Particle filters with applications

Frydendall, Jan

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Particle filters
with Applications

Jan Frydendall

January 19, 2009

DTU Informatics

Contents

1 The state space model

The type of filtering that we are interested is state space filtering. Hence, we are given an arbitrary state space model with a set of observations. The task is now to construct a mechanism that would enable us to do estimations of the state space model parameters and reconstruction of the states. In our setting we are mostly interested in non linear state space models. The notion on linear state space models is well covered in [11]. The general form of the state space model is the following [14]

$$\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t, \mathbf{v}_t) \tag{1}$$

$$\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{w}_t), \tag{2}$$

where $\mathbf{f}_t : \mathcal{R}^n \times \mathcal{R}^m \to \mathcal{R}^n$ is the model transition operator and $\mathbf{v}_t \in \mathcal{R}^m$ is a white noise process that is not dependent on the past and current states. The $\mathbf{h}_t : \mathcal{R}^p \times \mathcal{R}^r \to \mathcal{R}^r$ is the observation operator that relates the states to the observations. The observations noise $\mathbf{w}_t \in \mathcal{R}^r$ is also and white noise process that is not dependent on the past and current states and the system noise. It is assumed that the PDF of $\mathbf{v}_t$ and $\mathbf{w}_t$ is known and that we also know the initial distribution of $p(\mathbf{x}_1|\mathbf{y}_0) = p(\mathbf{x}_0)$ together with transition and observation operator for all $t \in \{0, \cdots, T\}$. The task is to construct the PDF of the current state $\mathbf{x}_t$ given all the information avaliable to us $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. Much as we have done with Kalman filter and the Extended Kalman filter in [11, 12] we will do the construction in two steps, a prediction and a update step. Assume that we have the PDF $p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$ at time step $t-1$ we can now construct the $p(\mathbf{x}_{1:t}|\mathbf{y}_{1:t-1})$ with the transition operator,

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{1:t-1}, \tag{3}$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the transition probability that is generated from the transition model with known $\mathbf{v}_{t-1}$. The transition PDF is generated from the Markov state model (1) and the known PDF of $\mathbf{v}_{t-1}$, hence

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{v}_{t-1})p(\mathbf{v}_{t-1}|\mathbf{x}_{t-1})d\mathbf{v}_{t-1}, \tag{4}$$

by assumption we have that $p(\mathbf{v}_{t-1}) = p(\mathbf{v}_{t-1}|\mathbf{x}_{t-1})$, thus we get

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \int \delta(\mathbf{x}_t - \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})p(\mathbf{v}_{t-1})d\mathbf{v}_{t-1}, \tag{5}$$

the delta function $\delta(\cdot)$ indicates that we do not known $\mathbf{x}_{t-1}$ and $\mathbf{v}_{t-1}$ explicitly. If we did we could easily get $\mathbf{x}_t$ from (1) [14]. When the measurements becomes avaliable we can update the prior via Bayes rule [4],

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})d\mathbf{x}_t} \tag{6}$$

The likelihood PDF, $p(\mathbf{y}_t|\mathbf{x}_t)$ is defined by the observation operator and the known distribution $\mathbf{w}_t$,

$$p(\mathbf{y}_t|\mathbf{x}_t) = \int \delta(\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_t, \mathbf{w}_t))p(\mathbf{w}_t)d\mathbf{w}_t. \tag{7}$$

The equation (6) is used to update the prediction prior (3) when new measurements $\mathbf{y}_t$ become avaliable. This is required if we want to obtain the posterior of the states. The equation (3) and (6) is the solution of the Bayes recursive estimation problem [14]. The only known analytical solution to these equations is the Kalman filter, if we assume that the our state space model is linear and the noise process are normal distributed.

An example of a nonlinare state space model is

$$x_t = 0.5x_{t-1} + 25\frac{x_{t-1}}{1 + x_{t-1}^2} + 8cos(1.2(t-1)) + v_t \tag{8}$$

$$y_t = \frac{x_t^2}{20} + w_t, \tag{9}$$

where $v_t \sim \mathcal{N}(0, \sigma_v^2)$ and $w_t \sim \mathcal{N}(0, \sigma_w^2)$ are normal distributed PDF with known variance. This model is widely used in literature [16, 14, 4],when it comes to particle filtering. In our example we will use the following initial conditions, $\sigma_v^2 = 10$, $\sigma_w^2 = 1$ and $x_0 \sim \mathcal{N}(0, 10)$. In figure 1 is the state space model represented.

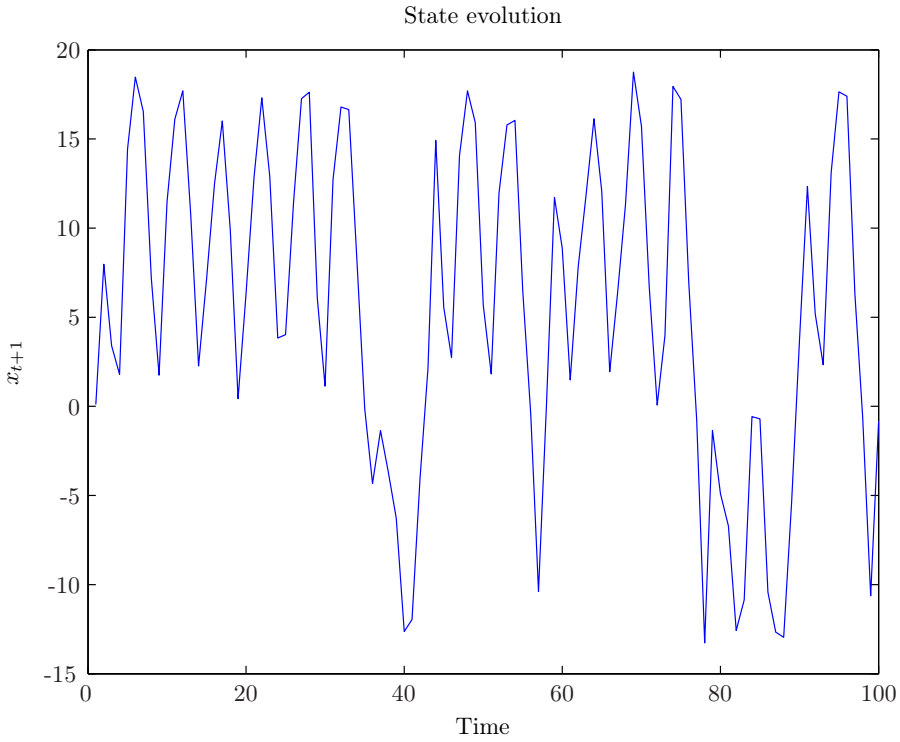

Figure 1: The simulate states of the model (8)

2 The preliminaries

In order to take full advantage of the particle filters, we have to address the two most important key elements in the particle filter cocktail. Namely the Monte Carlo interpretation and the Importance sampling steps. First we will address the Monte Carlo requirement.

2.1 Perfect Monte Carlo Simulation

To put Monte Carlo simulation in lay man terms; Monte Carlo simulation is just a another way of interprete integrals as sums. However, this loose definition will not do us any good mathematically. In order to get a rigorous mathematical definition we assume the following. Assume that we have some function $f(x) \in \mathcal{R}$ that we would like to know the expectation. We use the definition of the expectation by assuming that we have a given probability distribution function (pdf) $p(x, y) \in \mathcal{R}$. The expectation can be written as

$$E[f(x)] = \int_{-\infty}^{\infty} f(x)p(x, y)dx, \tag{10}$$

most often the integral can not be solve analytically and therefore we most find another way. One way is the Monte Carlo simulation, Lets assume that we have N independent and identically

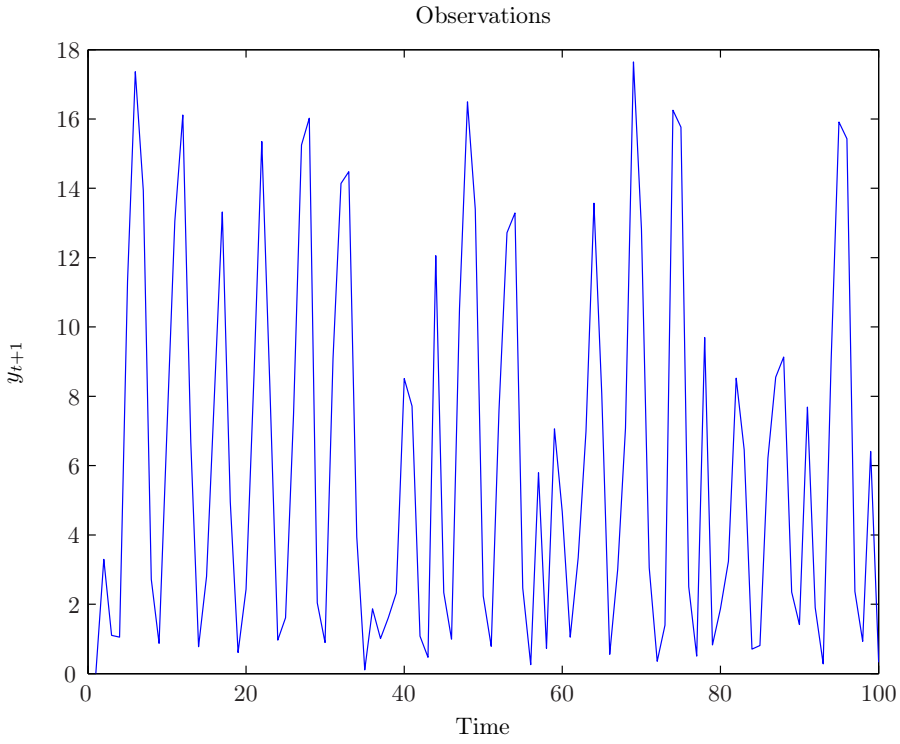

Figure 2: The observation from the state evolution through the observation operator (9)

distributed (i.i.d.) random samples from $\{\mathbf{x}_{0:t}^{(i)}; i = 1, \ldots, N\}$ according to our pdf $p(\mathbf{x}_{0:t}, \mathbf{y}_{1:t})$. As in the book by [4] we will define the empirical distribution as a sum of dirac delta functions,

$$p_N(d\mathbf{x}_{0:t}, \mathbf{y}_{0:t}) = \sum_{i=1}^{N} \delta_{\mathbf{x}_{0:t}^i}(d\mathbf{x}_{0:t}), \tag{11}$$

where $\delta_{\mathbf{x}_{0:t}^i}(d\mathbf{x}_{0:t})$ is the probability mass located in $\mathbf{x}_{0:t}^i$. The $d\mathbf{x}_{0:t}$ just the denotes that we are in continuous formulation and that we can not specify the exact location of the probability mass, so we only specify the probability mass in a vicinity of $\mathbf{x}_{0:t}^i$ by make a small sphere around it. With this definition we able to estimate the expectation (10) as

$$\overline{E[\mathbf{f}_t(\mathbf{x}_{0:t})]} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{f}(\mathbf{x}_{0:t}^i), \tag{12}$$

where $\mathbf{f}_t$ is the function to be estimated and for generality we have made it time depending and multidimensional. According to law of large numbers, the expectation (12) will converge almost surely to (10), i.e $\overline{E[\mathbf{f}_t(\mathbf{x}_{0:t})]} \xrightarrow[N\to\infty]{a.s.} E[\mathbf{f}_t(\mathbf{x}_{0:t})]$ and if the posterior variance of $\mathbf{f}_t(\mathbf{x}_{0:t})$ is bounded i.e. $\sigma_{\mathbf{f}_t}^2 < \infty$ we then have from the central limit theorem that [15, 4]

$$\sqrt{N}\left(\overline{E[\mathbf{f}_t(\mathbf{x}_{0:t})]} - E[\mathbf{f}_t(\mathbf{x}_{0:t})]\right) \underset{N\to\infty}{\Longrightarrow} \mathcal{N}(0, \sigma_{f_t}^2),$$

where $\underset{N\to\infty}{\Longrightarrow}$ denotes converges in distribution.

2.2 Bayesian Importance Sampling

Here we will develop the idea of Importance sampling, which in short is to find a proper proposal distribution by considering a clever scaling distribution. Consider again the expectation of f_t

$$\begin{aligned} E[\mathbf{f}_t(\mathbf{x}_{0:t})] &= \int \mathbf{f}_t(\mathbf{x}_{0:t}) \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\ &= \int \mathbf{f}_t(\mathbf{x}_{0:t}) \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\ &= \int \mathbf{f}_t(\mathbf{x}_{0:t}) \frac{\omega_t(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}, \end{aligned}$$

where we $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ is the new proposal distribution and $\omega_t(\mathbf{x}_{0:t})$ is the importance weights

$$\omega_t(\mathbf{x}_{0:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}. \tag{13}$$

The distribution $p(\mathbf{y}_{1:t})$ is not depending on $\mathbf{x}_{0:t}$ and therefore it can be brought outside the integral. We can again use that the distribution is the marginalization of $p(\mathbf{y}_{1:t}) = \int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})d\mathbf{x}_{0:t}$ and rewrite the expectation

$$\begin{aligned} E[\mathbf{f}_t(\mathbf{x}_{0:t})] &= \frac{1}{p(\mathbf{y}_{1:t})} \int \mathbf{f}_t(\mathbf{x}_{0:t})\omega_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\ &= \frac{\int \mathbf{f}_t(\mathbf{x}_{0:t})\omega_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t}) \frac{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} d\mathbf{x}_{0:t}} \\ &= \frac{\int \mathbf{f}_t(\mathbf{x}_{0:t})\omega_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}}{\int \omega_t(\mathbf{x}_{0:t}) q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t}} \\ &= \frac{E_{q(\cdot|\mathbf{y}_{1:t})}(\omega_t(\mathbf{x}_{0:t})\mathbf{f}_t(\mathbf{x}_{0:t}))}{E_{q(\cdot|\mathbf{y}_{1:t})}(\omega_t(\mathbf{x}_{0:t}))}, \end{aligned}$$

where $E_{q(\cdot|\mathbf{y}_{1:t})}$ denotes the expectation of $\omega_t(\mathbf{x}_{0:t})$ with respect to the proposal distribution $q(\cdot|\mathbf{y}_{1:t})$. With this in mind we can now express the expectation as

$$E[\mathbf{f}_t(\mathbf{x}_{0:t})] = \frac{\frac{1}{N}\sum_{i=1}^N \mathbf{f}_t(\mathbf{x}^i_{0:t})\omega_t(\mathbf{x}^i_{0:t})}{\frac{1}{N}\sum_{i=1}^N \omega_t(\mathbf{x}^i_{0:t})} \tag{14}$$

$$= \sum_{i=1}^N \mathbf{f}_t(\mathbf{x}^i_{0:t})\tilde{\omega}_t(\mathbf{x}^i_{0:t}), \tag{15}$$

where $\tilde{\omega}^i_t$ are the normalized importance weights,

$$\tilde{\omega}^i_t = \frac{\omega^i_t}{\sum_{j=1}^N \omega^j_t}. \tag{16}$$

The expectation (14) is biased as long as N is finite. However, from the law of strong numbers the estimate is asymptotically unbiased, for a good discussion of this look in [6]. As N tens to infinity, the posterior density function can be approximated arbitrarily well by the point-mass estimate.

$$\hat{p}(d\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^N \tilde{\omega}_t(\mathbf{x}^i_{0:t})\delta_{\mathbf{x}^i_{0:t}}(d\mathbf{x}_{0:t}). \tag{17}$$

3 Particle Filters

After all the sung and dance in the previous section we now able to formulate the particle filters. The first an most straight forward particle filter is the one called the Sequential Importance Sampling (SIS). The idea is to draw samples from our prior distribution and then assign an importance weights to each particle at every time step. Put into a more formal frame, we first have to make some assumptions on the Importance weights. In order to sample from the proposal distribution $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, we have to assume that we can compute the sequential estimate of the posterior distribution at time t without modifying the previously simulate states $\mathbf{x}_{0:t-1}$ [15]. Thus, the following distribution can be used

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}). \tag{18}$$

Hence the current state is independent of the future observations. Along with the Markov properties of the states

$$p(\mathbf{x}_{0:t}) = p(\mathbf{x}_0)\prod_{j=1}^{t} p(\mathbf{x}_j|\mathbf{x}_{j-1}) \tag{19}$$

and that the observations are conditionally independent given the states

$$p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) = \prod_{j=1}^{t} p(\mathbf{y}_j|\mathbf{x}_j) \tag{20}$$

If we now apply the above assumption onto the Importance weights (13) we can establish the following recursive importance weights

$$\begin{aligned}\omega_t &= \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \\ &= \omega_{t-1}\frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})}\frac{1}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})} \\ &= \omega_{t-1}\frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})}\end{aligned} \tag{21}$$

With the above recursive importance weights we can now at each time step generate a new importance weight, i.e. a new state reconstruction, since we are able to calculate the likehood $p(\mathbf{y}_t|\mathbf{x}_t)$ and the transition probabilities $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, given that we have a prober proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$.

Until now we have just stated that the proposal weights are something that is essential in the this estimation approach. However, finding the prober proposal distribution is a very difficult task. Much of the current research is devoted to finding prober proposal distribution or avoiding them.

3.1 Proposal distributions

The most critical ingredient in the particle filters that uses importance sampling are the proposal distributions. The foremost property of the proposal distribution is to minimize the variance of the importance weights conditional on $\mathbf{x}_{0:t-1}$ and $\mathbf{y}_{1:t}$ [5]. The prober choice is then the distribution $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$, that minimizes the variance of the importance weights. However, this result does not do much for us, it is only of theoretical interest. Among practitioners the following proposal distribution is popular [14]

$$q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{22}$$

where the proposal distribution now is distributed as the transition prior. Although we are given up the requirement of the minimizing variance of the importance weights, we have on the other

hand got a proposal distribution which most easier to implement and to sample from. If we substitute the (22) into the (21) we get a very nice result

$$\omega_t = w_{t-1}\frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{x}_t|\mathbf{x}_{t-1})} = w_{t-1}p(\mathbf{y}_t|\mathbf{x}_t). \tag{23}$$

In the next section we will give some examples of the SIS and the other variant called the SIR.

4 SIS

If we use the idea of importance sampling (23) from the previous section we can formulate the two basic particle filters. The first filter is the SIS filter and the generic algorithm is shown below in algorithm (1).

1: **procedure** SIS
2: **Initialization,** $t = 0$
For $i = 1, \cdots, N$, sample $\mathbf{x}_0^i \sim p(\mathbf{x}_0)$ and set $t = 1$.
3: **Importance sampling step**
For $i = 1, \cdots, N$, sample $\mathbf{x}_t^i \sim p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and set $\tilde{\mathbf{x}}_{0:t}^i = \left(\mathbf{x}_{0:t-1}^i, \tilde{\mathbf{x}}_t\right)$.
For $i = 1, \cdots, N$, evaluate the importance weights $\omega_t = p(\mathbf{y}_t|\mathbf{x}_t)$
Normalize the importance weights.
Set $t \rightarrow t + 1$ and go to step 3
4: **end procedure**

Algoritme 1: The generic SIS algorithm

When we use the transition prior as proposal distribution the filters are often called *Bootstrap* filters.

The SIS filters has one major problem that over shadows everything else. The key idea with the SIS filters is to select the particles that have the most probable outcome compared to the observation given the predictions. With this algorithm we are thinning out the particles that do not represent the state and observations. When this thinning is applied at every time step we more or less thin out all particles except one. This problem is known as degeneracy and can be illustrated in the figure (3). Put in more mathematically formalism we say that the variance of the weights increases over time. If we again look at the definition of the importance (13) weights we have,

$$\begin{aligned}\omega_t(\mathbf{x}_{0:t}) &= \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \\ &= \frac{p(\mathbf{y}_{1:t}, \mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \\ &= \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})p(\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \\ &\propto \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})},\end{aligned} \tag{24}$$

where we have used that $p(\mathbf{y}_{1:t})$ is a constant. The ratio (24) is called the importance ratio and it can be shown that the variance of this ration will increase over time. This has been done by ([10, 5]). Following the argument from [15] we want to have that the proposal and the posterior density to be close, i.e. that is the proposal distribution has full support over the true posterior density. Taking the expectation of the importance ratio with respect to the proposal distribution

$$E_{q(\cdot|\mathbf{y}_{1:t})}\left(\frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}\right) = 1 \tag{25}$$

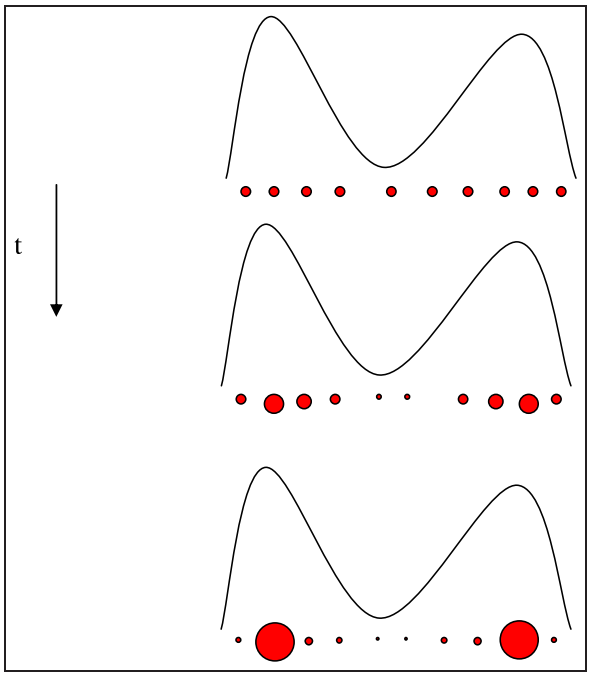

Figure 3: The degeneracy problem of the SIS filters. At each time step the most probable particles are given an weight that represents the probability that prediction is close to the observations.

and the variance

$$var_{q(\cdot|\mathbf{y}_{1:t})}\left(\frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}\right) = E_{q(\cdot|\mathbf{y}_{1:t})}\left(\left(\frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} - 1\right)^2\right), \tag{26}$$

this gives an indication of that we want the variance of the importance weights to be close to zero in order to get good estimates. However, when the variance of the weights increase over time we get inaccurate estimations.

5 SIR

When doing filtration with SIS algorithm one has to start out with a very large numbers of particle in order make sure that some of the particle survives. In order to reduce the particle set size and thereby reducing the computational time, we will consider the SIR filter. The idea with the SIR filter is to resample the particles at each time step. We still uses the importance sampling algorithm, however, after we have assign weights to the particles will only keep the particles that have more weight then $1/N$ and then resample the particles that has less weights from the surviving particles.

Assume that to each particle $\mathbf{x}_{0:t}^i$ we can assign an weight $N_i \in \mathcal{N}$ such that $\sum_{i=1}^N N_i = N$ and that we can rewrite (17) as

$$P_N(d\mathbf{x}_{0:t} \mid \mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} N_t^{(i)} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t}) \tag{27}$$

The above assumption leads to the following interpretation of the weights. At each time step we assign the particles importance weights and then we resample the particles such that after the resampling step all the particles will have equal probability $1/N$. We do not change the number of

particles in the set we only discard the ones that are unlikely and multiply the ones that survives such that the total number of particles are N. The SIR algorithm can also be written in a generic algorithm

1: **procedure** SIR
2: **Initialization,** $t = 0$
For $i = 1, \cdots, N$, sample $\mathbf{x}_0^i \sim p(\mathbf{x}_0)$ and set $t = 1$.
3: **Importance sampling step**
For $i = 1, \cdots, N$, sample $\mathbf{x}_t^i \sim p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and set $\tilde{\mathbf{x}}_{0:t}^i = \left(\mathbf{x}_{0:t-1}^i, \tilde{\mathbf{x}}_t\right)$.
For $i = 1, \cdots, N$, evaluate the importance weights $\omega_t = p(\mathbf{y}_t|\mathbf{x}_t)$
Normalize the importance weights.
4: **Resampling step**
Multiply/Suppress samples $\tilde{\mathbf{x}}_{0:t}^i$ with high/low importance weights $\tilde{\omega}_t^i$, respectively, to obtain N random samples $\mathbf{x}_{0:t}^i$ approximately distributed to $p(\mathbf{y}_t|\mathbf{x}_t)$
For $i = 1, \cdots, N$ set $\omega_t^i = \tilde{\omega}_t^i = \frac{1}{N}$
Set $t \to t + 1$ and go to step 3
5: **end procedure**

Algoritme 2: The generic SIR algorithm

5.1 Resampling

The selection of the particles and the resampling step can be done in many ways. Here we only discuss the very basic resampling algorithm, namely the multinomial sampling. We want to find a mapping from $\{\mathbf{x}_{0:t}^i, \tilde{\omega}_t^i\} \to \{\mathbf{x}_{0:t}^i, N^{-1}\}$. The mapping is given in [14] and can be written as

$$\sum_{j=1}^{M-1} \tilde{\omega}_t^j < u_i \leq \sum_{j=1}^{M} \tilde{\omega}_t^j, \tag{28}$$

where u_i is a random sampling from the uniform distribution $\mathcal{U}(0, 1]$. This procedure is repeated for $i = 1, \cdots, N$. This algorithm can be written in generic form

1: **procedure** MULTINOMIAL
2: For $i = 1, \cdots, N$, sample $u^i \sim \mathcal{U}(0, 1]$
3: calculate the discrete c.d.f. of u_i and $\tilde{\omega}_t^i$
4: set $j = 1$ and $i = 1$
5: **while** $j \leq N$ **do**
6: **if** c.d.f.(u_i) - c.d.f.$(\tilde{\omega}_t^i) \leq 0$ **then**
7: $idx(j) = i$
8: $j = j + 1$
9: **else**
10: $i = i + 1$
11: **end if**
12: **end while**
13: **end procedure**

Algoritme 3: The generic multinomial sampling algorithm

where $idx(i)$ is the index function. The graphical interpretation of the algorithm is given in figure (4).

The c.d.f. of u_i is seen as the strait line in the figure (4) and the step line is the c.d.f. of $\tilde{\omega}_t^i$. At the top of the figure is the particle weights shown. When the particles have great mass then the step

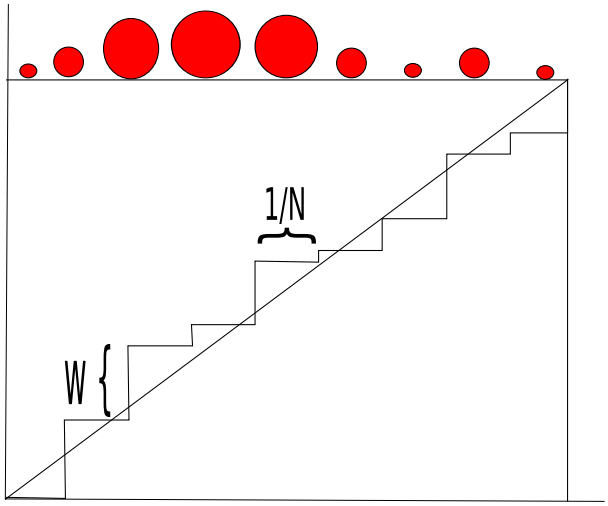

Figure 4: The graphical interpretation of the algorithm (3).

function is above the strait line and the particles are preserved and multiply to the locations where the particles have little mass. When the particles have small mass the step function is below the strait line and the particular particle is killed and gets resampled from one of the particles that have great mass. There exist of corse more resampling algorithms in literature see for example [3, 7] for a survey of the most common used.

6 Application

In this section we will give examples of the particle filters. We will use the state space model (1) and (2) as the test bed of the filtration. First we will try the SIS filter. In the filtration we will use 500 particles and the noise is assumed to be normal distributed. Thus, the model noise $Q \sim \mathcal{N}(0, 10)$ and the observation noise $R \sim \mathcal{N}(0, 1)$. The observation operator H is given as $H = y^2$, where y is the input from the predictions. Hence, the filter does not know explicitly if the sign of the states and therefore around zero the filter will have trouble with the estimation since the there is no indication of the dynamics of the state. The likelihood function $p(\mathbf{y_t}|\mathbf{x_t})$ will be given as an Gaussian bell

$$\omega_t^i = \frac{\exp(-0.5((\mathbf{y}_t^i - \mathbf{H}\mathbf{x}_t^i)^T\mathbf{R}^{-1}(\mathbf{y}_t^i - \mathbf{H}\mathbf{x}_t^i))}{\sum_{j=1}^N \exp(-0.5((\mathbf{y}_t^j - \mathbf{H}\mathbf{x}_t^j)^T\mathbf{R}^{-1}(\mathbf{y}_t^j - \mathbf{H}\mathbf{x}_t^j)))}. \tag{29}$$

In the figure (5) the reconstruction of the states are shown and below is the scatter plot of the reconstructed states and the observations. It is not obvious that this filtration is very good since from the plot we can see that the reconstruction is following the observations and the scatter plot is also close to the strait line. However, when we consider the the effective particle size i.e. the inverse of the variance of the importance weights,figure (6).

We can see that the particle set is reduced to effectively one single particle. This means that the filtration is only consisting of one realization of the process and therefore we can not assume that we will converge to the true states. This can also be seen from the density plot in the same figure. The densities starts out as a concentration around the prior, however, as the simulation continues the particle sizes is reduced and the densities get smoother out and will not carry any useful information trough to the prediction.

As mention earlier, one way of taking the degeneracy into account is to include a resampling step after each importance sampling step to keep the particles alive.

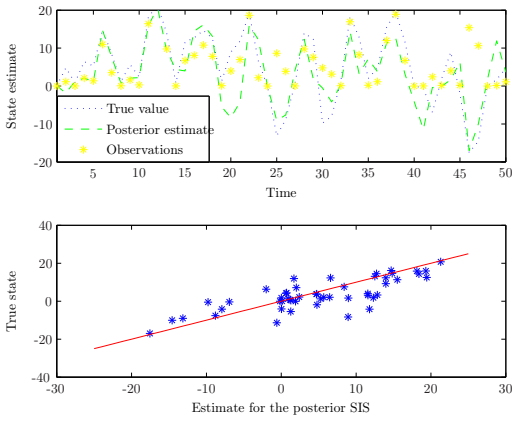

Figure 5: The reconstruction of the states with the SIS algorithm

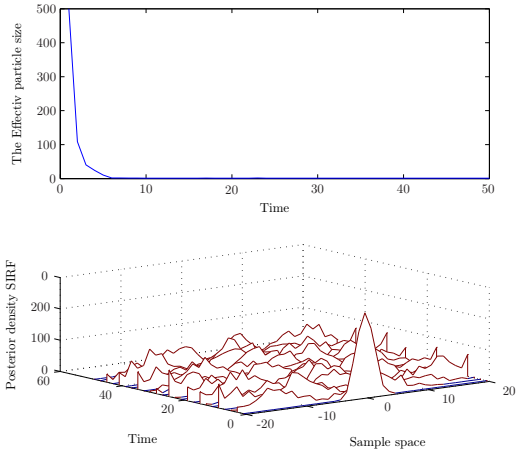

Figure 6: The effective particle size and the posterior densities from the SIS estimation

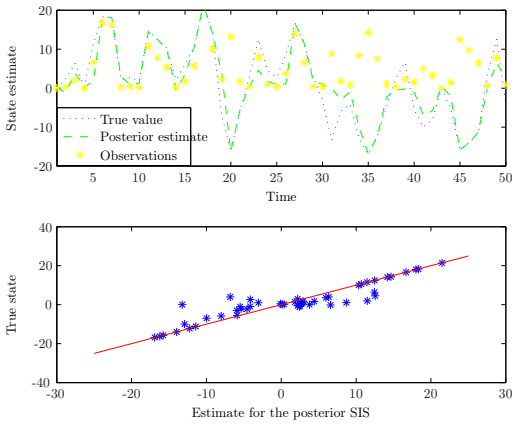

Figure 7: The reconstruction of the states with the resampling at every time step

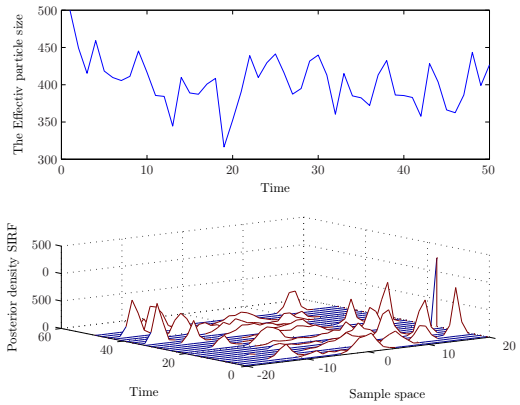

Figure 8: The effective particle size and the posterior densities with the resampling of the particles

In the figure (7) we can see that the reconstruction is better then the one from (5). (We are using the same seed in the random number generator). Remember that the observations in the figures are generated trough the observation operator and therefore we can not compared them directly to the states.

Looking at the effective particle size figure (8) we can that the resampling step is keeping the particle set well stirred and the all the particles carry information. This is also evident from the posterior density plot. We see that since we have a much effective particle set the densities will carry information thorough to the next prediction step. In this plot it is also clear that the posterior densities are not Gaussians and the strength of the particle filters should be obvious compared to the Extended Kalman Filter.

However, the resampling step is increasing the variance of the posterior estimates and therefore it should not be performed unless it is necessary. Therefore it is suggested that we use a form of threshold sampling for example $N_eff \leq \frac{2}{3}N$, where $N_eff = (\sum_{j=1}^{N}(\omega_t^j)^2)^{-1}$. This means that we use the SIS filter at every time and when the effective particle size is reduced to less then the threshold then we perform the resampling step. Put into the generic algorithm form The

1: **procedure** SIS WITH THRESHOLD SAMPLING
2: **Initialization,** $t = 0$
For $i = 1, \cdots, N$, sample $\mathbf{x}_0^i \sim p(\mathbf{x}_0)$ and set $t = 1$.
3: **Importance sampling step**
For $i = 1, \cdots, N$, sample $\mathbf{x}_t^i \sim p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and set $\tilde{\mathbf{x}}_{0:t}^i = \left(\mathbf{x}_{0:t-1}^i, \tilde{\mathbf{x}}_t\right)$.
For $i = 1, \cdots, N$, evaluate the importance weights $\omega_t = p(\mathbf{y}_t|\mathbf{x}_t)$
Normalize the importance weights.
4: **if** $N_eff \leq \beta N$ **then**
5: **Perform the resampling step**
6: Multiply/Suppress samples $\tilde{\mathbf{x}}_{0:t}^i$ with high/low importance weights $\tilde{\omega}_t^i$, respectively, to obtain N random samples $\mathbf{x}_{0:t}^i$ approximately distributed to $p(\mathbf{y}_t|\mathbf{x}_t)$
For $i = 1, \cdots, N$ set $\omega_t^i = \tilde{\omega}_t^i = \frac{1}{N}$
7: **end if**
Set $t \rightarrow t + 1$ and go to step 3
8: **end procedure**

Algoritme 4: The generic SIS with threshold sampling algorithm

estimation with SIS with threshold sampling gives the same result as with SIR. However, we have made sure that with this approach has some sort of variance reduction on the posterior estimates. The estimation with SIS with threshold sampling can be seen in the figures (9) and (10).

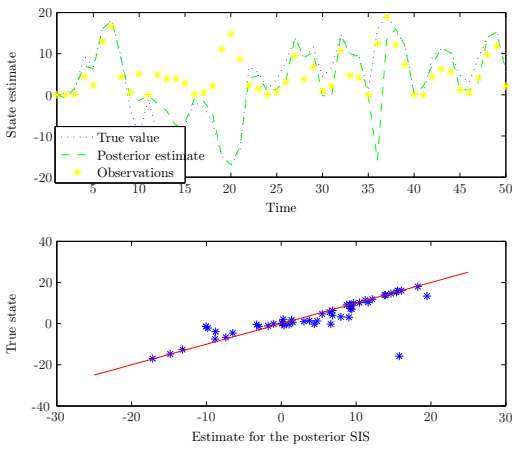

Figure 9: The reconstruction of the states with the threshold sampling

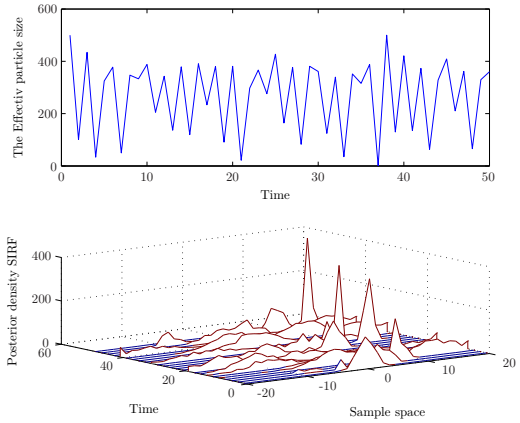

Figure 10: The effective particle size and the posterior densities from the threshold sampling

The final comparison is done with run two independent filtration with $N = 250$ and $N = 500$ with same noise assumptions. In table (6) the mean and the variance and the root mean squared error of the 1-step prediction error is shown. (Runs with higher particle numbers has also been conducted, however, results showed that the estimation had converge i.e. there was no improvement in the bias, variance and rms).

Final words on importance particle filtering, the trouble with particle filters is that on never know how many particles to use. In our simple state space model we only had to reconstructed one state, however, if the state space model goes to higher dimensions the number of particles could very well increase by a factor 1000. There are many reference in litterateur on how to reduce the particle size and how to implement them. The auxiliary SIR and the kalman filter hybrids as being the most common. However, when we straying away from the Kalman Filter and are trying with other non-parametric filters, we should be very aware of the *no free lunch* theorem. The theorem states that may the SIR filter is better on this type of model that we have uses

$N = 1000$	SIS		SIR		$Neff \leq 2/3N$	
N	250	500	250	500	250	500
mean	-2.133	-1.676	-0.514	-1.012	-1.008	-0.821
var	120.086	34.556	32.068	10.935	18.430	9.454
rms	11.164	6.113	6.051	3.708	4.7939	3.512

Table 1: The statistics of two simulation with particle filters, for respectively $N = 250$ and $N = 500$

here, however, could very well be that is the other way around with another model [2]. With this in mind the particle filters is very easy and a very good estimator for non-linear models. With growing computational power the particle filters are getting better and better.

7 Particle smoothers

In this section the notion of smoothing will be investigated. In this section two particle smoothers will be discussed.

1. The Forward-Backward Smoother (FBS)
2. The two filter smoother (TFS)

7.1 The Forward-Backward Smoother

The FBS is the simplest to construct it relies on a forward filtering in time up till the desired time to obtain the marginal distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. Then a backward sweep is thourgh the data set is conducted to modify the importance weights so that they now represent the smoothed distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. The algorithm for the FBS is simple and intuitive However, the FBS relies on that the filter distribution has support where the smoothed density is significant [9].

The quest is to construct the marginal distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ from the forward filter and the backward recursion.

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:T}) &= \int p(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{y}_{1:T}) d\mathbf{x}_{t+1} \\ &= \int p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T}) p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}_{1:t}) d\mathbf{x}_{t+1} \\ &= p(\mathbf{x}_t|\mathbf{y}_{1:t}) \int \frac{p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T}) p(\mathbf{x}_{t+1}|\mathbf{x}_t)}{\int p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t}) d\mathbf{x}_t} d\mathbf{x}_{t+1}, \end{aligned} \tag{30}$$

where

- $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is the filtered density
- $p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})$ is the smoothed density
- $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ is the dynamics of the model
- $\int p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t}) d\mathbf{x}_t$ is the state prediction

The recursion is approximated in the usual way by defining the an empirical distribution [9]

$$\hat{p}(d\mathbf{x}_t|\mathbf{y}_{1:T}) = \sum_{i=1}^{N} \omega_{t|T}^i \delta_{\mathbf{x}_t^i} d\mathbf{x}_t \tag{31}$$

and inserted into (30) thus we get the empirical approximation of (30)

$$\omega_{t|T}^i = \omega_{t+1|T}^i \left[\sum_{j=1}^{N} \omega_{t+1|T}^j \frac{p(\mathbf{x}_{t+1}^j|\mathbf{x}_t^i)}{\sum_{k=1}^{N} \omega_t^k p(\mathbf{x}_{t+1}^j|\mathbf{x}_t^k)} \right]. \tag{32}$$

The (32) is of the order $\mathcal{O}(N^2)$ by noticing that the denominator can be calculated inpendently from i for each j. The algorithm for the FBS can be seen in algorithm 5 In figure (7.1) the marginal distribution of the FBS is shown. The difference between the forward filter and the FBS is not significant. The figure also underlines the problem with FBS. The smoother can not change the support of the particles if there located in the wrong part of the state space. No new information is added to the smoohter, so the only option is to relocate the partices in the given support.

1: **procedure** FBS
2: *Filtering* For $t = 1, \cdots, T$, perform the particle filtering to obtain the weighted measure $\{\mathbf{x}_t^i, \omega_t^i\}_{i=1}^N$
3: *Initialization* For $i = 1, \cdots, N$, set $\omega_{T|T}^i = \omega_T^i$
4: *Backward recursion* For $t = T-1, \cdots, 1$ and $i = 1, \cdots, N$ evaluate
$$\omega_{t|T}^i = \omega_{t+1|T}^i \left[\sum_{j=1}^N \omega_{t+1|T}^j \frac{p(\mathbf{x}_{t+1}^j | \mathbf{x}_t^i)}{\sum_{k=1}^N \omega_t^k p(\mathbf{x}_{t+1}^j | \mathbf{x}_t^k)} \right]$$
5: **end procedure**

Algoritme 5: The generic Forward-Backward smoother

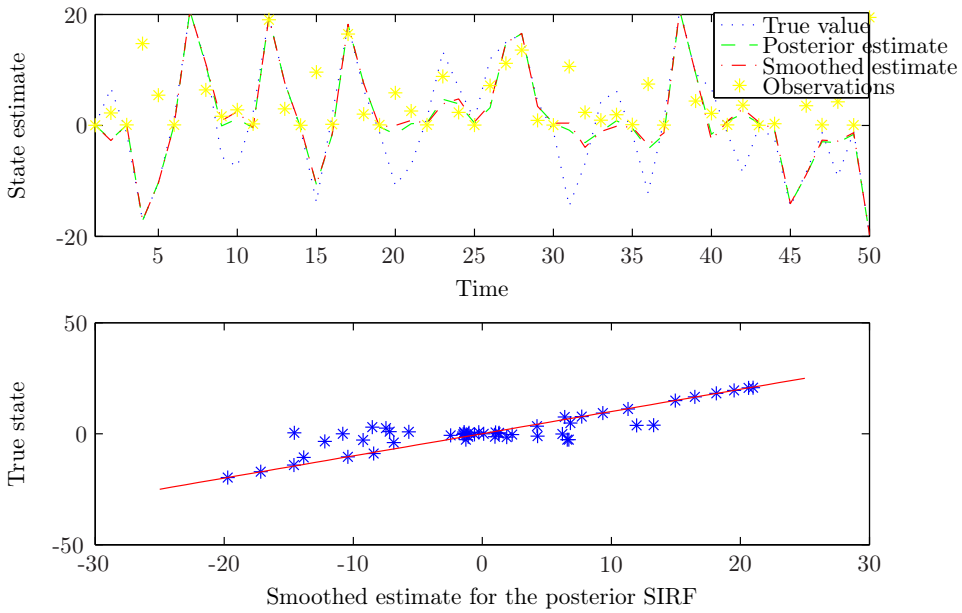

Figure 11: The smoothed estimate of the FBS and the scatter plot of the truth vs. the smoothed estimate

7.2 The two filter smoother

The marginal smoothed posterior distribution can be computed by combining the output of tow independent filters [1]. The two filters that is need is first the normal particle filter and a filter that runs backward in time. The normal particle filter is just any one of the SMC filters that calculates $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. The backward filter our barckward information filters is due to [13] and dates back to 1966. The backward filter calculates $p(\mathbf{y}_{t:T}|\mathbf{x}_t)$ backward in time. Combining the forward and backward filter we can obtain the smoothed marginal distribution $p(\mathbf{x}_t|\mathbf{y}_{1:T})$, hence

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:T}) &= p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \mathbf{y}_{t:T}) \\ &= \frac{p(\mathbf{x}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{t:T}|\mathbf{y}_{1:t-1}), \mathbf{x}_t)}{p(\mathbf{y}_{t:T}|\mathbf{y}_{1:t-1})} \\ &\propto p(\mathbf{x}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{t:T}|\mathbf{x}_t) \\ &\propto p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{y}_{t:T}|\mathbf{x}_t) \end{aligned} \tag{33}$$

The last line of (33) is the TFS the first density is the bayesian filter and the second density is the backward filter. This definition the of the TFS the normal forward filtering backward smoothing is now reduced to a pure filtration assumption. The backward filter can be calculated trough the

backward information filter [13]

$$p(\mathbf{y}_{t:T}|\mathbf{x}_t) = \int p(\mathbf{y}_{t+1:T}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{y}_t|\mathbf{x}_t)d\mathbf{x}_{t+1} \tag{34}$$

The problem with $p(\mathbf{y}_{t:T}|\mathbf{x}_t)$ is that is not a proper probability density in argument $\mathbf{x}_t$ and therefore the integral over $\mathbf{x}_t$ might not be finite [9]. Therefore normal sequential Monte Carlo approximation can not be used without making unrealistic assumptions, such as assuming $p(\mathbf{y}_{t:T}|\mathbf{x}_t) < \infty$ [8]. In order to make the TFS work for an arbitrary model the assumption derived in [1] and [9] will be used.

As state above the $p(\mathbf{y}_{t:T}|\mathbf{x}_t)$ is not a probability measure and therefore we can not apply the usually Monte Carlo methods since the constraint of these methods are that the probability densities have be finite. However, by introducing a clever artificial distribution over $\mathbf{x}_t$ with density $\gamma(\mathbf{x}_t)$ will ensure that the integral will be finite.

7.2.1 Artificial distribution

Let $\{\gamma_t(\mathbf{x}_t)\}$ be a sequence of probability distributions for $t = 1, \cdots, T$ such that for

$$p(\mathbf{y}_{t:T}|\mathbf{x}_t) > 0 \Rightarrow \gamma_t(\mathbf{x}_t) > 0 \tag{35}$$

and for the case $t = T$

$$\tilde{p}(\mathbf{x}_T|\mathbf{y}_t) = \frac{p(\mathbf{y}_T|\mathbf{x}_t)\gamma_T(\mathbf{x}_T)}{\int p(\mathbf{y}_T|\mathbf{x}_t)\gamma_T(\mathbf{x}_T)d\mathbf{x}_T}. \tag{36}$$

and for the case $t = \{2, \cdots, T-1\}$

$$\tilde{p}(\mathbf{x}_T|\mathbf{y}_t) = \frac{\gamma_T(\mathbf{x}_t)\prod_{i=t+1}^{T} p(\mathbf{x}_{i+1}|\mathbf{x}_i)\prod_{i=t}^{T} p(\mathbf{y}_i|\mathbf{x}_i)}{\int\cdots\int \gamma_T(\mathbf{x}_t)\prod_{i=t+1}^{T} p(\mathbf{x}_{i+1}|\mathbf{x}_i)\prod_{i=t}^{T} p(\mathbf{y}_i|\mathbf{x}_i)d\mathbf{x}_{t:T}} \tag{37}$$

Thus the general case for $t = \{1, \cdots, T\}$

$$p(\mathbf{y}_{t:T}|\mathbf{x}_t) = \tilde{p}(\mathbf{y}_{t:T})\frac{\tilde{p}(\mathbf{x}_T|\mathbf{y}_t)}{\gamma_T(\mathbf{x}_t)}, \tag{38}$$

where

$$\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t:T}) = \int\cdots\int \tilde{p}(\mathbf{x}_{t:T}|\mathbf{y}_{t:T})d\mathbf{x}_{t:T}. \tag{39}$$

The proof for $t = T$ follows strait from the definition. For $t = \{1, \cdots, T-1\}$

$$\begin{aligned}
p(\mathbf{y}_{t:T}|\mathbf{x}_t) &= \int\cdots\int p(\mathbf{y}_{t:T}, \mathbf{x}_{t+1:T}|\mathbf{x}_t)d\mathbf{x}_{t+1:T} \\
&= \int\cdots\int p(\mathbf{x}_{t+1:T}|\mathbf{x}_t)p(\mathbf{y}_{t:T}, \mathbf{x}_{t:T})d\mathbf{x}_{t+1:T} \\
&= \int\cdots\int \prod_{i=t+1}^{T} p(\mathbf{x}_{i+1}|\mathbf{x}_i)\prod_{i=t}^{T} p(\mathbf{y}_i|\mathbf{x}_i)d\mathbf{x}_{t+1:T} \\
&= \int\cdots\int \frac{\gamma_T(\mathbf{x}_t)}{\gamma_T(\mathbf{x}_t)}\prod_{i=t+1}^{T} p(\mathbf{x}_{i+1}|\mathbf{x}_i)\prod_{i=t}^{T} p(\mathbf{y}_i|\mathbf{x}_i)d\mathbf{x}_{t+1:T} \\
&= \tilde{p}(\mathbf{y}_{t:T})\int\cdots\int \frac{\tilde{p}(\mathbf{x}_{t:T}|\mathbf{y}_{t:T})}{\gamma_T(\mathbf{x}_t)}d\mathbf{x}_{t+1:T} \\
&= \tilde{p}(\mathbf{y}_{t:T})\frac{\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t:T})}{\gamma_t(\mathbf{x}_t)}
\end{aligned}$$

7.2.2 The prediction and update steps

The prediction step with the backward filter is defined as.

$$\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t+1:T}) \triangleq \int \tilde{p}(\mathbf{x}_{t+1}|\mathbf{y}_{t+1:T}) \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)\gamma_t(\mathbf{x}_t)}{\gamma_{t+1}(\mathbf{x}_{t+1})} d\mathbf{x}_{t+1} \tag{40}$$

Again it has to be stessed that $\tilde{p}(\mathbf{x}_{t+1}|\mathbf{y}_{t+1:T})$ is not a probability measure if $\gamma_{t+1}(\mathbf{x}_{t+1}) \neq \int(\mathbf{x}_{t+1}|\mathbf{x}_t)\gamma_t(\mathbf{x}_t)d\mathbf{x}_t$. In order to have a finite intgral the artificial distribution $\{\gamma_t(\mathbf{x}_t)\}$ should selcted such that

$$\frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)}{\gamma_{t+1}(\mathbf{x}_{t+1})} < \infty, \tag{41}$$

for any $(\mathbf{x}_{t+1}, \mathbf{x}_t) \in \Omega$. Put in a more loosly tune we say that $\gamma_{t+1}(\mathbf{x}_{t+1})$ has have thicker tails than $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ for any $\mathbf{x}_t$ [1]. Before the update step is defined the following needs to be calculated

$$\begin{aligned} p(\mathbf{y}_{t:T}|\mathbf{x}_t) &= \int p(\mathbf{y}_{t+1:T}|\mathbf{x}_t)p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{y}_t|\mathbf{x}_t)d\mathbf{x}_t \\ &= \int \frac{\tilde{p}(\mathbf{x}_{t+1}|\mathbf{y}_{t+1:T})}{\gamma_{t+1}(\mathbf{x}_{t+1})} p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{y}_t|\mathbf{x}_t)d\mathbf{x}_t \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{\gamma_t(\mathbf{x}_t)} \int \tilde{p}(\mathbf{x}_{t+1}|\mathbf{y}_{t+1:T}) \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)\gamma_t(\mathbf{x}_t)}{\gamma_{t+1}(\mathbf{x}_{t+1})} d\mathbf{x}_{t+1} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t+1:T})}{\gamma_t(\mathbf{x}_t)} \end{aligned} \tag{42}$$

with the above the update step can now be defined

$$\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t:T}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t+1:T})}{\int p(\mathbf{y}_t|\mathbf{x}_t)\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t+1:T})d\mathbf{x}_t}, \tag{43}$$

which has been renormalized to be a probability measure. The $\gamma_t(\mathbf{x}_t)$ in the denomenator of the previous equation (42) will cancel out in the combination of the forward and backward filter.

7.2.3 The combination step

The combination of the forward and backward filter yields the marginal smoothed distribution. Thus, for $t = \{2, \cdots, T-1\}$

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:T}) &\propto p(\mathbf{x}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{t:T}|\mathbf{x}_t) \\ &\propto \frac{p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t:T})}{\gamma_t(\mathbf{x}_t)} \\ &\propto \frac{\int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1})d\mathbf{x}_{t-1}\tilde{p}(\mathbf{x}_t|\mathbf{y}_{t:T})}{\gamma_t(\mathbf{x}_t)}, \end{aligned} \tag{44}$$

and for $t = 1$

$$p(\mathbf{x}_1|\mathbf{y}_{1:T}) \propto \frac{\mu(\mathbf{x}_1)\tilde{p}(\mathbf{x}_1|\mathbf{y}_{1:T})}{\gamma_1(\mathbf{x}_1)} \tag{45}$$

The idea is to could construct the backward filter with a finite probability measure, therefore the $\gamma_t(\mathbf{x}_t)$ in the denomenator. When the forward and backward filter are combined the effect of the artificial distribution is cancel out.

The final approximation of the combination step is a follows

$$\hat{p}(d\mathbf{x}_t|\mathbf{y}_{1:T}) \propto \sum_{i=1}^{N} \tilde{\omega}_t^i \sum_{j=1}^{N} \omega_{t-1}^j \frac{p(\tilde{\mathbf{x}}_t^i|\mathbf{x}_{t-1}^j)}{\gamma_t(\tilde{\mathbf{x}}_t^i)} \delta_{\tilde{\mathbf{x}}_t^i} d\mathbf{x}_t \tag{46}$$

7.2.4 The algorithm

To complet the derivation of Monte Carlo sampling of the two filter smoother the algorithm is given in (6)

procedure TFS
Forward filtering
For $t = 1, \cdots, T$, perform the particle filtering to obtain the weighted measure $\{\mathbf{x}_t^i, \omega_t^i\}_{i=1}^N$

Backward filtering
1. *Initialization*
For $i = 1, \cdots, N$, Sample candidates from the proposal distribution $\mathbf{x}_t^i \sim q(\dot{|}\mathbf{y}_T)$
Compute the importance weights:
$\tilde{\omega_T}^i \propto \frac{p(\mathbf{y}_T|\tilde{\mathbf{x}}_T^i)\gamma_T(\tilde{\mathbf{x}}_T^i)}{q(\tilde{\mathbf{x}}_T^i|\mathbf{y}_T)}$
2. For $t = T, \cdots, 2$ and $i = 1, \cdots, N$
Sample candidates from the proposal distribution $\mathbf{x}_t^i \sim q(\dot{|}\mathbf{x}_t^i, \mathbf{y}_{t-1})$
Calculate the backward importance weights:
$\tilde{\omega_{t-1}}^i \propto \frac{\tilde{\omega_t}^i p(\mathbf{y}_{t-1}|\tilde{\mathbf{x}}_{t-1}^i)\gamma_{t-1}(\tilde{\mathbf{x}}_{t-1}^i p(\tilde{\mathbf{x}}_{t-1}^i|\tilde{\mathbf{x}}_t^i)}{\gamma_t(\tilde{\mathbf{x}}_t^i)q(\tilde{\mathbf{x}}_{t-1}^i|\mathbf{x}_t^i, \mathbf{y}_t)}$
Resample if necessary
end procedure

Algoritme 6: The generic two filter smoother

As in the previous section an example with the TFS is given for the same setup as in the other experiments. The result of the TFS can be seen in the figure (7.2.4), note that the TFS can change the support of the posterior estimate from the forward filter and therefore the smoothed estimated is much closer to the truth. Also note the scatter plot in figure (7.1) there was some residuals from the symmetri from the likelihood kernel, which could be seen in the scatter plot as symmetri outliers. With the TFS filter these resudals have disapeared.

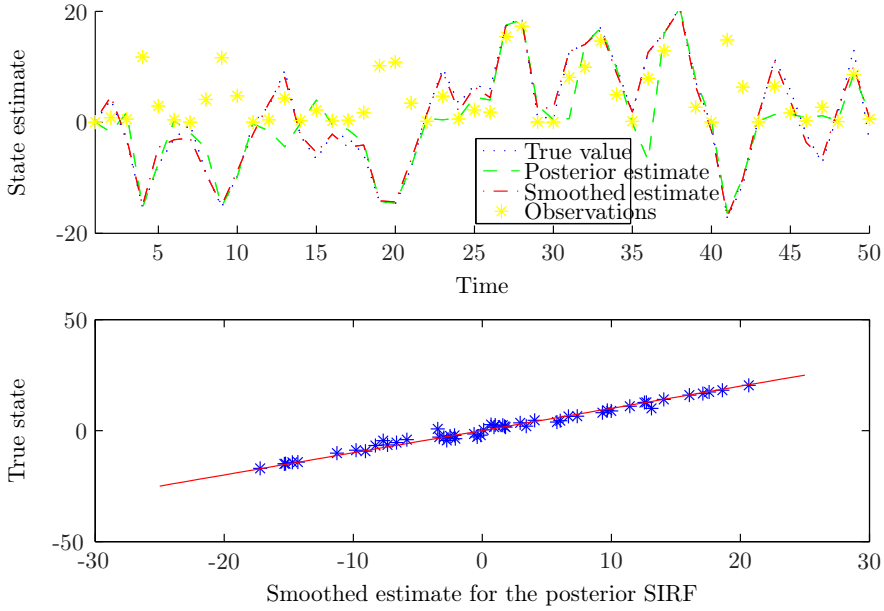

Figure 12: The smoothed estimate of the TFS and the scatter plot of the truth vs. the smoothed estimate

References

[1] Mark Briers. *Improved Monte Carlo Methods for State Space Models*. Doctoral thesis, University of Cambridge, Department of Engineering, 2007.

[2] Zhe Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. Technical report, Communications Research Laboratory, McMaster University, Hamilton ON, Canada, 2003.

[3] Randal Douc, Olivier Cappe, and Eric Moulines. Comparison of resampling schemes for particle filtering, 2005.

[4] A. Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, first edition, 2001.

[5] Arnaud Doucet, Neil J. Gordon, and Vikram Krishnamurthy. Particle filters for state estimation of jump markov linear systems, 2001.

[6] John Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57(6):1317–1339, 1989.

[7] Jeroen D. Hol, Thomas B. Schon, and Richard Karlson. On resampling algorithms for particle filters, 2006.

[8] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[9] Mike Klass, Mark Briers, Nando de Freitas, Arnaud Doucet, Simon Maskell, and Dustin Lang. Fast particle smoothing: If i had a million particles, 2006.

[10] Augustine Kong, Jun S. Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of American Statistical Association*, 89(1):278–288, 1994.

[11] Henrik Madsen. *Time Series Analysis*. DTU press, 2th edition, 2006.

[12] Henrik Madsen and Jan Holst. *Modelling Non-Linear and Non-Stationary Time Series*. DTU press, 1th edition, 2000.

[13] D. Q. Mayne. A solution of the smoothing problem for linear dynamic systems. *Automatica*, 4:73–92, 1966.

[14] A.F.M. Smith N.J. Gordon, D.J. Salmond. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140, 1993.

[15] Rudolph van der Merwe, Arnaund Doucent, Nando de Freitas, and Eric Wan. The unscented particle filter. Technical report, Cambridge University Engineering Department, 2000.

[16] Mike West. Mixture models, monte carlo, bayesian updating and dynamic models. *Computing Science and Statistics*, 24:325–333, 1993.