrals. An alternative is to use other, more accurate integration methods. This raises, as we shall see in Section 3, various issues regarding design, but to understand this, we first review elements of the theory of numerical integration.

2.3 Gaussian quadrature.

Consider the case of a *scalar* process x_t, denoted x in this sequel. The integrands in (2.1) and (2.2) are of the form $h(x)p(x)$ with h some function

and p a density. Evaluations of integrals are often efficiently carried out by Gaussian quadrature. The simplest among these rules is the Gauss-Legendre method. The standard form is

$$\int_{-1}^{1} h(x)p(x)dx = \sum_{i=1}^{m} \gamma^{(i)} h(\xi^{(i)})p(\xi^{(i)}) + \mathcal{E}_0, \tag{2.7}$$

where the abscissas $\xi^{(1)}, \ldots, \xi^{(m)}$ and the positive weights $\gamma^{(1)}, \ldots, \gamma^{(m)}$ are tabulated in numerical literature; see also Press, Teukolsky, Vetterling and Flannery (1992) for a simple computer program. For a general interval (2.7) changes into

$$\int_{A}^{B} h(x)p(x)dx = \sum_{i=1}^{m} w^{(i)} h(x^{(i)})p(x^{(i)}) + \mathcal{E}, \tag{2.8}$$

where

$$x^{(i)} = \frac{1}{2}(A + B + (B - A)\xi^{(i)}), \qquad w^{(i)} = \frac{1}{2}(B - A)\gamma^{(i)}. \tag{2.9}$$

The error term $\mathcal{E}$ is

$$\mathcal{E} = \frac{1}{2}(B - A)^{2m+1} \frac{(m!)^4}{\{(2m)!\}^3(2m + 1)} (ph)^{(2m)}(\omega), \tag{2.10}$$

where $(ph)^{(2m)}(\omega)$ is the derivative of order $2m$ of ph at some $\omega \in (A, B)$. The error term thus vanishes if the integrand is a polynomial of degree $< 2m$, and the accuracy is almost startling otherwise. Assuming $B - A = 10$, $\mathcal{E}$ being of order 10^{-10} times the derivative for $m = 10$ and of 10^{-32} for $m = 20$ for $B - A = 10$ (but higher order derivatives often grows!).

The integral approximation (2.8-2.9) has a probabilistic interpretation. Suppose the probability mass of p is negligible outside the interval (A, B). Define

$$\hat{p}(x^i) = w^{(i)} p(x^{(i)}) \tag{2.11}$$

and insert $h \equiv 1$ in (2.8). Then

$$\sum_{i=1}^{m} \hat{p}(x^{(i)}) \approx \int_{A}^{B} p(x)dx \approx 1,$$

so that $\hat{p}(x^{(1)}), \ldots, \hat{p}(x^{(m)})$ is almost a proper probability vector. Normalisation is usually not worth the bother. Let $\hat{p}$ be the distribution assigning these probabilities to $x^{(i)}$, $i = 1, \ldots, m$. Then (2.8) expresses that $\hat{p}$ is an approximation to p in the sense that the expectation of $h(x)$ for any *smooth* function h is almost equal for the two distributions. Note that $\hat{p}(x^{(i)})$ and $p(x^{(i)})$ deviate.

Other quadrature rules produce similar particle approximations through an idea similar to importance sampling. *Choose* some density ψ and note that $\int hp = \int hR\psi$ where $R = p/\psi$. Quadrature with ψ as a weight function yields

$$\int h(x)p(x)dx = \int h(x)R(x)\psi(x)dx = \sum_{i=1}^{m} \gamma^{(i)} h(\xi^{(i)})R(\xi^{(i)}) + \mathcal{E}, \qquad (2.12)$$

where abscissas $\xi^{(i)}$ and weights $\gamma^{(i)}$, which depend on ψ, differ from those in (2.7). The error term now vanishes if hR is polynomial of degree less that $2m$. The particle approximation (similar to (2.11)) becomes $x^{(i)} = \xi^{(i)}$ and $\hat{p}(x^{(i)}) = \gamma^{(i)} p(x^{(i)})/\psi(x^{(i)})$.

2.4 Quadrature filters.

Quadrature filters are constructed by replacing the density $\tilde{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ in (2.6) by a particle approximation based on the quadrature formulas in Section 2.3. This leads to the following recursive scheme, where

$$\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) = \sum_{j=1}^{N_{t-1}} p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(j)})\hat{p}(\mathbf{x}_{t-1}^{(j)}|\mathbf{y}_{1:t-1}) \qquad (2.13)$$

$$\widehat{C}_t = \sum_{i=1}^{N_t} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) \qquad (2.14)$$

$$\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) = \widehat{C}_t^{-1} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}), \qquad (2.15)$$

for $i = 1, \ldots, N_t$. Implementing the scheme (2.13-2.15) when $n_x > 1$ raises a number of problems. The pretty mathematical theory of Gaussian quadrature described above is inherently one-dimensional and must be applied sequentially, one state variable at a time. The weights $w^{(i)}$ in (2.14) and (2.15) will then be *products* of weights from each of the n_x variables; see Section 3.2 for details. This also suggests that the computational requirements will grow rapidly with n_x. The number of operations in quadrature filtering is proportional to N_t^2 for each step of the recursion. When $n_x = 1$, as in the example, N_t equaled the order m of the quadrature rule. But if n_x abscissas are used for each variable of the state vector, then $N_t = m^{n_x}$, easily a huge number. A particle representation of n_x *independent* variables needs actually such a large N_t to be accurate, but if there are correlations, it may be possible to cut it down.

2.5 Numerical error.

It is possible to gain some insight into how numerical error propagates by elementary methods. Let

$$\delta_t = \log\{\hat{p}(\mathbf{y}_{1:t})\} - \log\{p(\mathbf{y}_{1:t})\} = \sum_{s=1}^{t} \log \widehat{C}_s - \sum_{s=1}^{t} \log C_s \tag{2.16}$$

be the accumulated error in the log-likelihood function, and define

$$\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \exp(\delta_{t-1})\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) - p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \tag{2.17}$$

as an indicator of numerical error present in $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. The factor $\exp(\delta_{t-1})$ is to avoid normalisations in the recursion (2.18) below.

By elementary manipulations it is easily proved (see appendix) that

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = C_t^{-1} \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{y}_t|\mathbf{x}_t)\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t + \eta(\mathbf{x}_{t+1}), \tag{2.18}$$

where

$$\eta(\mathbf{x}_{t+1}) = \exp(\delta_t)[\sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_t) - \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)\tilde{p}(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t]. \tag{2.19}$$

If we insert (2.3) into (2.1) it emerges that the sequence of posterior densities $\{p(\mathbf{x}_t|\mathbf{y}_{t-1})\}$ themselves satisfy exactly the same recursion (2.18) but for the term $\eta(\mathbf{x}_{t+1})$ which signifies the contribution of the numerical error at time t. Suppose now that r_t is the smallest real number for which

$$|\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \leq r_t p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \tag{2.20}$$

so that r_t controls the *relative error* in $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. Then, by inserting (2.20) in the integrand (2.18) it follows that

$$r_{t+1} \leq r_t + \tilde{\eta}(\mathbf{x}_{t+1}), \tag{2.21}$$

where

$$\tilde{\eta}(\mathbf{x}_{t+1}) = \eta(\mathbf{x}_{t+1})/p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \tag{2.22}$$

is the relative numerical error introduced at time t.

Relative error in the approximations *diminishes* through the next step of the scheme. New error is brought, as well, but the old relative error does not blow up as the recursion is progressed.

It is for Gaussian quadrature filters possible to give a rigorous error statement. Suppose we have been able to find approximations so that

$$|\sum_{j=1}^{N_{t-1}} w_t^{(j)} p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(j)}) p(\mathbf{x}_{t-1}^{(j)}|\mathbf{y}_{1:t-1}) - p(\mathbf{x}_t^{(i)}|\mathbf{y}_{t-1})| \leq \varepsilon p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) \quad (2.23)$$

for all i, t. Note that this condition is for $p(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t})$ *known* and for a *finite set of* $\mathbf{x}_{t+1}^{(i)}$ *values*, which is always possible to obtain my choosing N_t large enough. It is then proved in the appendix that

$$r_{t+1} \leq r_t(1+\varepsilon)^t + \varepsilon. \quad (2.24)$$

which implies that

$$r_t \leq (1+\varepsilon)^t - 1. \quad (2.25)$$

By defining

$$\varepsilon(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \exp(\delta_{t-1})\hat{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}) - p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) \quad (2.26)$$

and replacing condition (2.23) with

$$|\sum_{i=1}^{N_t} w_t^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) - p(\mathbf{y}_t|\mathbf{y}_{t-1})| \leq \varepsilon p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) \quad (2.27)$$

for all t, it is by similar arguments possible to show that

$$|\varepsilon(\mathbf{y}_t|\mathbf{y}_{1:t-1})| \leq [r_t(1+\varepsilon)+\varepsilon] p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) \quad (2.28)$$

But

$$\varepsilon(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \exp(\delta_{t-1})\widehat{C}_t - C_t = [\exp(\delta_t) - 1] p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$$

which results in that

$$|\exp(\delta_t) - 1| \leq r_t(1+\varepsilon) + \varepsilon \leq (1+\varepsilon)^{t+1} - 1 \quad (2.29)$$

and consequently that

$$\delta_t \leq (t+1)\log(1+\varepsilon), \quad (2.30)$$

that is, the error is approximately linear in t. This shows that although the error will increase with time, it will not accumulate sharply. Note that in general, a better limit for the error in the likelihood evaluation is not possible. This can be seen by considering the situation where all observations are independent, where it is straightforward to show that the error involved in the computation of the log-likelihood is the sum of errors at each time point.

2.6 A small illustrative example

In order to compare the quadrature filter with the Monte Carlo filter, we will consider the following one-dimensional nonlinear reference model (Doucet 1998):

$$x_t = \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + e_t \tag{2.31}$$

$$z_t = \frac{x_t^2}{20} + v_t \tag{2.32}$$

where $x_0 \sim \mathcal{N}(0,5)$, e_t and v_t are mutually independent white Gaussian noises, $e_t \sim \mathcal{N}(0,\sigma_e^2)$ and $e_t \sim \mathcal{N}(0,\sigma_v^2)$ with $\sigma_e^2 = 10$ and $\sigma_v^2 = 1$. Data were simulated according to this model for $t = 1,...,T = 100$. The quadrature filter was implemented as described in Section 2.4, while rejection sampling with the predictive distribution as a proposal was used for the Monte Carlo filter. Although more effective Monte Carlo filters could be used, our main issue here is to compare the variability in the result, making the choice of Monte Carlo filter less important.

The filters were first run with a large number of particles ($N_t = N = 500$ for the quadrature filter and $N_t = N = 2000$ for the Monte Carlo filter). The results from the two filters agreed in this case, giving us the "true" posterior as a reference. In Figure 1 the estimates of the predictive density $p(x_{t+1}|y_{1:t})$ for $t = 100$ are displayed. For both filters, $N = 50$ were used. The true density is given by the solid curve. The Monte Carlo filter (long dashed curve) is able to find the two modes of the density, but is somewhat wrong on the weights on these modes. The quadrature filter (dotted) curve, on the other hand, is able to obtain the true density very well on almost all parts.

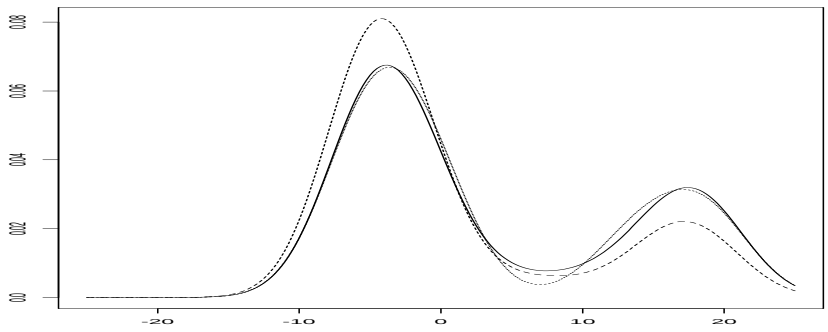

Figure 1: *True and estimated predictive density $p(x_{101}|y_{1:100})$ for the reference model. The true density is given as a solid curve, the Monte Carlo filter estimate is given as a long dashed curve, while the quadrature estimate is given by dotted curve.*

3 Case studies from ecology

3.1 Problem area and models.

Population biology is an area where optimal filtering is likely to reap substantial benefits, and we shall in this section indicate what can be achieved with such methods. It is generally recognized that animal populations oscillates stationary due to feedback links between predator and prey and interactions with environment. A reasonable mathematical framework, see Royama (1992), Stenseth, Bjørnstad and Falck (1996), is to impose non-linear autoregressive models of low order on logarithms of abundance. Parameters of the models, influenced by vegetation and climatic conditions, might vary between sites. There is a chronic shortage of data. The usual type are the number of animals caught in traps annually or seasonly, sometimes gathered by scientific experiments, sometimes through proxies like the number of skins delivered to a company. The famous lynx series, used in countless textbooks on statistical time series analysis (for example Tong (1990)) are of the latter type.

The traditional approach in ecology (Stenseth, Bjørnstad and Saitoh 1996, Stenseth, Falck, Bjørnstad and Krebs 1997, Stenseth, Falck, Chain, Bjørnstad, Donoghue and Tong 1998)) is to identify the data with the animal population, but it is clearly more correct to regard data and population as different entities and employ a state model. This yields a more realistic picture of statistical uncertainties, deal with zero observations and open for the utilization of sources of data that have traditionally *not* been used scientifically; see below. Populations are, on first approximation, described by *linear* autoregressive models on *log-scale*, say

$$x_t = a_1 x_{t-1} + \ldots a_r x_{t-q} + e_t, \tag{3.1}$$

where $\{e_t\}$ is independent noise with zero mean and constant variance. The mean of x_t is not estimable under the measurement models introduced below and has been subtracted out. Thus, in (3.1) x_t is the *difference* from the long term average. Interest is directed towards the period (i.e. the location of the maximum of the spectral density), the autoregressive coefficients (which have ecological interpretations) and the order q of the series (which relates to ecological hypotheses).

The linear model (3.1) is useful, but it fails to deal with the hypothesis that the built-up phase of animal abundance tends to be *slower* than when the population is on the decline. There is for the lynx empirical evidence supporting such an hypothesis; see Stenseth et al. (1997). One possibility, utilised by these authors, is to impose a TAR (truncated autoregressive) model of the type outlined in Tong (1990). The single autoregressive relationship is then

split into two separate ones, i.e.

$$x_t = \begin{cases} a_1 x_{t-1} + \ldots a_r x_{t-q} + e_t, & \text{if } x_{t-d} \leq \theta; \\ b_1 x_{t-1} + \ldots b_r x_{t-q} + e_t, & \text{if } x_{t-d} > \theta. \end{cases} \tag{3.2}$$

This expresses that the population changes into a different phase (i.e. decline) d time units after the threshold θ has been passed. Mathematical properties of these models are summarised in Tong (1990).

We shall consider three different measurement regimes $\{y_t\}$. The most common ones are *counts* (based on trappings). The natural model for y_t given x_t is then

$$p(y_t|x_t) = \frac{\lambda_t^{y_t}}{y_t!}\exp(-\lambda_t), \qquad \log(\lambda_t) = \alpha + \beta x_t. \tag{3.3}$$

Note that the parameter β influences the distribution of $\{y_t\}$ through $\beta\sigma_e$, where σ_e is the standard deviation of the series $\{e_t\}$ in (3.1) and (3.2). It is therefore not possible to estimate β and σ_e jointly. The same problem occurs in the other observation models (3.4) and (3.5) below.

Other types of data are *categorical* based on judgments by experienced field observers. These sources of information have never been utilised scientifically, and it would be a welcome additional source of information in population biology if they could. One type of assessment is *binary*, y_t being 0 or 1, $y_t = 0$ signifying a population below average and $y_t = 1$ above. Such judgments are with errors. There can be no precise definition as to what these categories precisely means. A plausible model is

$$p(y_t|x_t) = p_t^{y_t}(1-p_t)^{y_t}, \qquad p_t = \frac{\exp(\alpha + \beta x_t)}{1 + \exp(\alpha + \beta x_t)}, \tag{3.4}$$

Here β is close to zero for bad observers and very large for good ones. The other parameter α captures *bias* in observer evaluation (upwards if $\alpha > 0$ and downwards if $\alpha < 0$). Other observations may be in terms of *differences* $x_t - x_{t-1}$ (*ratios* on the original scale). We shall in the next section discuss an example where $y_t \in \{-1, 0, 1\}$, where $y_t = -1$ means decline from one time point to the next, $y_t = 0$ signifies (at most) moderate changes and $y_t = 1$ growth. A possible model is now

$$p(y_t|x_t) = \begin{cases} c\exp(-\alpha_n - \beta(x_t - x_{t-1})) & \text{if } y_t = -1 \\ c & \text{if } y = 0 \\ c\exp(-\alpha_p + \beta(x_t - x_{t-1})) & \text{if } y_t = -1, \end{cases} \tag{3.5}$$

where c is a constant forcing the three probabilities to sum to one. Interpretations of the three parameters α_n, α_p and β are similar to those in (3.4).

3.2 Quadrature filters in practice.

Constructing quadrature filters is the question of selecting particles $\mathbf{x}_t^{(i)}$ and their associated weights $w_t^{(i)}$ in the scheme (2.13-2.15) above. In the present instance several issues are involved in this: How should joint particle grids in several variables be defined? How is the discontinuity in the TAR model (3.2) handled? The process $\{x_t\}$ describing the population is stationary and varies within a well defined, in practice limited region. Should that be utilised?

It is most convenient, and no loss in ideas, to answer these questions when the autoregressive processes (3.1) and (3.2) have order $q = 2$. The state vector at t is then (x_t, x_{t-1}) with state representation

$$\begin{pmatrix} x_t \\ x_{t-1} \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ x_{t-2} \end{pmatrix} + \begin{pmatrix} e_t \\ 0 \end{pmatrix}. \tag{3.6}$$

Note the zero noise in the second component. The relation (2.1) of the exact filter becomes

$$p(x_t, x_{t-1}|y_{1:t-1}) = \int_{-\infty}^{\infty} p(x_t|x_{t-1}, x_{t-2}) p(x_{t-1}, x_{t-2}|y_{1:t-1}) dx_{t-2}. \tag{3.7}$$

which is a unidimensional integral in spite of the state vector having two variables and would have been univariate even for general q. Autoregressive processes are special in that only one (of q) relationships of the state representation has non-zero noise. This is responsible for the univariate integral (3.7) and is actually a blessing. A general quadrature filter requires $O(N_t^2)$ operations for each step t of the recursion, but the constructions below are much faster.

However, the degenerate noise in (3.6) makes it less clear-cut how the joint grid for (x_t, x_{t-1}) should be constructed. At first glance it may seem natural to use the version on the left of Figure 2, which utilizes the correlation structure between x_t and x_{t-1}. Skew grids, as in Figure 2 left, follows by applying univariate particle approximations to the two factors sequentially. First quadrature abscissas $x_{t-1}^{(j)}$ (needed for later integration) are laid out for x_{t-1} and the second approximation is conditioned on the first. The problem with this design is that the number of particles will grow explosively. Suppose there are N_{t-1} different particles for x_{t-1}. Each of them is unchanged as it moves ahead and each gives birth to m different particles for x_t, making $N_t = mN_{t-1}$. One way to keep this under control is to *change* the grid with *fewer* particles before progressing, but that requires the posterior densities to be interpolated numerically which add to the error and is not cheap computationally. Interpolation from skew grids, such as that in Figure 2 left, is also cumbersome to implement, especially in higher dimensions.

We have tested such versions, and they certainly work, but for stationary autoregressive processes we tend to prefer the simpler solution in Figure 2

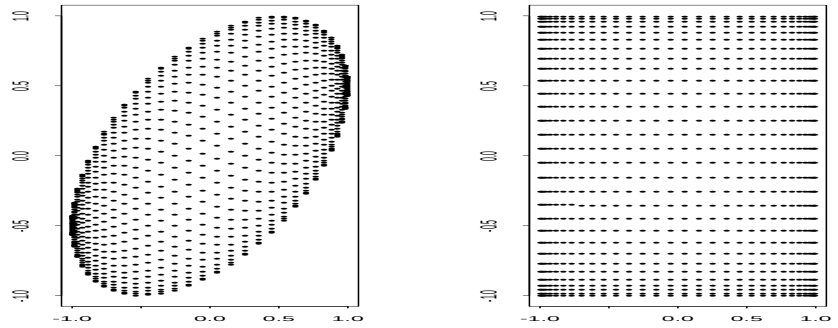

Figure 2: *Particles for* (x_t, x_{t-1}) *as Gauss-Legendre abscissas in each dimension. On the left: Particles constructed by conditioning* x_t *on* x_{t-1}*, on the right: particles constructed independently in each direction.*

right where the particles have been laid out as quadrature abscissas for the *marginal* densities of the two variables. This is less parsimonious as approximations to the joint distribution, but now the number of particles can be kept constant as the algorithm progresses without interpolation. In detail these methods work as follows when based on Gauss-Legendre quadrature:

First a particle approximation to the density $p(x_0, x_{-1})$ is needed as initialisation. In the linear, Gaussian case the corresponding distribution has zero means and an easily computable covariance matrix; see Priestley (1981). If σ_x is the standard deviation, we may apply (2.9) to each variable with $A = -B_0\sigma_x$ and $B = B_0\sigma_x$ for, say $B_0 = 5$. The square with edges $(\pm B_0\sigma_x, \pm B_0\sigma_x)$ then contains virtually the whole probability mass of the initial density $p(x_0, x_{-1})$, and the bivariate particle approximation becomes

Initialisation

$$x_0^{(i)} = x_{-1}^{(i)} = B_0\sigma_x\xi^{(i)} \qquad (3.8)$$
$$\hat{p}(x_0^{(i)}, x_{-1}^{(j)}) = p(x_0^{(i)}, x_{-1}^{(j)})B_0^2\gamma^{(i)}\gamma^{(j)}.$$

Closed form expressions for the joint density $p(x_0, x_{-1})$ is rarely available other than in the Gaussian, linear case. A technique which works well (and which was used for the TAR model) is to run the main recursion below a suitable number of steps *without data* from a start such as in (3.8). Since $\{x_t\}$ by definition is stationary, the recursion will settle at steady state. For the TAR model, where simple necessary and sufficient conditions for stationarity are not available (see Tong (1990)), this is actually a way of checking for stationarity.

Next comes the main recursive step. When adapting the general scheme (2.13-2.15) by means of Gauss-Legendre integration we obtain the following method (with $N_t = N$ for all t):

Algorithm 3.1 (Main recursion)
Choose A_t and B_t (see text).
Put

$$x_t^i = \frac{1}{2}(A_t + B_t + (B_t - A_t)\xi^{(i)}), \tag{3.9}$$

$$w_t^i = \frac{1}{2}(B_t - A_t)\gamma^{(i)}. \tag{3.10}$$

Update the probabilities by

$$\hat{p}(x_t^{(i)}, x_{t-1}^{(j)}|y_{1:t-1}) = \sum_{k=1}^{N} p(x_t^{(i)}|x_{t-1}^{(j)}, x_{t-2}^{(k)})\hat{p}(x_{t-1}^{(j)}, x_{t-2}^{(k)}), \tag{3.11}$$

$$\hat{q}(x_t^{(i)}, x_{t-1}^{(j)}|y_{1:t}) = p(y_t|x_t^{(i)}, x_{t-1}^{(j)})\hat{p}(x_t^{(i)}, x_{t-1}^{(j)}|y_{1:t-1})w_t^{(i)}w_{t-1}^{(j)} \tag{3.12}$$

$$\widehat{C}_t = \sum_{i=1}^{N}\sum_{j=1}^{N} \hat{q}(x_t^{(i)}, x_{t-1}^{(j)}|y_{1:t}) \tag{3.13}$$

$$\hat{p}(x_t^{(i)}, x_{t-1}^{(j)}|y_{1:t}) = \widehat{C}_t^{-1}\hat{q}(x^{(i)}, x^{(j)}|y_{1:t}), \tag{3.14}$$

for $i, j = 1, \ldots, N$.

The form of the density functions $p(y_t|x_t, x_{t-1})$ (equation (3.12)) will vary for the different measurements models. Extension to higher order processes is straightforward. A modification is needed for the TAR model to cope with the discontinuity at θ. The interval (A_t, B_t) must now be split into two pieces, i.e. (A_t, θ) and (θ, B_t), and the Gauss-Legendre abscissas applied to each part (if $\theta \notin (A_t, B_t)$ only one of the pieces is needed). The algorithm is in all other respects as above.

Several strategies can be used for selecting A_t and B_t. The simplest is to keep them unchanged, i.e. $A_t = A_0$ and $B_t = B_0$. The grids are then exactly the same everywhere, and the first two lines of the algorithm disappear. One advantage is that the transition probabilities $p(x_t^{(i)}|x_{t-1}^{(j)}, x_{t-2}^{(k)})$ do not change with t and can be prestored. This is our preferred method when dealing with the very noisy observations (3.4) and (3.5), but it is a possibility *only* because $\{x_t\}$ is stationary. In reality many of the particles $(x_t^{(i)}, x_{t-1}^{(j)})$ have very low probabilities a priori and can be removed with no noticeable effect. When $q = 3$, this trick can lead to huge reductions, up to several hundred times. When the data is more informative, as under the Poisson model (3.3), this strategy of fixed particles is not effective, and it is better to adapt (A_t, B_t) for each t. There are many ways to do so. A rough way is first to compute, from the observed y_t, an interval where x_t is highly likely to be located, then a similar one from the prediction density $p(x_t|y_{1:t-1})$ and use their *intersection* as (A_t, B_t). This worked well in all our examples. Note that this must be done *prior* to finding updated means and variances of x_t, and it is not a good idea to allow the computer to spend too much time on this detail.

Scenario	Population			Observations				
	a_1	a_2	σ_ε	Type	α	β	T	S
I	-0.5	-0.5	1.0	Poisson (3.3)	1.0	1.0	50	1
II	-0.5	-0.5	1.0	Logistic (3.4)	0.0	2.0	50	1
III	1.5	-0.9	1.0	Categorical (3.5)	-0.8	-0.8	17	100

Table 1: Scenarios for the linear autoregression used in Figure 3. S is the number of time series and T is the length of each time series.

3.3 Numerical experiments.

Different methods will in this section be used to evaluate likelihoods under the models in Section 3.1. Results produced by quadrature filters and Monte Carlo filters will be compared, and we shall for linear autoregression (3.1) also investigate a simple approximate technique proposed in Schnatter (1992), which is a special case of the so-called Gaussian sum filter in control engineering (Sorenson and Alspach 1971). Instead of working with particle approximations in the scheme described in Section 2.2 we then use *Gaussian* ones instead; see Schnatter (1992) for details.

We have in Figure 3 computed likelihood functions from simulated data based on the *linear* model (3.1) under the three scenarios shown in Table 1, all taken from estimates obtained from real data. All data are annual. Scenarios I and II represent lemmings or other small rodents. The periods of the population models (corresponding to the maximum of the spectral density function) is a little over three years and the length of the series (50 years) representative for real data. The third scenario is constructed from a real material for the Canadian snow shoe hare; see Bølviken, Glöckner and Stenseth (1999). The period is now much longer (about 9.6 years), and the research question is whether it is possible to recover the dynamics of the animal fluctuations from a large number of short series of 17 years.

It is fairly clear that the categorical data must contain information about the frequencies of oscillations of the underlying populations, but perhaps not equally obvious that they can recover other information about the dynamics. We have therefore in Figure 3 plotted likelihood profiles along curves of *constant* periods in the (a_1, a_2)-space. The values of the nuisance parameters in the measurement models (3.3-3.5) are the correct ones, except on the lower right where the horisontal axis represent variations of β. The results are encouraging. In scenario III (lower half of Figure 3) where much data are behind the computations, it is indicated that the parameters can be consistently estimated, in accordance with the theoretical results in Bickel, Ritov and Ryden (1998). The Monte Carlo effects are there much dampened as averages of 100 independent likelihoods, one for each 17 year series. The similarity between methods are great in all cases considered, but the Monte Carlo versions (based on 1000 replications) for scenario I and II are too erratic

Scenario	Population				
	a_1	a_2	b_1	b_2	σ_ε
I'	-0.5	0.0	-0.5	-0.7	1.0
III'	1.0	-0.4	1.0	-0.9	1.0

Table 2: Scenarios for the TAR autoregression used in Figure 4. S is the number of time series and T is the length of each time series. The observation models are equal to scenarios I and III in Table 1.

to optimize by numerical software. By contrast the likelihood surfaces for the two other methods are smooth. The closeness in shape between the results from the Schnatter approximation and the more accurate quadrature filter is noteworthy, and very impressive since Schnatter's method is extremely fast. However, there are numerical errors in the *level* of the likelihood curves produced by her method. If model fit is to be judged through AIC or some other likelihood-based criterion, her approximation is not good enough. The quadrature filter were ran with $m = 20$ abscissas in each direction in scenarios I and II, but in scenario III, where the parameters are much closer to the boundary of the stationary region, it was necessary to raise the number to $m = 50$. Computing time were for the Poisson case about 14 seconds for each likelihood evalulation on an Intel Pentium 200MHz MMX running on Linux 2.0.

A second round of experiments are presented for the TAR model (3.2) in Figure 4 under the scenarios shown in Table 2. Note that the two regimes are defined at lag $d = 2$. This is in accordance with empirical studies (Stenseth, Bjørnstad and Falck 1996). For the snowshoe hare (Scenario III') the model is the superposition of two stationary regimes, corresponding to periods of about 12 (a-regime) and 6 (b-regime) years. This reflects prevailing ecological theory, deduced from studies of the lynx, of a slow growth phase and a faster decline, but no empirical studies of this nature has as yet been undertaken for the hare itself. The results portrayed in in Figure 4, is however, encouraging and indicate that statistical studies may be able to estimate underlying nonlinear effects from categorical data.

It should be noted that for likelihood calculations, Monte Carlo estimates obtained by running the filter independently for each parameter set is probably not the thing to do. Some reduction in variability is obtained by using common random numbers, but in our situation this was not enough to give smooth enough likelihood surfaces. A more promising alternative is the simulated likelihood ratio method (Billio, Monfort and Robert 1998), in which the filter (a smoothed version) is only run for one parameter set (the null set) and the likelihood *ratio* can be estimated at all other parameter sets by ratios of full likelihoods (including the underlying state process). Neither this approach worked satisfactory in that although a smooth surface was obtained,

the bias was to high for parameter sets not close to the null set.

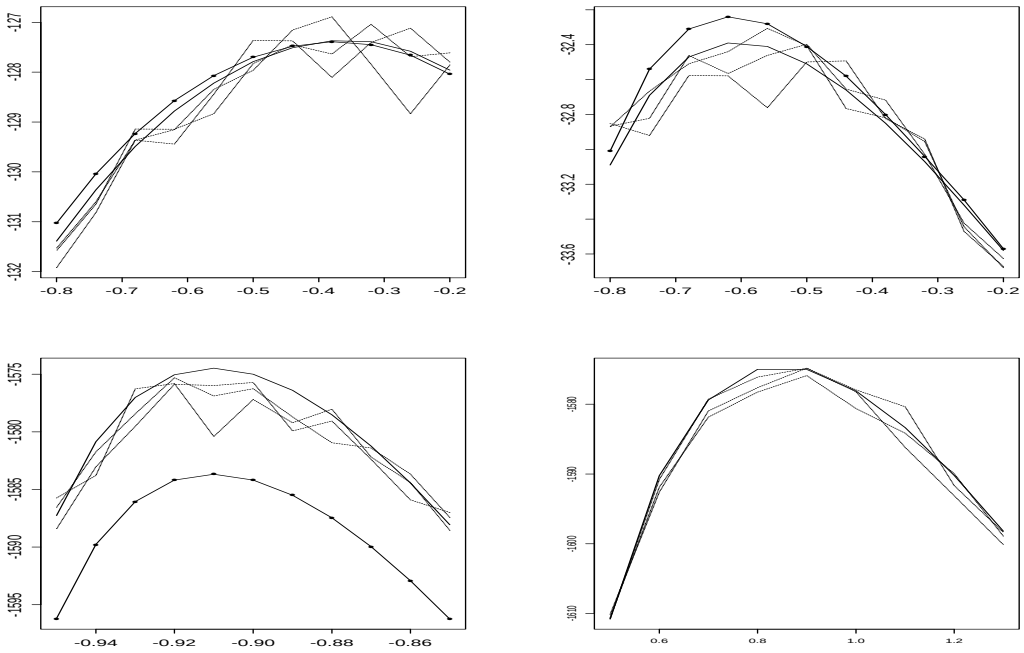

Figure 3: *Estimated profile (log-)likelihood curves for the AR(2) process with Poisson (upper left), binary (upper right) and categorical (lower left and right) observations. Scenarios are given in Table 1. The x-axis displays the a_2 values used, except for the last plot where the x-axis represents β-values. The quadrature filter estimate is given as a solid curve, the estimate based on the Schnatter filter is given as a solid curve with dots (not given in the last plot). The three dashed dotted curves are different Monte Carlo estimates.*

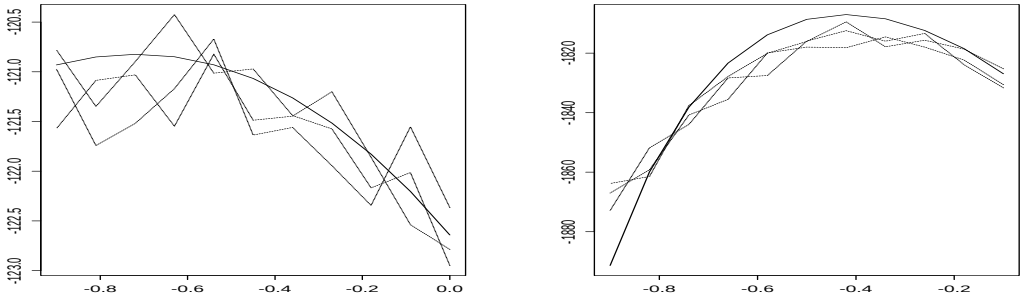

Figure 4: *Estimated profile (log-)likelihood curves for the TAR(2) process with Poisson (left) and categorical (right) observations. Scenarios are given in Table 2. The x-axis displays the a_2 (b_2 in right plot) values used. The quadrature filter estimate is given as a solid curve. The three dashed dotted curves are different Monte Carlo estimates.*

4 Concluding remarks

In this paper we have been concerned with filtering for problems with low-dimensional state vectors. In such situations, *deterministic* filters may be preferable. Some results on the errors involved are given. It is shown that the errors in likelihood calculations propagate linearly in time. However, since the errors at each step can be made very small with a moderate number of particles, the linear increase is in many situations not a problem.

Constructing quadrature filters is the question of selecting particles and their associated weights. In contrast to Monte Carlo filters, where the particles are automatically laid out in the high-probable regions, particles now have to be user-specified. Although many sophisticated designs are possible, our preference is to select the particles marginally on each variable. This result in many particles with weights close to zero, but this is outweighed by a much smaller computational burden and easier implementation. Further, for situations where the information in the observations is sparse, particles can successfully be specified directly from properties of the stationary distribution.

Deterministic filters are compared with Monte Carlo filters on some simulated data based on models representing animal populations. For these problems, likelihood-based inference is the primary concern, and large variabilities in these functions give problems if the likelihood is to be maximised. In such situations, deterministic filters are superior because of their high accuracy.

A Derivation of numerical errors

Equation (2.18) is verified by first noting that by (2.5) and (2.19)

$$\exp(\delta_t)\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \exp(\delta_t)\int p(\mathbf{x}_{t+1}|\mathbf{x}_t)\tilde{p}(\mathbf{x}_t|\mathbf{y}_{1:t})d\mathbf{x}_t + \eta(\mathbf{x}_{t+1}).$$

Hence

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)[\exp(\delta_t)\tilde{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) - p(\mathbf{x}_t|\mathbf{y}_{1:t})]d\mathbf{x}_t + \eta(\mathbf{x}_{t+1}),$$

or, by using (2.3) and (2.6)

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) =$$
$$\int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{y}_t|\mathbf{x}_t)[\exp(\delta_t)\widetilde{C}_t^{-1}\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) - C_t^{-1}p(\mathbf{x}_t|\mathbf{y}_{1:t-1})]d\mathbf{x}_t + \eta(\mathbf{x}_{t+1})$$

which reduces to (2.18).

The Proof of (2.25) starts with noting that by using (2.14), we may write

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \exp(\delta_t)\sum_{i=1}^{N_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})$$

which can be rewritten to

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \sum_{i=1}^{N_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})[\exp(\delta_t)\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - w_t^{(i)}p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t})] + \sum_{i=1}^{N_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})$$

Now, by using (2.2) and (2.14), the first term is equal to

$$\sum_{i=1}^{N_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})p(\mathbf{y}_t|\mathbf{x}_t^{(i)})w_t^{(i)}[\exp(\delta_t)\widehat{C}_t^{-1}\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) - C_t^{-1}p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1})]$$

which can be rewritten to

$$C_t^{-1}\sum_{i=1}^{N_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})p(\mathbf{y}_t|\mathbf{x}_t^{(i)})w_t^{(i)}[\exp(\delta_{t-1})\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) - p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1})]$$

by noting that $\exp(\delta_t) = \exp(\delta_{t-1})\widehat{C}_t C_t^{-1}$. The expression in the bracket is now equal to $\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})$, showing that

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = C_t^{-1}\sum_{i=1}^{N_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})p(\mathbf{y}_t|\mathbf{x}_t^{(i)})w_t^{(i)}\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1}) + \sum_{i=1}^{N_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \quad \text{(A.1)}$$

By (2.20) an upper limit for the first term on the right side of (A.1) is

$$r_t C_t^{-1}\sum_{i=1}^{N_t} w_t^{(i)}p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}),$$

which can be simplified to

$$r_t\sum_{i=1}^{N_t} w_t^{(i)}p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t})$$

using (2.2) again. By (2.23), this is less or equal to $r_t(1+\varepsilon)p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})$. By the same condition, the absolute value of the last two terms on the right hand side of (A.1) is less or equal to $\varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_t)$. Combining these upper limits gives

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \leq [r_t(1+\varepsilon)+\varepsilon]p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})$$

showing that

$$r_{t+1} \leq r_t(1+\varepsilon)+\varepsilon$$

from which (2.25) can be easily proven by recursion.

References

Bickel, P. J., Ritov, Y. and Ryden, T. (1998). Asymptotic normality of the maximum-likelihood estimator for general hidden markov models, *ANNALS OF STATISTICS* **26**(4): 1614–1635.

Billio, M., Monfort, A. and Robert, C. P. (1998). The simulated likelihood ratio (SLR) method, *Technical report*, University CaFoscari of Venice.

Bølviken, E., Glöckner, F. and Stenseth, N. C. (1999). Estimating parameters in autoregressive processes of low order from noisy, binary data, Manuscript under preparation.

Bucy, R. S. and Senne, K. D. (1971). Digital synthesis of Nonlinear Filters, *Automatica* **7**: 287–298.

Doucet, A. (1998). On sequential simulation-based method for Bayesian filtering, *Technical Report TR.310*, CUED/F-INFENG.

Kitagawa, G. (1987). Non-Gaussian state-space modelling of nonstationary time series (with discussion), *Journal of the American Statistical Association* **82**(400): 1032–1063.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-Gaussian non-linear state space models, *Journal of Computational and Graphical Statistics* **5**: 1–25.

Monahan, J. and Genz, A. (1997). Spherical-radial integration rules for Bayesian computation, *Journal of the American Statistical Association* **92**(438): 664–674.

Pole, A. and West, M. (1988). Efficient numerical integration in dynamic models, *Technical Report Warwick Research Report 136*, Dept. Statistics, Univ. Warwick.

Pole, A. and West, M. (1990). Efficient Bayesian learning in non-linear dynamic models, *J. Forecasting* **9**(2): 119–136.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992). *Numerical Recipes in C - The Arts of Scientific Computing*, second edn, Cambridge University Press.

Priestley, M. B. (1981). *Spectral Analysis and Time Series*, Probability and Mathematical Statistics, Academic Press, Inc.

Royama, T. (1992). *Analytical Population Dynamics*, Vol. 10 of *Population and community biology series*, Chapman & Hall, London.

Schnatter, S. (1992). Integration-based Kalman-filtering for a dynamic generalized linear trend model, *Computational Statistics & Data Analysis* **13**: 447–459.

Sorenson, H. W. and Alspach, D. L. (1971). Recursive Bayesian estimation using Gaussian sums, *Automatica* **7**: 465–479.

Stenseth, N. C., Bjørnstad, O. N. and Falck, W. (1996). Is spacing behaviour coupled with predation causing the microtine density cycle? A synthesis of current process-oriented and pattern-oriented studies, *Proceedings of the Royal Society of London, B* **263**: 1423–1435.

Stenseth, N. C., Bjørnstad, O. N. and Saitoh, T. (1996). A gradient from stable to cyclic populations of Clethrionomys rufocanus in Hokkaido, *Proceedings of the Royal Society of London, B* **263**: 1117–1126.

Stenseth, N. C., Falck, W., Bjørnstad, O. N. and Krebs, C. J. (1997). Population regulation in snowshoe hare and Canadian lynx: Asymmetric food web configurations between hare and lynx, *Proceedings of National Academy of Science, Washington* **94**: 5147–5152.

Stenseth, N. C., Falck, W., Chain, K. S., Bjørnstad, O. N., Donoghue, M. O. and Tong, H. (1998). From ecological patterns to ecological processes: Phase- and density-dependencies in the Canadian lynx cycle, *Proceedings of National Academy of Science, Washington* **95**: 15430–15435.

Tong, H. (1990). *Non-linear time series : A dynamical system approach*, Oxford statistical science series ; 6, Oxford : Clarendon Press,.

West, M. and Harrison, J. (1997). *Bayesian forecasting and dynamic models*, Springer Series in Statistics, second edn, Springer-Verlag, New York.