On using Likelihood-adjusted Proposals in Particle Filtering: Local Importance Sampling

Péter Torma
Eötvös Loránd University,
Pázmány Péter sétány 1/c
1117 Budapest, Hungary
tyus@axelero.hu

Csaba Szepesvári
Computer and Automation Research Institute
of the Hungarian Academy of Sciences,
Kende u. 13-17, 1111 Budapest, Hungary
szcsaba@sztaki.hu

Abstract

An unsatisfactory property of particle filters is that they may become inefficient when the observation noise is low. In this paper we consider a simple-to-implement particle filter, called 'LIS-based particle filter', whose aim is to overcome the above mentioned weakness. LIS-based particle filters sample the particles in a two-stage process that uses information of the most recent observation, too. Experiments with the standard bearings-only tracking problem indicate that the proposed new particle filter method is indeed a viable alternative to other methods.

1 Introduction

In this paper we consider filtering of non-linear stochastic processes. The problem studied can be formalized as follows. A sequence of values $Y_0, Y_1, Y_2, \ldots$ of some Euclidean space, governed by the equations

$$X_{t+1} = a(X_t; \xi_{t+1}), \quad X_0 \sim p_0(\cdot), \tag{1}$$
$$Y_{t+1} = b(X_{t+1}; \eta_{t+1}), \quad t = 0, 1, \ldots \tag{2}$$

is observed. Here X_t is the state of the system at time step t, p_0 is the initial distribution over the possible states at time step zero, and ξ_t, η_t are the process and observation noise processes; they are assumed to be composed of independent, identically distributed random variables and also to be independent of each other. The goal is to determine the posterior, $\pi_t(x) = p(X_t = x|Y_1, \ldots, Y_t)$, over the states at any time-step.

Three major factors can be identified that influence the performance of filtering algorithms: *(i)* the energy of the process noise; *(ii)* the energy of the observation noise; and *(iii)* the severity of 'perceptual aliasing' that makes the recovery of the state from the sequence of observations hard in the 'zero noise' limit.

Particle filters (see e.g. [**?**, **?**, **?**] and the references therein) approximate the posterior by empirical measures of the form

$$\pi_t(x) = \frac{\sum_{k=1}^N w_t^{(k)} \delta(x - X_t^{(k)})}{\sum_{k=1}^N w_t^{(k)}},$$

where $X_t^{(k)}$ and $w_t^{(k)}$ represent the kth particle's position and weight, respectively, and $\delta(\cdot)$ is Dirac's delta function. Here $X_t^{(k)}$ and $w_t^{(k)}$ are (random) quantities that depend on the sequence of past observations $Y_{1:t} \stackrel{\text{def}}{=} (Y_1, \ldots, Y_t)$. Given the empirical measure the estimate of the expectation of an arbitrary function h with respect to the posterior is obtained by

$$I_{t,N}(h) = \frac{\sum_{k=1}^N w_t^{(k)} h(X_t^{(k)})}{\sum_{k=1}^N w_t^{(k)}}. \tag{3}$$

The generic particle filter works by updating the particle's positions and weights in a recursive manner. The update is composed of two steps: computation of the particle's new positions is done by sampling from the so-called *proposal function*, followed by the update of the weights and an optional resampling step [**?**]. In the SIR filter [**?**] the particle's positions are updated independently of each other by sampling the kth particle's new position using $a(X_t^{(k)}; \xi_t^{(k)})$, where $\xi_t^{(k)}$ is sampled from the common underlying distribution of the process noise variables ξ_t, whilst the weight of the kth particle is computed by evaluating the observation likelihood $p(Y_t|X_{t+1}^{(k)})$.

When the level of the observation noise is low, the observation likelihood function becomes 'peaky' or concentrated around its modes (the modes correspond to the states that are locally most likely to 'cause' the most recent observation). If the position of a particle is not sufficiently close to one of these modes then the particle's weight will become small and thus the particle will bring in little information into the estimate of the posterior. If this happens for most of the particles then the quality of the approximation to the posterior degrade seriously. We call this problem the *"curse of reliable observations"*.

The curse of reliable observations is a well-known peculiarity of particle filters. The general advise to remedy

this problem is to use a proposal that depends on the most recent observation [**?**]. However, finding a proposal that is both tractable and still yields good performance can be notoriously hard. A straightforward alternative is to increase the number of particles until it is ensured that many particles are close to the peaks of the observation likelihood function. In high dimensional state-spaces this approach may require an enormous number of particles, which in turn slows down the filtering process. Hence methods that make it possible to keep the number of particles low are of considerable interest.

Since the inefficiency stems from the particles' positions not being close enough to the modes of the observation likelihood, it is a natural idea to let the modes of the likelihood 'attract' the particles. In this paper we consider algorithms that subscribe to this idea. The algorithms considered in this paper generate the particles' positions in a two-stage sampling process, where the first step uses the prior states, whilst the second uses the most recent observation.

In this paper we consider two methods, the first method, the *local likelihood sampling* (LLS) based particle filter introduced in [**?**], is used to motivate the second. In the LLS-filter the first sampling step is the same as in SIR, whilst in the second step of the LLS-filter a localized version of the observation likelihood function is used to adjust the particles' positions. Weights are calculated so that the process remains (locally) unbiased. In the case of the second algorithm proposed here, the first sampling step remains the same, whilst in the second step the observation likelihood is replaced by a user-chosen proposal density function that should be designed to be 'close' to the likelihood. The weight update equations are modified so that unbiasedness is retained. We call this second algorithm the *local importance sampling* based particle filter. A particular variant that employs mixture of Gaussian proposals is studied in greater detail. Experimental results with the standard bearings only tracking problem indicate the superiority of the proposed method to some of its alternatives.

2 Notation

Let us denote the transition kernel corresponding to the dynamics a (cf. (1)) by $K = K(u|v)$, i.e., $\int_{\mathcal{U}} K(u|v)\,du = P(a(v,\xi_t) \in \mathcal{U})$, where $\mathcal{U}$ is any measurable subset of the state-space. Further, let us denote the observation likelihood density by $r = r(y|x)$, i.e., $\int_{\mathcal{Y}} r(y|x)\,dy = P(b(x,\eta_t) \in \mathcal{Y})$, where $\mathcal{Y}$ is any measurable subset of the observation space.

3 Algorithms

3.1 LLS-filters

The basic idea of LLS-based particle filters [**?**], is to draw a sample from the prediction density as in SIR, but then allow the observation density to 'perturb' the position of the particles. A window-function (g) is used to localize the observation density's effect on the sample, hence the name of the procedure. The role of localization is to prevent particles 'jump around' in the state-space, i.e. to keep the information in the previous estimate of the posterior. The procedure is shown in Figure 1.

Initialize $\{(Z_0^{(k)}, 1/N)\}_{k=1}^N$ from the prior $p_0(\cdot)$.
For $t = 1, 2, \ldots$
  For $k = 1, 2, \ldots, N$
    Resample $S_{t-1} = \{(Z_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^N$ to obtain a new sample $({Z_{t-1}^{(k)}}', 1/N)$
    Predict $X_t^{(k)}$ by drawing a sample from $K(\cdot|{Z_{t-1}^k}')$
    Perturb $X_t^{(k)}$ by drawing $Z_t^{(k)}$ from $\frac{1}{\alpha_t^{(k)}} r(Y_t|\cdot) g(X_t^{(k)} - \cdot)$, where

$$\alpha_t^{(k)} = \int r(Y_t|x) g(X_t^{(k)} - x)\,dx\ .$$

    Update weight $w_t^{(k)}$ using

$$\hat{w}_t^{(k)} = \alpha_t^{(k)} \frac{K(Z_t^{(k)}|{Z_{t-1}^{(k)}}')}{K(X_t^{(k)}|{Z_{t-1}^{(k)}}')}\ .$$

  EndFor
  Normalize weights using $w_t^{(k)} = \hat{w}_t^{(k)} / \sum_{j=1}^N \hat{w}_t^{(j)}$.
EndFor

Figure 1. LLS-based Particle Filter

The following proposition was shown to hold in [**?**]:

Proposition 1 *Let (X_t, Y_t) evolve according to Equations (1)-(2). Assume that g is a non-negative, integrable function satisfying $I(g) = 1$. Let $\{(w_t^{(k)}, Z_t^{(k)})\}$ be the particle step obtained at time step t of the LLS-based particle filter. Then*

$$E[w_t^{(k)} h(Z_t^{(k)})|Y_{1:t}] = E[h(X_t)|Y_{1:t}] p(Y_t|Y_{1:t-1}).$$

In words, the statement of the proposition means that the weighted sample obtained at time step t represents the posterior properly, up to a constant factor (dependent only on the observations). As a consequence of the proposition, we get that $E[h(X_t)|Y_{1:t}]$ can be approximated using weighted averages of the form (3) and convergence results (almost sure convergence, convergence in distribution, mean-square, finite sample performance bounds) can be derived along the lines of previous proofs (see [**?**] and references therein).

What is more interesting is that in [**?**] a result was proven where it was shown that the LLS-filter can be more efficient than SIR (see Proposition 2, [**?**]) provided that the cross-correlation between the observation density and the observation density multiplied by the prediction density is much

larger than the cross-correlation between the convolution of the window function and the observation density, and the convolution of the window function and the product of the observation and prediction densities. Dropping conditioning on past observations and time indexes, let us denote by f the observation density, and by p the prediction density. Then the condition for improved performance has the form $\epsilon \leq \langle f, fp\rangle - \langle f * g, (fp) * g\rangle$. Here ϵ is a constant that depends on p and g (we omit the definition of ϵ due to the lack of space; the interested reader can find it in [**?**]) and $u * v$ denotes the convolution of u and v. Since for typical choices of the window function convolution with it cuts high frequencies, the condition can be expected to hold for a wide class of problems, especially when the prediction density, p, is 'broad' as compared to the observation density, f.

Note that the algorithm can be generalized easily to use window functions that change depending on $X_t^{(k)}$. One just needs to replace $g(X_t^{(k)} - x)$ in the sample-perturbation step by $g(X_t^{(k)} - x; X_t^{(k)})$ and redefine $\alpha_t^{(k)}$ accordingly: $\alpha_t^{(k)} = \int p(Y_t|x) g(X_t^{(k)} - x; X_t^{(k)})\, dx$. The simplest application of this is to fit g to match the energy distribution of the observation noise. Similarly g can be made dependent on the observation Y_t.

3.2 Local Importance Sampling

One problem with LLS-filters is that they depend on whether sampling sampling from $r(Y_t|\cdot)g(X_t^{(k)} - \cdot)/\alpha_t^{(k)}$ can be implemented efficiently. One possible remedy for this problem is to introduce a proposal density to replace $r(Y_t|\cdot)$. The corresponding algorithm, called Local Importance Sampling, is given in Figure 2.

The following proposition can be shown to hold (the proof is omitted due to the lack of space):

Proposition 2 *Let (X_t, Y_t) evolve according to Equations (1)-(2). Assume that g is a non-negative, integrable function satisfying $I(g) = 1$ and let $q_{x,y}(\cdot) > 0$ be a bounded, integrable function for all x, y. Let $\{(w_t^{(k)}, Z_t^{(k)})\}$ be the particle set obtained at time step t of the LIS-based particle filter. Then*

$$E[w_t^{(k)} h(Z_t^{(k)})|Y_{1:t}] = E[h(X_t)|Y_{1:t}] p(Y_t|Y_{1:t-1}).$$

As a consequence of this result, the LIS-based particle filter enjoys similar theoretical properties as SIR. Building on the previous argument that shows that LLS is more efficient than the naive algorithm, one expects that under similar conditions LIS will also be more efficient provided that the proposal function $q_{x,y}$ fits $r(y|\cdot)$ around x for any x, y. The efficiency of the new algorithm will be demonstrated in the next section on the standard bearings-only tracking problem in the next section. However, first let us consider an important practical variant of this algorithm.

Initialize a sample set $\{(Z_0^{(k)}, 1/N)\}_{k=1}^N$ according to the prior $p_0(\cdot)$.
For $t = 1, 2, \ldots$
  For $k = 1, 2, \ldots, N$
    Resample $S_{t-1} = \{(Z_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^N$
      to obtain a new sample $(Z_{t-1}^{(k)\prime}, 1/N)$
    Predict $X_t^{(k)}$ by drawing a sample from $K(\cdot|Z_{t-1}^{k\prime})$
    Perturb $X_t^{(k)}$ by drawing $Z_t^{(k)}$ from
      $\frac{1}{\alpha_t^{(k)}} q_{X_t^{(k)}, Y_t}(\cdot) g(X_t^{(k)} - \cdot)$, where

$$\alpha_t^{(k)} = \int q_{X_t^{(k)}, Y_t}(x) g(X_t^{(k)} - x)\, dx\ .$$

    Update weight $w_t^{(k)}$ using

$$\hat{w}_t^{(k)} = \alpha_t^{(k)} \frac{r(Y_t|Z_t^{(k)})}{q_{X_t^{(k)}, Y_t}(Z_t^{(k)})} \frac{K(Z_t^{(k)}|Z_{t-1}^{(k)\prime})}{K(X_t^{(k)}|Z_{t-1}^{(k)\prime})}\ .$$

  EndFor
  Normalize weights using $w_t^{(k)} = \hat{w}_t^{(k)} / \sum_{j=1}^N \hat{w}_t^{(j)}$
EndFor

Figure 2. LIS-based Particle Filter

3.3 Using Gaussian Mixture Proposal in LIS-based particle filters

A particularly attractive, easy to implement LIS-based particle filter is obtained when the proposal function is chosen to be a mixture of Gaussians and the window function is chosen to be a Gaussian, too. The purpose of this section is to give the details of the resulting procedure.

Let u denote the state-observation pair (x, y) and choose $q_{x,y} = q_u$ to be a mixture of Gaussians with n components, having priors $p_1, \ldots, p_n$, means $m_{u,1}, \ldots, m_{u,n}$ and variances $\sigma_{u,1}^2, \ldots, \sigma_{u,n}^2$. Further, choose the window function to be a zero-mean Gaussian with variance σ_g^2. Mixture of Gaussians are attractive due to their universal approximation properties for continuous densities [**?**, **?**] and also due to their analytic tractability which we build heavily on here:

For implementing the LIS-based particle filter, one needs to be able to draw samples from $q_u(\cdot)g(x - \cdot)$, as well as to evaluate $(q_u * g)(x)$. In our case, as it is well known, $q_u(\cdot)g(x - \cdot)$ is a mixture of Gaussians with means

$$m_{u,i} = \frac{x\sigma_{u,i}^2 + m_{u,i}\sigma_g^2}{\sigma_g^2 + \sigma_{u,i}^2}, \tag{4}$$

variances and

$$\sigma_{u,i} = \frac{\sigma_g \sigma_{u,i}}{\sqrt{\sigma_g^2 + \sigma_{u,i}^2}} \tag{5}$$

and (un-normalized) weights:

$$L_{u,i} = \frac{p_i}{\sqrt{2\pi(\sigma_g^2 + \sigma_{u,i}^2)}} e^{-\frac{1}{2}\frac{(x-m_{u,i})^2}{(\sigma_{u,i}^2+\sigma_g^2)}}. \quad (6)$$

Hence, sampling from $q_u(\cdot)g(x-\cdot)$ can be implemented by first drawing an index from the normalized weights $(L_{u,1}/L_u, \ldots, L_{u,n}/L_u)$, where $L_u = \sum_{i=1}^n L_{u,i}$, and then drawing a sample from the appropriate Gaussian. Further, $(q_u * g)(x)$ is just equal to L_u.

Hence, LIS-based particle filters are particularly easy to implement when used with mixture of Gaussian proposals and a Gaussian window function. The calculations are further elaborated on in Figure 3 for a single particle and a single time-step, assuming that the prediction density is p and the observation density is f (time indexes and conditioning on previous samples/observations are dropped). Before presenting our experimental results we discuss some relevant related algorithms. In the experiments we will compare the performance of the proposed new method to that of one of these algorithms (AVM).

Inputs: prediction density (p), observation density (f), observation (Y)

- Draw the jth particle X_j from p
- Calculate the Gauss-Mixture parameters ($m_{X_j,Y,i}$, $\sigma_{X_j,Y,i}$, $L_{X_j,Y,i}$) using Equations (4)–(6).
- Draw an index k from $\left\{ \frac{L_{X_j,Y,i}}{L_{X_j,Y}} \right\}_{i=1}^n$
- Draw Z_j a Gaussian with mean $m_{X_j,Y,k}$ and variance $\sigma_{X_j,Y,k}$.
- Calculate the weight

$$w_j = \left(\sum_{i=1}^n L_{X_j,Y,i} \right) \frac{f(Z_j)}{q_{X_j,Y}(Z_j)} \frac{p(Z_j)}{p(X_j)}.$$

Figure 3. Local Importance Sampling with mixture-of-Gaussian proposals and Gaussian window function

4 Related Work

Due to space restrictions, we give only a few key references.

One of the best known particle filter whose aim is to overcome the curse of reliability is the *Auxiliary Variable Method (AVM)* introduced by Pitt and Shephard [**?**]. AVM uses a proposal density of the form $q(x_t|X_{t-1}^{(1)}, \ldots, X_{t-1}^{(1)}, Y_t) = \sum_{k=1}^N r(Y_t|\overline{X}_t^{(k)}) K(x_t|X_{t-1}^{(k)})$, where $\overline{X}_t^{(k)}$ is e.g. the expected next state for particle k (i.e. $\overline{X}_t^{(k)} = E[a(X_{t-1}^{(k)}, \xi_t)|X_{t-1}^{(k)}])$. Sampling is implemented by first doing a weighted resampling step using the weights $r(Y_t|\overline{X}_t^{(k)})$ and then drawing the next states using the transition density kernel K. AVM can be more efficient than SIR when the process noise variance is low and the observation likelihood is not too peaky. AVM is similar to LLS/LIS-filters in that it is also a two-stage scheme. However, in AVM the particles' position is still sampled from the prediction density, whilst in LLS/LIS-filters the observation directly influences the particles's positions. One issue with AVMs is that when the observation likelihood is peaky and the number of particles is not high enough then resampling the particle set using $\{r(Y_t|\overline{X}_t^{(k)})\}$ might not be successful at picking 'right' particles. A similar problem occurs when the prediction density is multi-modal (though in such a case $\overline{X}_t$ could be replaced by random samples from the prediction density). Also, when the prediction density has a large variance then even if the first stage is successful, sampling the particle's next state from K might yield to a set of particles that are spread out too much in the state-space. The advantage of AVM to LIS-filters is that in LIS-filters the user has to design the proposals, though this choice can be made automatic by employing standard density estimation procedures.

Another recent method is *likelihood sampling* considered e.g. in details in [**?**]. In this approach it is the likelihood function $p(Y_t|\cdot)$ that is used as the proposal, whilst the prediction density is used to calculate the weights. Thus the success of this method depends on whether the likelihood is a good predictor of the true state. For multi-modal likelihoods (when aliasing effects are severe) a large number of particles can be generated away from the likely next positions of the true state. These particles will get low weights in the weighting process and thus will have no significant effect on the estimated posterior. Hence the effective sample-size would be small in this case. Our method overcomes this problem by first sampling from the prediction density and hence concentrating the samples in the vicinity of the 'correct' peaks of the likelihood function and using a localized version of the likelihood.

The LS-N-IPS algorithm, introduced in [**?**], uses the prediction density to derive the new particle set which is then locally modified by climbing the observation likelihood. This algorithm introduces some bias and relies on the availability of a method to climb the observation likelihood. Although, according to the well-known bias-variance dilemma, introducing bias is not necessarily 'bad', LLS/LIS-filters may achieve roughly the same variance reduction that is possible to get using LS-N-IPS, but with no additional bias (weighted importance sampling itself yields biased estimate of the posterior). Further, LLS/LIS-filters do not require a hill-climbing algorithm.

Yet another recent method is the "Boosted Particle Filter" [**?**]. Using our notation, this method uses the following

proposal:

$$q(\cdot|X_{t-1}, Y_t) = \alpha(\overline{X}_t, Y_t)\, r(Y_t|\cdot) g(\overline{X}_t - \cdot) + \big(1 - \alpha(\overline{X}_t, Y_t)\big)\; K(\cdot|X_{t-1}),$$

where $\overline{X}_t$ is the expected next state given the current state X_{t-1} and

$$\alpha(\overline{x}, y) = \begin{cases} A, & \text{if } r_0 < \max_{x': d(x', \overline{x}) \le \lambda} r(y|x'); \\ B, & \text{otherwise.} \end{cases}$$

Here $0 < B \ll A < 1$, and r_0, λ are parameters to be chosen by the user. Hence, when the observation likelihood is sufficiently large in a neighborhood of the expected next state then the observation likelihood is used to draw the next sample; whilst if the observation likelihood is not sufficiently large then the prediction density is used to draw the next position. Note that the version of this algorithm presented in [**?**] uses heuristically derived approximations to the observation likelihood and it is slightly more complicated than the one presented here. In any case, the above proposal depends on the most recent observation and drawing samples from it can be done in an efficient manner. However, just like in the case of AVM, when the prediction density is multi-modal or when the prediction density has a large variance then since the expected value of the next state is a bad predictor of where the next state might be, the algorithm degrades to SIR.

5 Experiments

In this section we compare the performance of the LIS-filter to that of SIR and AVM on the standard 'bearings-only tracking' problem that has been considered previously by several authors [**?**, **?**, **?**, **?**]. In this problem the aim is to track the (horizontal) motion of a ship, while observing only angles to it. Without the loss of generality,let us assume that the coordinate system is fixed to the observer. The ship's state is assumed to follow a second order AR process, with its acceleration driven by white noise:

$$X_{t+1} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} X_t + \sigma_\eta \begin{pmatrix} \frac{1}{2} & 0 \\ 1 & 0 \\ 0 & \frac{1}{2} \\ 0 & 1 \end{pmatrix} \xi_t,$$

where X_t, $\xi_t \in \mathbb{R}^2$, $\xi_{t1}, \xi_{t2} \sim \mathcal{N}(0,1)$, and X_{t1}, X_{t3} represent the ship's vertical and horizontal positions, respectively, whilst X_{t2}, X_{t4} represent the ship's vertical and horizontal velocities. The initial state is sampled from a 4-dimensional Gaussian with a diagonal covariance matrix.

What makes this problem particularly challenging is that the observations depend on the state only through the angle, $\theta_t = \tan^{-1}(X_{t3}/X_{t1})$, at which the ship is observed. The observation noise is defined using a wrapped Cauchy density:

$$r(y|\theta_t) = \frac{1}{2\pi} \frac{1 - \rho^2}{1 + \rho^2 - 2\rho\cos(y - \theta_t)}.$$

This density (when ρ is close to one) is thought to reflect well a sonar's behaviour: angle measurements are typically very reliable, whilst possible outliers are well modelled by the heavy tails of the wrapped Cauchy distribution.

The parameters of the model used in the experiments are as follows: $\sigma_\eta = 0.001$, $\rho = 1 - 0.005^2$, and the initial state is sampled from a Gaussian with means $(-0.05, 0.001, 0.2, -0.055)$ and with a diagonal covariance matrix with diagonal entries given by $0.001 \times (0.5^2, 0.005^2, 0.3^2, 0.01^2)$. Figure **??** gives an example of the ship's motion and the observed angles.

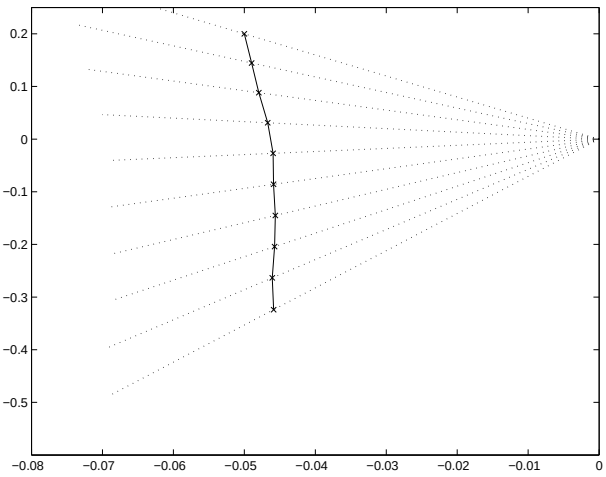

Figure 4. An example of the ship's motion together with the observation directions.

5.1 Results

The performance of the LIS-based particle filter was compared to that of AVM and SIR. All particle filters in these experiments use $N = 300$ particles.

For the implementation of the LIS-based particle filter the importance function $q_{x,y}$ is best described using polar coordinates: Basically, $q_{x,y}$ is a Gaussian in the angle coordinates with variance $\delta^2_{x,y} = 1 - \rho$. The window function g is defined in the angular space with dispersion $\delta_g = 0.05$.

Figure **??** shows particle clouds generated by the three algorithms for an arbitrary selected time-step. It should be clear from the figure that the particle set generated by LIS (shown as dots on the figure) is much better concentrated around the true state than the sets generated by both AVM ('+') and SIR ('$\times$'). In particular, the particles are more concentrated along the lines pointing towards the true state. Also, the particle sets generated by AVM are more concentrated around the true state than those generated by SIR. We note that a straightforwardly implemented likelihood sampling algorithm would perform much weaker than any of these algorithms as it would have no clue about the distance of the ship, and thus it would need to distribute samples evenly along the measurement lines. In order to get a

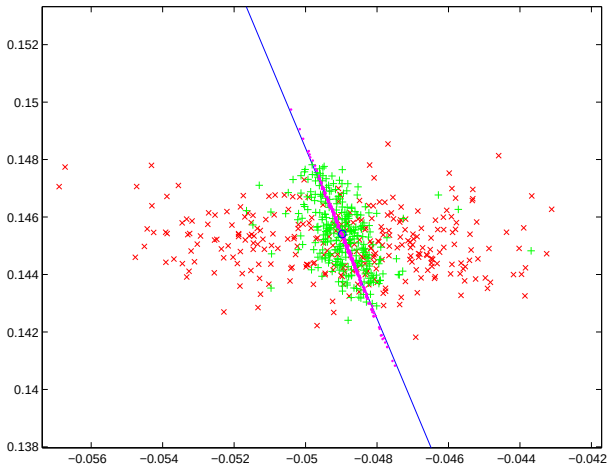

Figure 5. Particle clouds generated by AVM ('+'), SIR ('×') and LIS ('·') for the bearings only tracking problem.

more precise picture of the performance of these algorithms, we have measured the tracking performance by computing the Euclidean distance between the predicted and the actual ship positions as a function of time. The errors were measured with 20 independently generated tracking (measurement) sequences, and by running each algorithms 100 times on each of the 20 measurement sequences. Figure **??** shows the resulting tracking error of SIR, AVM and LIS as a function of the number of time steps. As expected, AVM performs better than baseline SIR, but LIS improves upon the performance of both SIR and AVM by a considerable margin. The observed performance differences were found to be significant.

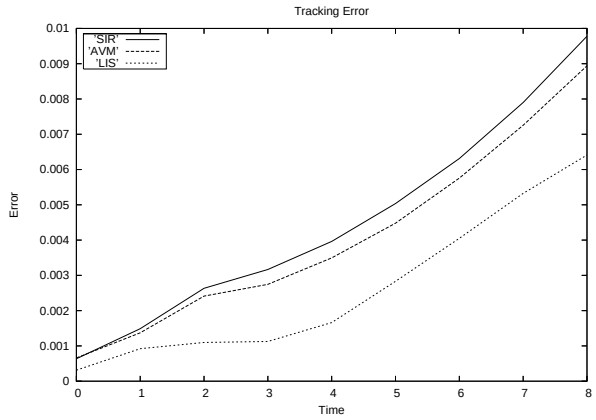

Figure 6. Tracking errors of SIR, AVM and LIS as a function of time

6 Conclusions

We have proposed a family of algorithms to enhance particle filters with the aim to overcome the 'curse of reliable observations'. The proposed algorithms, the LLS/LIS-filters of which LIS-filters were introduced here, are modifications of the standard SIR algorithm whereas after the prediction step the position of the particles are randomly re-sampled from a localized version of the observation density or a localized importance function. We argued that using the new method higher effective sample sizes can be achieved when the observations are reliable and when the design parameters of the new algorithm are chosen appropriately. One expects this increase to be reflected by an improved tracking performance. Experiments with the standard bearings-only tracking problem indicate that the proposed algorithm is indeed capable of improving the tracking performance as compared with the performance of SIR and AVM.

References

[1] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Department of Electrical Engineering, Linkoping, 2000.

[2] J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for nonlinear problems. *IEEE Proceedings-Radar Sonar and Navigation*, 146(1):2–7, 1999.

[3] D. Crisan and A. Doucet. A survey of convergence results on particle filtering for practitioners. *IEEE Trans. Signal Processing*, 50(3):736–746, 2002.

[4] W. B. D. Fox, S. Thrun and F. Dellaert. Particle filters for mobile robot localization. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, New York, 2001. Springer.

[5] A. Doucet. On sequential simulation based methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 1998.

[6] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Proc. Inst. Elect. Eng. F*, 140(2):107–113, 1993.

[7] P. D. Moral and G. Salut. Non-linear filtering using monte carlo particle methods. *C.R. Acad. Sci. Paris*, pages 1147–1152, 1995.

[8] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, volume 1, pages 28–39, 2004.

[9] M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filter. *Journal of the American Statistical Association*, 94:590–599, 1999.

[10] D. Scott. *Multivariate Density Estimation. Theory, Practice, and Visualization*. J. Wiley & Sons, New York, London, Sydney, 1992.

[11] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. J. Wiley & Sons, New York, London, Sydney, 1985.

[12] P. Torma and C. Szepesvári. LS-N-IPS: an improvement of particle filters by means of local search. *Proc. Non-Linear Control Systems(NOLCOS'01) St. Petersburg, Russia*, 2001.

[13] P. Torma and C. Szepesvári. Enhancing particle filters using local likelihood sampling. *ECCV2004, Prague. Lecture Notes in Computer Science*, pages 16–28, 2004.