12

Improving Regularised Particle Filters

Christian Musso
Nadia Oudjane
Francois Le Gland

12.1 Introduction

The optimal filter computes the posterior probability distribution of the state in a dynamical system, given noisy measurements, by iterative application of prediction steps according to the dynamics of the state, and correction steps taking the measurements into account. A new class of approximate nonlinear filter has been recently proposed, the idea being to produce a sample of independent random variables, called a particle system, (approximately) distributed according to this posterior probability distribution. The method is very easy to implement, even in high-dimensional problems, since it is sufficient in principle to simulate independent sample paths of the hidden dynamical system.

The earliest contribution in this field was (Gordon et al. 1993), which proposed to use sampling / importance resampling (SIR) techniques of (Rubin 1988) in the correction step. If the resampling step were skipped, then the method would reduce to the weighted particle method, a very inefficient method in which the measurements are only used to update the weights (likelihood) associated with the non-interacting particle system. The positive effect of the resampling step is to automatically concentrate particles in regions of interest of the state-space. However, it was already observed in (Gordon et al. 1993) that some difficulties could occur with the particle or bootstrap filter, if the hidden dynamical system was noise-free, or has very small noise, and also (for different reason however) if the observation noise had very small variance. Heuristics were proposed there to remedy these difficulties.

We present a more systematic approach in Section 12.2, one based on regularisation of the empirical distribution associated with the particle system, using the kernel method,; see (Silverman 1986). This results in two

A. Doucet et al. (eds.), *Sequential Monte Carlo Methods in Practice*

different approximations, called pre-regularised particle filter (pre-RPF) and post-regularised particle filter (post-RPF), depending on whether the regularisation step is taken before or after the correction step. Notice that an efficient implementation of the resampling step in the pre-RPF is the kernel filter (KF), which has been proposed in (Hürzeler and Künsch 1998). To handle efficiently the (difficult) case in which the observation noise has very small variance, we present an improved approximation in Section 12.3, based on a progressive correction (PC) principle, in which the correction step is split into sub-correction steps associated with a decreasing sequence of (fictitious) variance matrices for the observation noise. In Section 12.4, we present an additional improvement of the KF, called the local rejection regularised particle filter (L2RPF) that reduces the computing cost involved in the resampling step. In Section 12.5, simulation results are presented for several 2D–tracking problems, with bearings–only measurements, and with range and bearing measurements.

To be more specific, we consider the following model, with two sequences $(\mathbf{x}_t)_{t\geq 0}$ and $(\mathbf{y}_t)_{t\geq 0}$, called state and observation, and taking values in $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_y}$ respectively.

- In full generality, the state sequence $(\mathbf{x}_t)_{t\geq 0}$ is defined as an inhomogeneous Markov chain, with transition probability kernel Q_t, i.e.

$$\mathbb{P}(\mathbf{x}_t \in \mathrm{d}\mathbf{x} \mid \mathbf{x}_{0:t-1}) = \mathbb{P}(\mathbf{x}_t \in \mathrm{d}\mathbf{x} \mid \mathbf{x}_{t-1}) = Q_t(\mathbf{x}_{t-1}, \mathrm{d}\mathbf{x}) \ ,$$

  for all $t \geq 1$, and with initial probability distribution p_0. For instance, $(\mathbf{x}_t)_{t\geq 0}$ could be defined by the equation

$$\mathbf{x}_t = F_t(\mathbf{x}_{t-1}, \mathbf{w}_t) \ ,$$

  where $(\mathbf{w}_t)_{t\geq 0}$ is a sequence of independent random variables, not necessarily Gaussian, independent of the initial state $\mathbf{x}_0$.

- The observation sequence $(\mathbf{y}_t)_{t\geq 0}$ is related to the state sequence $(\mathbf{x}_t)_{t\geq 0}$ by

$$\mathbf{y}_t = H_t(\mathbf{x}_t) + \mathbf{v}_t \ ,$$

  for all $t \geq 0$, where $(\mathbf{v}_t)_{t\geq 0}$ is a sequence of independent random variables, not necessarily Gaussian, independent of the state sequence $(\mathbf{x}_t)_{t\geq 0}$. It is assumed that $\mathbf{v}_t$ has a probability density, i.e. $\mathbf{v}_t \sim g_t(\mathbf{v})\,\mathrm{d}\mathbf{v}$, for all $t \geq 0$. This special form of the observation equation allows us to define the likelihood function (see below) which is necessary for the methods proposed here.

The problem of nonlinear filtering is to compute at each time t the conditional probability distribution $\pi_{t|t}$ of the state $\mathbf{x}_t$ given a realisation of the observation sequence $\mathbf{y}_{0:t} = (\mathbf{y}_0, \cdots, \mathbf{y}_t)$ up to time t. Even if the optimal filter is in general difficult to compute, its algorithm can be easily

described in two steps. Introducing the conditional probability distribution $\pi_{t|t-1}$ of the state $\mathbf{x}_t$ given a realisation of the observation sequence $\mathbf{y}_{0:t-1} = (\mathbf{y}_0, \cdots, \mathbf{y}_{t-1})$ up to time $(t-1)$, the transition from $\pi_{t-1|t-1}$ to $\pi_{t|t}$ can be described by

$$\pi_{t-1|t-1} \xrightarrow[\text{Prediction}]{(1)} \pi_{t|t-1} = Q_t^* \, \pi_{t-1|t-1} \xrightarrow[\text{Correction}]{(2)} \pi_{t|t} = \Psi_t \cdot \pi_{t|t-1} \ . \tag{12.1.1}$$

1. Prediction: this step consists of the application of the transition probability kernel Q_t to $\pi_{t-1|t-1}$, i.e.

$$\pi_{t|t-1}(\mathrm{d}\mathbf{x}') = Q_t^* \, \pi_{t-1|t-1}(\mathrm{d}\mathbf{x}') = \int \pi_{t-1|t-1}(\mathrm{d}\mathbf{x}) \, Q_t(\mathbf{x}, \mathrm{d}\mathbf{x}') \ . \tag{12.1.2}$$

2. Correction: this step consists of Bayes rule. Since $\mathbf{y}_t$ is independent of the past observations $\mathbf{y}_{0:t-1}$ given $\mathbf{x}_t$, then introducing the likelihood function $\Psi_t(\mathbf{x}) = g_t(\mathbf{y}_t - H_t(\mathbf{x}))$, the correction step can be written as the projective product

$$\pi_{t|t}(\mathrm{d}\mathbf{x}) = \frac{\Psi_t(\mathbf{x}) \, \pi_{t|t-1}(\mathrm{d}\mathbf{x})}{\langle \pi_{t|t-1}, \Psi_t \rangle} = (\Psi_t \cdot \pi_{t|t-1})(\mathrm{d}\mathbf{x}) \ . \tag{12.1.3}$$

Throughout this article, $S^N(\pi)$ denotes the N-empirical distribution from π, i.e.

$$S^N(\pi) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\mathbf{x}^{(i)}} \quad \text{with } (\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}) \text{ i.i.d.} \sim \pi \ .$$

Notice that the correction step applied to $S^N(\pi)$ yields

$$\Psi \cdot S^N(\pi) = \sum_{i=1}^{N} \omega^{(i)} \, \delta_{\mathbf{x}^{(i)}} \quad \text{with } (\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}) \text{ i.i.d.} \sim \pi$$

$$\text{and } \omega^{(i)} = \Psi(\mathbf{x}^{(i)}) / \sum_{j=1}^{N} \Psi(\mathbf{x}^{(j)}) \ .$$

12.2 Particle filters

Particle methods are essentially based on Monte Carlo methods. The Monte Carlo principle allows approximation of a probability measure when a sample $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)})$ from that probability is given. This is a direct con-

sequence of the law of large numbers, which states the weak convergence of $S^N(\pi)$ to π with rate $1/\sqrt{N}$.

12.2.1 The (classical) interacting particle filter (IPF)

The classical particle filter has appeared under several names in the literature, such as interacting particle filter (IPF) in (Del Moral 1996), sampling / importance resampling (SIR) in (Gordon et al. 1993), or branching particle filter (BPF) in (Crisan and Lyons 1997), where the method is developed for the continuous-time case. In all these cases, the transition from $\pi^N_{t-1|t-1}$ to $\pi^N_{t|t}$ is described by the following two steps.

$$\pi^N_{t-1|t-1} \xrightarrow[\text{Sampled Prediction}]{(1)} \pi^N_{t|t-1} = S^N(Q_t^* \pi^N_{t-1|t-1}) \xrightarrow[\text{Correction}]{(2)} \pi^N_{t|t} = \Psi_t \cdot \pi^N_{t|t-1}. \tag{12.2.4}$$

In practice, those two steps consist of:

1. Sampled prediction
   - Resampling: generate the particles $(\mathbf{x}^{(1)}_{t-1|t-1}, \cdots, \mathbf{x}^{(N)}_{t-1|t-1})$ i.i.d. $\sim \pi^N_{t-1|t-1}$.
   - Evolution: independently for all i, generate the particles $\mathbf{x}^{(i)}_{t|t-1} \sim Q_t(\mathbf{x}^{(i)}_{t-1|t-1}, \cdot)$.

   Then set $\pi^N_{t|t-1} = S^N(Q_t^* \pi^N_{t-1|t-1}) = \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} \delta_{\mathbf{x}^{(i)}_{t|t-1}}$.

2. Correction: for all i, compute
   - $\omega_t^{(i)} = \Psi(\mathbf{x}^{(i)}_{t|t-1}) / \displaystyle\sum_{j=1}^{N} \Psi(\mathbf{x}^{(j)}_{t|t-1})$.

   Then set $\pi^N_{t|t} = \displaystyle\sum_{i=1}^{N} \omega_t^{(i)} \delta_{\mathbf{x}^{(i)}_{t|t-1}}$.

Remark 1. Here, resampling is equivalent to simulating the random vector $(N^{(1)}, \cdots, N^{(N)})$ representing the number of occurrences of each particle in the new system. This random vector follows a multinomial

distribution with parameters $(N, N, (\omega^{(i)}_{t-1|t-1})_{1\leq i\leq N})$ and can be rapidly simulated, in order $O(N)$ steps, in the following way.

1. Generate directly ordered uniform variables $u^{(1)} \leq \cdots \leq u^{(N)}$, see the Malmquist theorem in (Devroye 1986, p. 212).

2. For all i, set $p_i = \sum_{j=1}^{i} \omega^{(j)}_{t-1|t-1}$, and return the number $N^{(i)}$ of $u^{(j)}$'s lying within the interval $[p_i, p_{i+1})$.

In the first step of the sampled prediction, i.e. in the resampling step, we generate N i.i.d. random variables according to the weighted discrete probability distribution $\pi^N_{t|t} = \sum_{i=1}^{N} \omega^{(i)}_t \, \delta_{\mathbf{x}^{(i)}_{t|t-1}}$. The more likely particles are selected, so that the particle system concentrates in regions of interest of the state-space. This produces a new particle system according to which several particles may have the same location. If the dynamical noise is small or nonexistent, the variety of the particle system decreases at each time step because of the accumulation of repetitions in the sample, and the particle system ultimately concentrates on a single point of the state-space. This phenomenon is called *particle degeneracy*, and causes the divergence of the filter.

Nevertheless, the IPF is proved to converge to the optimal filter, in the weak sense, with rate $1/\sqrt{N}$, but the error is not uniformly bounded in time (except under some strong mixing assumptions on the transition kernel Q_t see (Del Moral and Guionnet 1998b)) which explains why some divergent behaviors are still observed; see (Del Moral and Guionnet 1998a) for an analysis of the asymptotic behavior.

We propose to add a step in the preceding algorithm, to ensure the diversity of the particle system as time progresses. The resulting filters, called regularised particle filters (RPF), are presented below.

12.2.2 Regularised particle filters (RPF)

The main idea consists in changing the discrete approximation $\pi^N_{t|t}$ to a continuous approximation such that the resampling step is changed into simulations from an absolutely continuous distribution, hence producing a new particle system with N different particle locations. In doing this, we implicitly assume that the optimal filter $\pi_{t|t}$ has a smooth density, which is the case in most applications. From the theoretical point of view, this additional assumption allows us to produce strong approximations of the optimal filter, in L^1 or L^2 sense. In practice, this provides approximate filters which are much more stable in time than the IPF.

Regularisation of an empirical measure

Let the regularisation kernel K be a symmetric probability density function on $\mathbb{R}^{n_x}$, such that

$$K \geq 0 \ , \quad \int K(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1 \ , \quad \int \mathbf{x}\,K(\mathbf{x})\,\mathrm{d}\mathbf{x} = 0 \ , \quad \int \|\mathbf{x}\|^2\,K(\mathbf{x})\,\mathrm{d}\mathbf{x} < \infty \ ,$$

and for any bandwidth $h > 0$, define the rescaled kernel

$$K_h(\mathbf{x}) = \frac{1}{h^{n_x}}\,K(\frac{\mathbf{x}}{h})$$

for any $\mathbf{x} \in \mathbb{R}^{n_x}$. For any probability distribution ν on $\mathbb{R}^{n_x}$, the regularisation of ν is the absolutely continuous probability distribution $K_h * \nu$ with density

$$\frac{\mathrm{d}(K_h * \nu)}{\mathrm{d}\mathbf{x}}(\mathbf{x}) = \int K_h(\mathbf{x} - \mathbf{u})\,\nu(\mathrm{d}\mathbf{u}) \ ,$$

where $*$ denotes the convolution operator. If $\nu = \Psi \cdot S^N(\pi) == \sum_{i=1}^{N} \omega^{(i)}\,\delta_{\mathbf{x}^{(i)}}$ is a discrete probability distribution on $\mathbb{R}^{n_x}$, where $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)})$ is a sample from π, then

$$\frac{\mathrm{d}(K_h * \nu)}{\mathrm{d}\mathbf{x}}(\mathbf{x}) = \frac{1}{h^{n_x}} \sum_{i=1}^{N} \omega^{(i)}\,K(\frac{1}{h}\,(\mathbf{x} - \mathbf{x}^{(i)})) = \sum_{i=1}^{N} \omega^{(i)}\,K_h(\mathbf{x} - \mathbf{x}^{(i)})$$

see Figure 12.1.

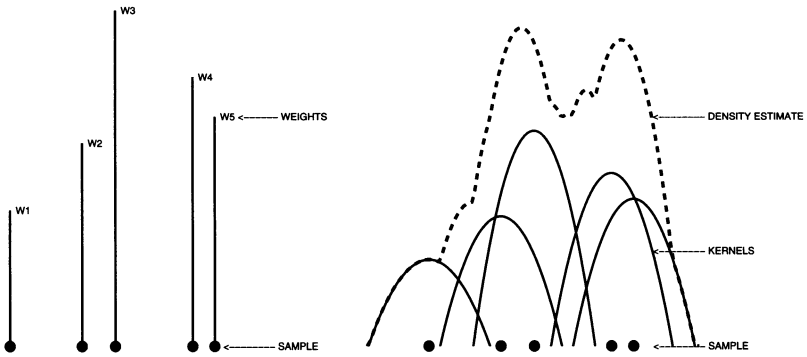

Figure 12.1. Regularisation of an empirical measure

The kernel and bandwidth are chosen so as to minimise the mean integrated error $\mathbb{E}\|K_h * \nu - \Psi \cdot \pi\|_1$, or the mean integrated square error $\mathbb{E}\|K_h * \nu - \Psi \cdot \pi\|_2^2$ between the posterior distribution $\Psi \cdot \pi$ and the corresponding regularised weighted empirical measure. In the special case of a

classical equally weighted sample, $\omega^{(i)} = 1/N$ for $i = 1, \cdots, N$, the density estimation theory, as described in (Silverman 1986, Devroye 1987), provides the optimal choice for the kernel,

$$K_{\text{opt}}(\mathbf{x}) = \begin{cases} \dfrac{n_x + 2}{2c_{n_x}} (1 - \|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{12.2.5}$$

and when the underlying density is Gaussian with unit covariance matrix, the optimal choice for the bandwidth,

$$h_{\text{opt}} = A(K)\, N^{-\frac{1}{n_x+4}} \quad \text{with} \quad A(K) = [8\, c_{n_x}^{-1}\, (n_x + 4)\, (2\sqrt{\pi})^{n_x}]^{\frac{1}{n_x+4}}\,, \tag{12.2.6}$$

where c_{n_x} is the volume of the unit sphere of $\mathbb{R}^{n_x}$. K_{opt} defined above is called the Epanechnikov kernel. To reduce the computing cost of generating from the regularised measure we can replace the Epanechnikov kernel by the Gaussian kernel, the optimal bandwidth associated (when the underlying density is Gaussian with unit covariance matrix) is then

$$h_{\text{opt}} = A(K)\, N^{-\frac{1}{n_x+4}} \quad \text{with} \quad A(K) = (4/(n_x + 2))^{\frac{1}{n_x+4}}\,. \tag{12.2.7}$$

In the general case of an arbitrary underlying density π, we make two approximations when assuming that the density is Gaussian with covariance matrix S equal to the empirical covariance matrix of the sample $(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)})$ from π. We then apply a linear transformation to achieve unit covariance (whitening). That is, $\mathbf{x}^{(i)}$ is changed into $A^{-1}\mathbf{x}^{(i)}$, where $A\,A^T = S$. The covariance matrix of the new particle system is then the unit matrix, and the bandwidth (12.2.6) or (12.2.7) can be used directly. This reduces to use the following rescaled regularisation kernel,

$$\frac{(\det A)^{-1}}{h^{n_x}} K\Big(\frac{1}{h} A^{-1}\mathbf{x}\Big)\,. \tag{12.2.8}$$

To handle the case of multi-modal densities, we chose $h = h_{\text{opt}}/2$; see Figure 3.3 in (Silverman 1986) .

Remark 2. Generating from the Epanechnikov kernel (12.2.5) consists of generating $\sqrt{\beta}\,T$, where β follows a beta distribution with parameters $(n_x/2, 2)$, and T is uniformly distributed over the unit sphere of $\mathbb{R}^{n_x}$, see (Devroye and Györfi 1985, pp. 236–237).

Two types of regularised particle filters are proposed depending on whether the regularisation step is taken before or after the correction step.

The post–regularised particle filter

The post–regularised particle filter (post–RPF) has been proposed in (Musso and Oudjane 1998, Oudjane and Musso 1999) and compared with the IPF

when applied to classic tracking problems such as the bearings–only problem or the range and bearing problem with multiple dynamic models. A theoretical analysis is developed in (Le Gland, Musso and Oudjane 1998).

We denote by $\tilde{\pi}^N_{t|t}$ the post–regularised approximation of $\pi_{t|t}$. The transition from $\tilde{\pi}^N_{t-1|t-1}$ to $\tilde{\pi}^N_{t|t}$ consists of these three steps

$$\tilde{\pi}^N_{t-1|t-1} \xrightarrow[\text{Sampled Prediction}]{(1)} \tilde{\pi}^N_{t|t-1} \xrightarrow[\text{Correction}]{(2)} \tilde{\nu}^N_t = \Psi_t \cdot \tilde{\pi}^N_{t|t-1} \xrightarrow[\text{Regularisation}]{(3)} \tilde{\pi}^N_{t|t} = K_h * \tilde{\nu}^N_t \ . \tag{12.2.9}$$

Because of the regularisation step (3), the first stage of the sampled prediction (1) is changed. Instead of simply resampling the particle system, the following algorithm is implemented, independently for all i

1. Generate $I \in \{1, \cdots, N\}$, with $\mathbb{P}(I = j) = \omega^{(j)}_{t|t}$.
2. Generate $\varepsilon \sim K$, Epanechnikov (12.2.5) or Gaussian kernel.
3. Compute $\mathbf{x}^{(i)}_{t|t} = \mathbf{x}^{(I)}_{t|t} + h\,\Gamma_t\,\varepsilon$, with $h = h_{\text{opt}}$ given by (12.2.6) or (12.2.7). If whitening is used, $\Gamma_t = A_t$ the square root of the empirical covariance matrix, see (12.2.8), otherwise $\Gamma_t = I$.

In terms of complexity, this algorithm is similar to the IPF because it only requires N additional generations from the kernel K at each time step. Moreover, the post-RPF markedly improves upon the IPF performance, especially in cases of small dynamic noise, as would be expected.

The pre-regularised particle filter

A theoretical analysis of the pre-regularised particle filter (pre-RPF) is developed in (Le Gland et al. 1998). Otherwise, an improved version of the pre–RPF, the kernel filter (KF), is proposed in (Hürzeler and Künsch 1998). This latter version is adapted in Section 12.4 to reduce the computing cost.

We denote by $\bar{\pi}^N_{t|t}$ the pre-regularised approximation of $\pi_{t|t}$. The transition from $\bar{\pi}^N_{t-1|t-1}$ to $\bar{\pi}^N_{t|t}$ consists of the following three steps:

$$\bar{\pi}^N_{t-1|t-1} \xrightarrow[\substack{\text{Sampled}\\ \text{Prediction}}]{(1)} \bar{\pi}^N_{t|t-1} \xrightarrow[\text{Regularisation}]{(2)} \bar{\nu}^N_{t|t-1} = K_h * \bar{\pi}^N_{t|t-1}$$

$$\xrightarrow[\text{Correction}]{(3)} \bar{\pi}^N_{t|t} = \Psi_t \cdot \bar{\nu}^N_{t|t-1} \,. \tag{12.2.10}$$

The cost of the sampled prediction rises because it requires the implementation of the following rejection algorithm (see Section 12.3.1) independently for all i.

1. Generate I uniformly in $\{1, \cdots, N\}$.
2. Generate $\varepsilon \sim K$, Epanechnikov (12.2.5) or Gaussian kernel, and U uniformly on $[0, 1]$.
3. Put $\mathbf{X} = \mathbf{x}^{(I)}_{t|t-1} + h\,\Gamma_t\,\varepsilon$, with $h = h_{\text{opt}}$ given by (12.2.6) or (12.2.7). If whitening is used, $\Gamma_t = A_t$ the square root of the empirical covariance matrix, see (12.2.8), otherwise $\Gamma_t = I$.
4. If $\Psi_t(\mathbf{X}) > U \sup_{\mathbf{x}\in\mathbb{R}^{n_x}} \Psi_t(\mathbf{x})$, then return $\mathbf{x}^{(i)}_{t|t} = \mathbf{X}$, otherwise go to step 1.

Both RPF's are proved to converge to the optimal filter in the weak sense, with rate $(h^2+1/\sqrt{N})$. The term h^2 corresponds to the error owing to regularisation. When $h = 0$, one recovers the rate of convergence of the IPF. In the strong L^1 sense, the error estimate is proportional to $(h^2+1/\sqrt{N\,h^{n_x}})$ for the post–RPF and pre–RPF, but, as for the IPF, those error estimates are in general not uniformly bounded in time.

12.3 Progressive correction

In this section we propose methods to approximate the correction step in controlling the local cost, between $(t-1)$ and t, in terms of error and computing time.

Lemma 12.3.1. *Let π and π' be two probability distributions on $\mathbb{R}^{n_x}$, let Q be a transition probability kernel, and let Ψ be a positive bounded function*

in $\mathbb{R}^{n_x}$. *Then the following inequalities hold:*

$$\|Q^*\,\pi - Q^*\,\pi'\|_{\mathrm{TV}} \leq \|\pi - \pi'\|_{\mathrm{TV}}\ , \tag{12.3.11}$$

$$\|\Psi \cdot \pi - \Psi \cdot \pi'\|_{\mathrm{TV}} \leq \delta\,\|\pi - \pi'\|_{\mathrm{TV}}\ , \tag{12.3.12}$$

with $\delta = \sup_{\mathbf{x}\in\mathbb{R}^{n_x}} \Psi(\mathbf{x})/\langle \pi, \Psi\rangle \geq 1$.

Inequality (12.3.12) is sharp, see (Hürzeler 1998, pp. 99–100), and shows that the correction step can induce tremendous errors when δ is high (which can occur when the measurement noise is small or when the state variable is unusual because then the denominator $\langle \pi, \Psi\rangle$ is small), whereas the prediction step is always a contraction, as seen from inequality (12.3.11). This is why we shall focus on the correction step. Henceforth, we are interested in methods to approximate the posterior probability distribution $\Psi \cdot \pi$. Remember that in the context of filtering the probability distribution π to be updated is the prediction filter (12.1.1). Afterwards, δ will be called the cost coefficient of the correction.

12.3.1 Focus on the correction step

In particle filtering, the main difficulties arise at the correction step. There are two kinds of problem encountered, depending on the method used to approximate the correction.

Weighting correction:

$\pi \to \Psi \cdot (S^N(\pi)) \to K_h * (\Psi \cdot (S^N(\pi)))$

As diagram (12.2.9) shows, this is the method used in the post–RPF. The correction step consists only in weighting the points of a sample from π. The likelihood of any other point of the state-space is not used. When measurements are accurate, which results in high δ, the likelihood function concentrates in a small region of the state-space that can be too small to contain points of the particle system. If that happens, we commonly observe the divergence of the filter during simulations, as suggested by (12.3.12). However, the computing time is cheap, and independent of δ.

Rejection correction:

$\pi \to K_h * (S^N(\pi)) \to \Psi \cdot (K_h * (S^N(\pi)))$

As diagram (12.2.10) shows, this is the method used in the pre–RPF. Here, the correction is applied directly to the regularised probability distribution. Hence, each point of the support of the probability density is updated and the problem induced by the discrete correction is solved. Unfortunately, this method requires to generate from π a random number of variables with negative binomial (or Pascal) distribution of parameter $(N, 1/\delta)$, and mean equal to $N\delta$ (as the sum of N independent variables

geometrically distributed with parameter $1/\delta$). This implies that, when δ is high, we always observe a great increase of the computing time. However, during simulations we have never observed the divergence of the filter, even when δ is high.

Whatever the method chosen to approximate the correction, the case of high δ is difficult to deal with. It implies either an increase in the error or an increase in the computing time. The aim of progressive correction is to offset these outcomes.

12.3.2 Principle of progressive correction

In principle, progressive correction is based on the following elementary observation. If π is a probability distribution on $\mathbb{R}^{n_x}$, and if Ψ, Ψ_1 and Ψ_2 are three bounded non-negative functions defined on $\mathbb{R}^{n_x}$, such that $\Psi = \Psi_1\ \Psi_2$, then it is equivalent to correct π by the function Ψ, or to correct π by Ψ_1 and then correct the resulting probability distribution by Ψ_2. In other words, $\Psi \cdot \pi = \Psi_2 \cdot (\Psi_1 \cdot \pi)$.

Progressive correction consists of splitting the correction step into several sub-correction steps for which the coefficient δ is well controlled. To do so, we need to choose a decomposition $(\Psi_1, \cdots, \Psi_n)$ of the likelihood function Ψ. This problem will be discussed in Section 12.3.3. For the moment, assume that this decomposition is given. Let $\nu_k = \Psi_k \cdot \nu_{k-1}$ denote the distribution resulting from the first k sub-corrections, $1 \leq k \leq n$, with $\nu_0 = \pi$. Let δ_k denote the cost coefficient corresponding to the k^{th} sub-correction, i.e. $\delta_k = \sup_{\mathbf{x} \in \mathbb{R}^{n_x}} \Psi_k(\mathbf{x}) / \langle \nu_{k-1}, \Psi_k \rangle$. It is easy to check that the total cost coefficient is equal to the product of the sub-cost coefficients, i.e. $\delta = \delta_1 \cdots \delta_n$ as soon as all fictitious likelihood functions Ψ_k reach their maximums in the same point of the state-space. There are two kinds of progressive correction methods depending on the method used to implement each sub-correction step.

Progressive weighting correction

Let $\tilde{\nu}_k$ denote the approximation of ν_k by successive weighting corrections

$$\tilde{\nu}_k = K_h * (\Psi_k \cdot (S^N(\tilde{\nu}_{k-1}))) \ , \quad 1 \leq k \leq n \ , \quad \text{with } \tilde{\nu}_0 = \pi.$$

Just as δ_k, we define $\tilde{\delta}_k = \sup_{\mathbf{x} \in \mathbb{R}^{n_x}} \Psi_k(\mathbf{x}) / \langle \tilde{\nu}_{k-1}, \Psi_k \rangle$. Let $\mathcal{S}_k$ denote the σ–field generated by the random variables simulated in the transition from $\tilde{\nu}_0$ to $\tilde{\nu}_k$. Using inequality (12.3.12) and a generalisation of density estimation error estimates yields

Proposition 12.3.2. *If ν_{k-1} is absolutely continuous, with density given by $\dfrac{\mathrm{d}\nu_{k-1}}{\mathrm{d}\mathbf{x}} \in W^{2,1}$, then the local mean error induced by progressive*

weighting correction satisfies

$$\mathbb{E}[\,\|\tilde{\nu}_k - \nu_k\|_1 \mid \mathcal{S}_k] \leq \tilde{\alpha}\, h^2 + \tilde{\delta}_k\, [\frac{\tilde{\beta}}{\sqrt{N\, h^{n_x}}} + \|\tilde{\nu}_{k-1} - \nu_{k-1}\|_1\,] \;, \quad (12.3.13)$$

with $\tilde{\alpha} = \alpha(K, \Psi_k \cdot \nu_{k-1})$, *and* $\tilde{\beta} = \beta(K, \tilde{\nu}_{k-1})$.

Inequality (12.3.13) shows that to minimise the local error, at each time step, we have to choose Ψ_k such that $\tilde{\delta}_k$ is well controlled.

Progressive rejection correction

Let $\bar{\nu}_k$ denote the approximation of ν_k by successive rejection corrections

$$\bar{\nu}_k = \Psi_k \cdot (K_h * (S^N(\bar{\nu}_{k-1}))) \;, \quad 1 \leq k \leq n \;, \quad \text{with } \bar{\nu}_0 = \pi.$$

By analogy with $\tilde{\delta}_k$, we define $\bar{\delta}_k = \sup_{\mathbf{x} \in \mathbb{R}^{n_x}} \Psi_k(\mathbf{x}) / \langle \bar{\nu}_{k-1}, \Psi_k \rangle$. Using inequality (12.3.12) and classical density estimation error estimates yields

Proposition 12.3.3. *If* ν_{k-1} *is absolutely continuous, with density given by* $\dfrac{\mathrm{d}\nu_{k-1}}{\mathrm{d}\mathbf{x}} \in W^{2,1}$, *then the local mean error induced by progressive rejection correction satisfies*

$$\mathbb{E}[\,\|\bar{\nu}_k - \nu_k\|_1 \mid \mathcal{S}_k] \leq \bar{\delta}_k\, [\bar{\alpha}\, h^2 + \frac{\bar{\beta}}{\sqrt{N\, h^{n_x}}} + \|\bar{\nu}_{k-1} - \nu_{k-1}\|_1\,] \;, \quad (12.3.14)$$

with $\bar{\alpha} = \alpha(K, \nu_{k-1})$ *and* $\bar{\beta} = \beta(K, \bar{\nu}_{k-1})$.

In addition, the number of random variables required by progressive rejection correction (with the decomposition $(\Psi_1, \cdots, \Psi_n)$ introduced above) is a random number with mean $N\,(\bar{\delta}_1 + \cdots + \bar{\delta}_n)$. Recall that the mean number of simulations required by the direct rejection correction method is $N\,\delta = N\,(\delta_1 \cdots \delta_n)$. We can see here that progressive correction allows us to change the product $(\delta_1 \cdots \delta_n)$ into the sum $(\bar{\delta}_1 + \cdots + \bar{\delta}_n)$, which implies in general a lower computing cost.

12.3.3 Adaptive choice of the decomposition

An adaptive method is described to chose the decomposition $(\Psi_1, \cdots, \Psi_n)$. For simplicity, we consider only the case of an additive Gaussian measurement noise with covariance matrix R_t.

Assume that the decomposition $(\Psi_1, \cdots, \Psi_{k-1})$ is already given up to step $k-1 < n$, and that the first $(k-1)$ sub-corrections have already been computed. We are then, given $\bar{\nu}_{k-1}$, in the case of progressive rejection correction, and given $\tilde{\nu}_{k-1}$, in the case of progressive weighting correction, respectively. The aim is to find Ψ_k such that $\bar{\delta}_k \leq \delta_{\max}$, and $\tilde{\delta}_k \leq \delta_{\max}$ respectively, where $\delta_{\max} > 1$ is a control parameter given by the

user and determining the maximum cost coefficient of each sub-correction step. Recall that

$$\frac{\mathrm{d}\tilde{\nu}_k}{\mathrm{d}\mathbf{x}}(\mathbf{x}) = \frac{\mathrm{d}(K_h * (\Psi_k \cdot (S^N(\tilde{\nu}_{k-1}))))}{\mathrm{d}\mathbf{x}}(\mathbf{x}) = \sum_{i=1}^{N} \tilde{\omega}_k^{(i)} K_h(\mathbf{x} - \tilde{\mathbf{x}}_{k-1}^{(i)}) ,$$

$$\frac{\mathrm{d}\bar{\nu}_k}{\mathrm{d}\mathbf{x}}(\mathbf{x}) = \frac{\mathrm{d}(\Psi_k \cdot (K_h * (S^N(\bar{\nu}_{k-1}))))}{\mathrm{d}\mathbf{x}}(\mathbf{x}) \propto \sum_{i=1}^{N} \Psi_k(\mathbf{x}) K_h(\mathbf{x} - \bar{\mathbf{x}}_{k-1}^{(i)}) ,$$

where $(\tilde{\mathbf{x}}_{k-1}^{(1)}, \cdots, \tilde{\mathbf{x}}_{k-1}^{(N)})$ i.i.d. $\sim \tilde{\nu}_{k-1}$ and $(\bar{\mathbf{x}}_{k-1}^{(1)}, \cdots, \bar{\mathbf{x}}_{k-1}^{(N)})$ i.i.d. $\sim \bar{\nu}_{k-1}$ respectively. Notice that we are interested in Ψ_k or any other proportional function. The choice of Ψ_k is made among functions of the family

$$\Psi_k(\mathbf{x}) = \exp\{-\frac{1}{2\,\lambda_k} (\mathbf{y}_t - H_t(\mathbf{x}))^T R_t^{-1} (\mathbf{y}_t - H_t(\mathbf{x}))\} , \qquad (12.3.15)$$

where $\mathbf{y}_t$ is the current measurement, and $\lambda_k > 0$ is a parameter to be determined. We make the following Monte Carlo approximations:

$$\tilde{\delta}_k = \frac{1}{\langle \tilde{\nu}_{k-1}, \Psi_k \rangle} \approx \frac{1}{\sum_{i=1}^{N} \Psi(\tilde{\mathbf{x}}_{k-1}^{(i)})} , \quad \bar{\delta}_k = \frac{1}{\langle \bar{\nu}_{k-1}, \Psi_k \rangle} \approx \frac{1}{\frac{1}{N} \sum_{i=1}^{N} \Psi_k(\bar{\mathbf{x}}_{k-1}^{(i)})} .$$

Henceforth, the case of rejection correction and weighting correction are identical, so we shall focus on only one of them. The choice of λ_k is then determined by the equation

$$\min_{1 \le i \le N} \Psi_k(\bar{\mathbf{x}}_{k-1}^{(i)}) = 1/\delta_{\max} . \qquad (12.3.16)$$

It appears here that $\delta_{\max}$ is necessarily larger than 1, since Ψ_k is smaller than 1. Condition (12.3.16) is chosen because it automatically implies $\bar{\delta}_k \le \delta_{\max}$ and is sufficient to determine λ_k. Indeed, it follows from (12.3.16) that

$$\lambda_k = \frac{\max_{1 \le i \le N} (\mathbf{y}_t - H_t(\bar{\mathbf{x}}_{k-1}^{(i)}))^T R_t^{-1} (\mathbf{y}_t - H_t(\bar{\mathbf{x}}_{k-1}^{(i)}))}{2 \log(\delta_{\max})} .$$

Notice that the sequence $(\lambda_1, \cdots, \lambda_{k-1}, \cdots)$ is decreasing. Indeed, the particles concentrate at each sub-correction step closer to the real state, which implies that $\max_{1 \le i \le N} (\mathbf{y}_t - H_t(\bar{\mathbf{x}}_{k-1}^{(i)}))^T R_t^{-1} (\mathbf{y}_t - H_t(\bar{\mathbf{x}}_{k-1}^{(i)}))$ decreases. The size n of the decomposition is a-priori unknown and determined during the implementation of the algorithm. The algorithm runs as long as $\lambda_k \ge 1$. If $\lambda_k < 1$, then n is set to k, and $\Psi_n = \Psi/(\Psi_1 \cdots \Psi_{n-1})$.

Given $\delta_{\max}$ and a maximum number $n_{\max}$ of sub-correction steps, computing Ψ_k at each sub-correction step is done in the following two steps.

1. Compute $\lambda_k = \dfrac{\max_{1\leq i\leq N}(H_t(\bar{\mathbf{x}}_{k-1}^{(i)}) - \mathbf{y}_t)^T R_t^{-1} (H_t(\bar{\mathbf{x}}_{k-1}^{(i)}) - \mathbf{y}_t)}{2\log(\delta_{\max})}$.
2. If $\lambda_k \geq 1$, and $k \leq n_{\max}$, then compute Ψ_k using (12.3.15), otherwise set $n = k$, and $\Psi_n = \Psi/(\Psi_1 \cdots \Psi_{k-1})$.

Remark 3. More generally, i.e. if the measurement noise is not necessarily Gaussian, it is still possible to implement the progressive correction principle as follows. Indeed, let

$$\Psi_k(\mathbf{x}) = [\Psi(\mathbf{x})]^{1/\lambda_k} = [g_t(\mathbf{y}_t - H_t(\mathbf{x}))]^{1/\lambda_k} ,$$

where $\mathbf{y}_t$ is the current measurement, and $\lambda_k > 0$ is a parameter to be determined. The choice of λ_k is determined by (12.3.16) again, which yields

$$\lambda_k = \frac{\max_{1\leq i\leq N}[-\log g_t(\mathbf{y}_t - H_t(\bar{\mathbf{x}}_{k-1}^{(i)}))]}{\log(\delta_{\max})} .$$

Otherwise, the algorithm is the same as in the Gaussian case.

12.4 The local rejection regularised particle filter (L2RPF)

The L2RPF presented below uses regularisation, with the Epanechnikov kernel defined in (12.2.5) which is optimal in the L^2 sense, and the rescaling procedure (12.2.8). This filter is based on the kernel filter (KF) (Hürzeler and Künsch 1998), a local rejection method that produces samples according to (12.4.22) below and that is faster than the classical rejection method; see Section 12.2.2. Observing that the Epanechnikov kernel has compact support, the maximum $c_t^{(i)}$ of the likelihood is taken around each particle $\mathbf{x}_{t|t-1}^{(i)}$, and the computational cost gain is $\sum_{i=1}^{N} c_t^{(i)}/(N \max_{1\leq i\leq N} c_t^{(i)})$. However, this cost remains high, especially when the particles are dispersed or when the variance of the measurement noise is small.

12.4.1 Description of the filter

We introduce the computing control parameter $\alpha_t \in [0, 1]$. At each time step, this adaptive parameter is computed so as to take into account the

computing capability by means of the evaluation of the acceptance probability of the rejection loop AL below. It allows us to alternate weighted sample methods such as post–RPF, which generates samples according to (12.4.23) below, and local rejection methods such as KF, which generates samples according to (12.4.22) below. The proposed L2RPF allows a precise correction step in a given computing time. Given a prediction sample $(\mathbf{x}_{t|t-1}^{(1)}, \cdots, \mathbf{x}_{t|t-1}^{(N)})$ with covariance matrix $S_t = A_t A_t^T$, and a scalar α_t, we use the following algorithm AL to generate a corrected sample independently for all i.

1. Generate $I \in \{1, \cdots, N\}$, with $\mathbb{P}(I = j) \propto c_t^{(j)}(\alpha_t)$.
2. Generate $\varepsilon \sim K$, Epanechnikov kernel (12.2.5), and U uniformly on $[0, 1]$.
3. Put $\mathbf{X} = \mathbf{x}_{t|t-1}^{(I)} + h\, A_t\, \varepsilon$, with $h = h_{\text{opt}}$ given by (12.2.6).
4. If $\Psi_t(\mathbf{X}) \geq \alpha_t\, c_t^{(I)}(\alpha_t)\, U$, then return $\mathbf{x}_{t|t}^{(i)} = \mathbf{X}$, otherwise go to step 1.

The coefficients $c_t^{(i)}(\alpha_t)$ are computed below, and satisfy

$$c_t^{(i)}(\alpha_t) \geq \sup_{\mathbf{x} \in \Sigma_i(\alpha_t)} \Psi_t(\mathbf{x}) \,, \tag{12.4.17}$$

where the supremum is taken on an ellipsoid centered at the particle $\mathbf{x}_{t|t-1}^{(i)}$

$$\Sigma_i(\alpha_t) = \{\mathbf{x} \in \mathbb{R}^{n_x} \,:\, (\mathbf{x} - \mathbf{x}_{t|t-1}^{(i)})^T\, S_t^{-1}\, (\mathbf{x} - \mathbf{x}_{t|t-1}^{(i)}) \leq \alpha_t^2\, h^2\} \,. \tag{12.4.18}$$

Proposition 12.4.1. *The L2RPF algorithm produces a sample according to an absolutely continuous probability distribution $\hat{\pi}_{t|t}^{\alpha_t}$, with density*

$$\frac{\mathrm{d}\hat{\pi}_{t|t}^{\alpha_t}}{\mathrm{d}\mathbf{x}}(\mathbf{x}) \propto \sum_{i=1}^{N} c_t^{(i)}(\alpha_t)\, \min(1, \frac{\Psi_t(\mathbf{x})}{\alpha_t\, c_t^{(i)}(\alpha_t)}) K_h(A_t^{-1}\, (\mathbf{x} - \mathbf{x}_{t|t-1}^{(i)})). \tag{12.4.19}$$

Indeed, the probability distribution of the random variable $\mathbf{X}$ generated by AL is characterised as follows, for any test function ϕ defined on $\mathbb{R}^{n_x}$.

$$\mathbb{E}_{I,U,\varepsilon}[\phi(\mathbf{X})] = \mathbb{E}[\phi(\mathbf{x}_{t|t-1}^{(I)} + h\,A_t\,\varepsilon)\,\mathbf{1}_{(\Psi_t(\mathbf{x}_{t|t-1}^{(I)} + h\,A_t\,\varepsilon) \geq \alpha_t\,c_t^{(I)}(\alpha_t)\,U)}]$$

$$\propto \sum_{i=1}^{N} c_t^{(i)}(\alpha_t) \int \phi(\mathbf{x}_{t|t-1}^{(i)} + h\,A_t\,\mathbf{z})\ \min(1, \frac{\Psi_t(\mathbf{x}_{t|t-1}^{(i)} + h\,A_t\,\mathbf{z})}{\alpha_t\,c_t^{(i)}(\alpha_t)})\ K(\mathbf{z})\mathrm{d}\mathbf{z}$$

$$\propto \sum_{i=1}^{N} c_t^{(i)}(\alpha_t) \int \phi(\mathbf{x})\ \min(1, \frac{\Psi_t(\mathbf{x})}{\alpha_t\,c_t^{(i)}(\alpha_t)})\ K(\frac{1}{h}\,A_t^{-1}\,(\mathbf{x} - \mathbf{x}_{t|t-1}^{(i)}))\ \mathrm{d}\mathbf{x}$$

$$\propto \sum_{i=1}^{N} c_t^{(i)}(\alpha_t) \int \phi(\mathbf{x})\ \min(1, \frac{\Psi_t(\mathbf{x})}{\alpha_t\,c_t^{(i)}(\alpha_t)})\ K_h(A_t^{-1}\,(\mathbf{x} - \mathbf{x}_{t|t-1}^{(i)}))\,\mathrm{d}\mathbf{x}\ .$$

The next proposition computes the acceptance probability of the algorithm, that is, the probability that a sample goes out of AL.

Proposition 12.4.2. *The acceptance probability P_a of L2RPF is*

$$P_\mathrm{a}(\alpha_t) = c \sum_{i=1}^{N} c_t^{(i)}(\alpha_t) \int \min(1, \frac{\Psi(\mathbf{x}_{t|t-1}^{(i)} + hA_t\mathbf{z})}{\alpha_t\,c_t^{(i)}(\alpha_t)})\ K(\mathbf{z})\,\mathrm{d}\mathbf{z} \quad (12.4.20)$$

$$\approx c \sum_{i=1}^{N} c_t^{(i)}(\alpha_t)\ \min(1, \frac{\Psi(\mathbf{x}_{t|t-1}^{(i)})}{\alpha_t\,c_t^{(i)}(\alpha_t)})\ , \quad (12.4.21)$$

with $c = 1/\sum_{i=1}^{N} c_t^{(i)}(\alpha_t)$.

Equation (12.4.20) is derived as was (12.4.19), and approximation (12.4.21) is obtained using an expansion of $\Psi_t(\mathbf{x}_{t|t-1}^{(i)} + h\,A_t\,\mathbf{z})$ around $h = 0$. This approximation is in general precise; see simulations in Figure 12.4. Notice that with the choice made in Section 12.4.2 below for the coefficients $(c_t^{(i)}(\alpha_t))_{1\leq i\leq N}$, the probability of acceptance $P_\mathrm{a}(\alpha_t)$ decreases when α_t increases. If $\alpha_t = 1$, then $\Sigma_i(\alpha_t)$ coincides with the support of $\mathbf{x} \mapsto K_h(A_t^{-1}\,(\mathbf{x} - \mathbf{x}_{t|t-1}^{(i)}))$; hence,

$$\hat{\pi}_{t|t}^{\alpha_t=1}(\mathbf{x}) \propto \sum_{i=1}^{N} \Psi_t(\mathbf{x})\ K_h(A_t^{-1}\,(\mathbf{x} - \mathbf{x}_{t|t-1}^{(i)}))\ , \quad (12.4.22)$$

which is exactly the KF density. But in this case P_a is minimal, i.e. the computing cost is maximal. On the other hand, if $\alpha_t = 0$, then $\Sigma_i(\alpha_t)$ reduces to the current particle $\mathbf{x}_{t|t-1}^{(i)}$, and taking equality in (12.4.17) yields

$c_t^{(i)}(\alpha_t) \propto \omega_t^{(i)}$; hence,

$$\hat{\pi}_{t|t}^{\alpha_t=0}(\mathbf{x}) \propto \sum_{i=1}^{N} \omega_t^{(i)} \, K_h(A_t^{-1}(\mathbf{x} - \mathbf{x}_{t|t-1}^{(i)})) \,, \tag{12.4.23}$$

which is exactly the post–RPF density, with whitening. In this case $P_{\mathrm{a}} = 1$, i.e. the computing cost is low. At each time step, the choice of α_t is done in the following way. Given a coarse discretisation of $[0, 1]$, we take the maximal value of α_t such that $P_{\mathrm{a}}(\alpha_t) \geq P_{\mathrm{a}}^{\min}$, where $P_{\mathrm{a}}^{\min}$ is given by the computing capability. The higher α_t is, the better the correction. When α_t is chosen, we put $N_e = N/P_{\mathrm{a}}(\alpha_t)$ the number of test samples that enter in the loop AL. For the problems presented in section 12.5, α_t is close to 0 for the first measurements, then increases to 1 when the particles concentrate on more likely regions of the state-space; see Figure 12.4. The L2RPF generalises both the post–RPF and the KF. The practical advantage of the L2RPF compared with the post–RPF is that it allows us to reduce significantly the number of particles. For example, the performances of the tracking problems are identical with $N = 1000$ in place of $N = 5000$. The additional cost is small. It was also observed that the variance of the particle system is smaller with the L2RPF (owing to the better correction step).

We now present a fast method to compute $c_t^{(i)}(\alpha_t)$.

12.4.2 Computing the coefficient $c_t^{(i)}(\alpha_t)$

By Lagrangian methods, we see that the coordinates $(x_1, \cdots, x_j, \cdots, x_{n_x})$ of any point in the ellipsoid $\Sigma_i(\alpha_t)$ centered on $\mathbf{x}_{t|t-1}^{(i)}$, see (12.4.18), verifies for all $j = 1, \cdots, n_x$,

$$x_j^{(i),\min} = x_j^{(i)} - \alpha_t h \sqrt{S_{jj}} \leq x_j \leq x_j^{(i)} + \alpha_t h \sqrt{S_{jj}} = x_j^{(i),\max} \tag{12.4.24}$$

where S_{jj} is the j^{th} diagonal term of S. Indeed, the optimisation of x_j over $\Sigma_i(\alpha_t)$, with the constraint, $\mathbf{x}^T S^{-1} \mathbf{x} < \alpha^2 h^2$, gives the system

$$\frac{\partial L}{\partial \mathbf{x}} = \Gamma_j + 2\lambda S^{-1}\mathbf{x} = 0 \,, \quad \mathbf{x}^T S^{-1} \mathbf{x} = \alpha^2 h^2 \,, \tag{12.4.25}$$

where λ is the Lagrange multiplier, and $\Gamma_j^T = (0, \cdots, 1, \cdots, 0)$ with 1 in the j^{th} position.
The solution of (12.4.25) is $\mathbf{x} = \pm \alpha_t h S_{\cdot j} / \sqrt{S_{jj}}$, where $S_{\cdot j}$ is the j^{th} column of S. Let $C_i(\alpha_t)$ be the hypercube, $\Sigma_i(\alpha_t) \subset C_i(\alpha_t)$,

$$C_i(\alpha_t) = \{\mathbf{x} | \, x_j^{(i),\min} \leq x_j \leq x_j^{(i),\max}, 1 \leq j \leq n_x\} \,. \tag{12.4.26}$$

$c_t^{(i)}(\alpha_t)$ will be chosen as the maximum of Ψ on $C_i(\alpha_t)$. We can also take the hypercube containing the ellipsoid $\Sigma_i(\alpha_t)$, i.e. the smallest hypercube

with faces orthogonal to the eigenvectors of S_t. Assume that the measurement function H_k, $k = 1, \cdots, n_y$ is monotone in each coordinate in the neighbourhood of the current particle. For example, if we measure an angle $H_k(\mathbf{x}) = \arctan(x_1/x_2)$, H_k increases when x_1 increases and H_k decreases when x_2 increases (if x_1, $x_2 > 0$). The extreme values of H_k on the hypercube $C_i(\alpha_t)$ are for $k = 1, \cdots, n_y$,

$$\forall \mathbf{x} \in C_i(\alpha_t), \quad H_k^{\min} = H_k(\mathbf{x}_m^{\text{extr}}) \leq H_k(\mathbf{x}) \leq H_k(\mathbf{x}_M^{\text{extr}}) = H_k^{\max}, \quad (12.4.27)$$

where $\mathbf{x}_{(\,)}^{\text{extr}}$ is one of the vertices of $C_i(\alpha_t)$ with coordinates $\mathbf{x}_j^{(i),\min}$ or $\mathbf{x}_j^{(i),\max}$ (12.4.24).

Assume that the components $(v_1, \cdots, v_{n_y})$ of the measurement noise v are independent, (otherwise it suffices to whiten the measurement vector) with densities $(g_1, \cdots, g_{n_y})$ decreasing around the origin. It can be seen that the maximum of the likelihood on $C_i(\alpha_t)$ verifies

$$\sup_{x \in \Sigma_i(\alpha_t)} \Psi(\mathbf{x}) \leq \sup_{x \in C_i(\alpha_t)} \prod_{k=1}^{n_y} g_k(\mathbf{y}_k - H_k(\mathbf{x})) \quad (12.4.28)$$

$$= \prod_{k=1}^{n_y} g_k(\mathbf{y}_k - H_k^{\text{extr}}(\mathbf{x})) = c_t^{(i)}(\alpha_t), \quad (12.4.29)$$

where $H_k^{\text{extr}} = H_k^{\min}$ if $\mathbf{y}_k \leq H_k^{\min}$, $H_k^{\text{extr}} = \mathbf{y}_k$ if $H_k^{\min} \leq \mathbf{y}_k \leq H_k^{\max}$ and $H_k^{\text{extr}} = H_k^{\max}$ if $\mathbf{y}_k \geq H_k^{\max}$.

12.5 Applications to tracking problems

We present three two-dimensional-tracking problems to which L2RPF is applied. Classical post–RPF has also been applied, for which the results are similar to the L2RPF in terms of estimator standard deviation (std), as defined below. But it was observed that the variances of the particle clouds are larger than when L2RPF was used, owing to the correction error. To estimate the error committed by approximate filters on each coordinate j of the state, we do the following Monte Carlo approximations with M runs.

$$\{\mathbb{E}[\hat{\mathbf{x}}_{j,t} - \mathbb{E}(\hat{\mathbf{x}}_{j,t})]^2\}^{1/2} \approx \{\frac{1}{M}\sum_{k=1}^{M}[\hat{\mathbf{x}}_{j,t|t}^{(k)} - \frac{1}{M}\sum_{k=1}^{M}\hat{\mathbf{x}}_{j,t|t}^{(k)}]^2\}^{1/2} \quad \text{[std]}$$

$$\mathbb{E}(\hat{\mathbf{x}}_{j,t}) - \mathbb{E}(\mathbf{x}_{j,t}) \approx \frac{1}{M}\sum_{k=1}^{M}[\hat{\mathbf{x}}_{j,t|t}^{(k)} - \mathbf{x}_{j,t}], \quad \text{[bias]}$$

where $\mathbf{x}_{j,t}$ is the j^{th} coordinate of the real state $\mathbf{x}_t$ (x–y position and velocity) of the target, and $\hat{\mathbf{x}}_{j,t|t}^{(k)}$ is the j^{th} coordinate of the filter estimate for the k^{th} Monte Carlo run. The number of Monte Carlo runs is $M = 50$.

The filters initialisation, $\hat{\mathbf{x}}_{0|-1}$ is a Gaussian variable centered on the true state $\mathbf{x}_0$, with covariance matrix $P_{0|0}$. The number of particles used for the L2RPF is $N = 1000$. The acceptance probability is $P_{\mathrm{a}}^{\min} = 0.2$, and the associated computing cost is about 20 times greater than if the extended Kalman filter (EKF) is employed. In the problems considered below, there is no dynamic noise, and we can easily compute the Cramer–Rao lower bound (CRLB). This gives the minimal std of any unbiased estimator. For another application of the CRLB in particle filtering, see Bergman (2001: this volume).

12.5.1 Range and bearing

The target follows a uniform straight motion (USM). Noisy measurements of the distance between the origin and the target, and the angle between the horizontal line and the line of sight are available. That is, we have $H_t(\mathbf{x}) = \begin{pmatrix} \sqrt{x_1^2 + x_3^2} \\ \arctan(x_3/x_1) \end{pmatrix}$, where (x_1, x_3) denote the horizontal and vertical positions, and (x_2, x_4) the horizontal and vertical velocities.

Progressive correction

We compare the performance of the post–RPF (with $N = 5000$ particles), and the post–RPF with progressive correction (PC) (with $N = 1000$ particles) with the CRLB, as a function of the range measurement noise std. The observation time is $T = 500s$ with inter-observation time Δ=5s, and the initial state is $\mathbf{x}_0$=(50km, 0m/s, 50km, 5m/s) with $P_{0|0}$=[diag(0.5km, 50m/s, 0.5km, 50m/s)]^2. For each value of the range measurement noise std on abscissa, we report on ordinate the path error of the horizontal position and velocity estimates, averaged between time $20s$ and $500s$. The std of angle measurement noise is 1 degree. The maximum cost coefficient is $\delta_{\max} = 10$, and the maximum number of sub-correction steps is $n_{\max} = 25$.

In Figure 12.2 we report the std of the position and velocity estimates. We can see that PC highly improves the performance of the post–RPF, since it gives some efficient results (close to CRLB) uniformly with the measurement noise std. In Figure 12.3 (left) we report the ratio of the computing time of the post–RPF with and without PC, w.r.t. the computing time of the EKF, for each value of the measurement noise std. We can observe that when PC is used, the computing time decreases when the std of the measurement noise increases, from 24 times greater than for the EKF to less than 12 times. In the same way, we can observe in Figure 12.3 (right) that $n_{\max} = 25$ is only reached for the first measurements, for problems with small measurement noise (std=0.1m, 3.3m) where δ is known to be high. Hence, with PC the computing time is automatically adjusted to the problem's complexity.

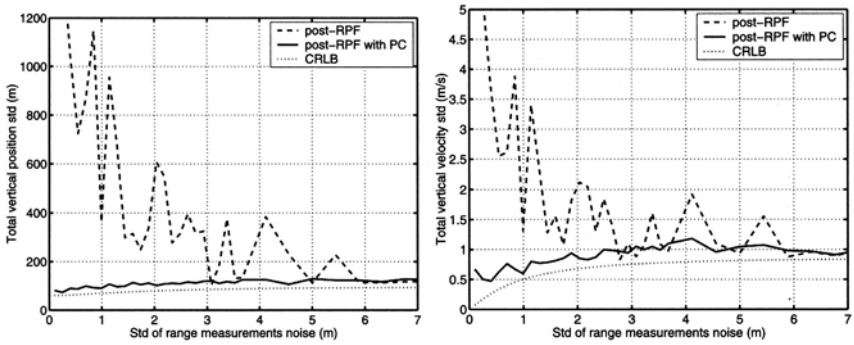

Figure 12.2. Vertical position and velocity std.

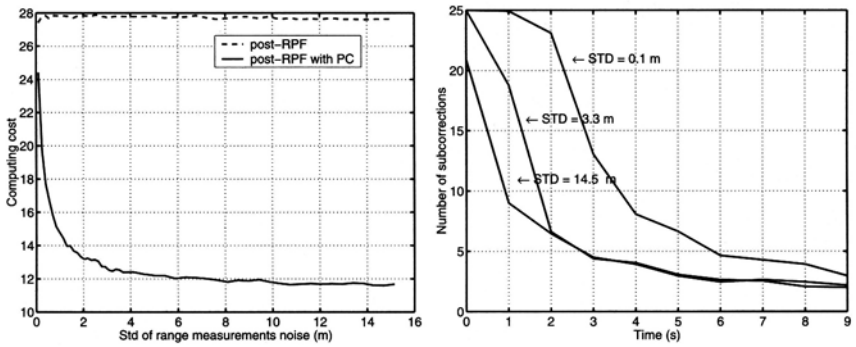

Figure 12.3. Evolution of the computing cost.

L2RPF

Below, the L2RPF and the EKF are compared with the CRLB. Here, the initial state is $\mathbf{x}_0$=(5km, -20m/s, 5km, -20m/s) with $P_{0|0}$=[diag(0.5km, 50m/s, 0.5km, 50m/s)]^2, and the observation time is T=200s with Δ=1s. The std of range and angle measurements noise is equal to 1m (case of accurate measurements) and 1 degree, respectively.

We can observe in Figure 12.4 that the control parameter is close to 0 for the first measurements (post–RPF) and increases to 1 (KF). Notice that the theoretical P_a is close to the empirical, around the value 0.65. We observe in Figure 12.5 that the bias is close to 0, and in Figure 12.6 that unlike the EKF, the L2RPF converges rapidly to the CRLB.

12.5.2 Bearings–only

The target has a USM initialised on $\mathbf{x}_0$=(10km, -10m/s, 10km, 10m/s), with $P_{0|0}$=[diag(5km, 30m/s, 5km, 30m/s)]^2. The observer is initially at

Time (s) / Pa	10	50	120	160	200
Emp.	0.333	0.634	0.555	0.709	0.673
Theo.	0.210	0.631	0.554	0.708	0.673

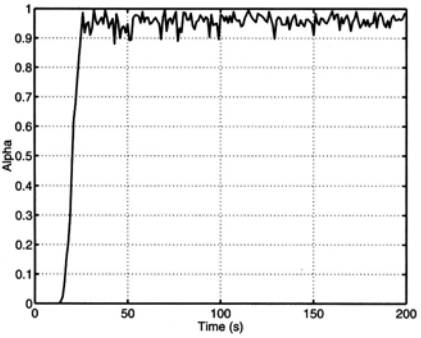

Figure 12.4. Acceptance probability P_a and control parameter α.

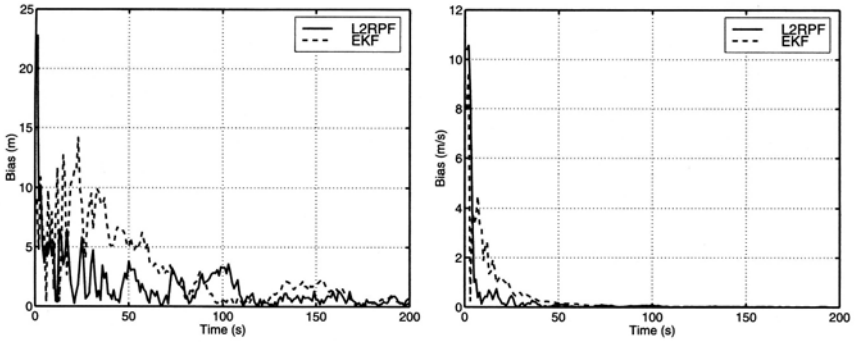

Figure 12.5. Horizontal position and velocity bias.

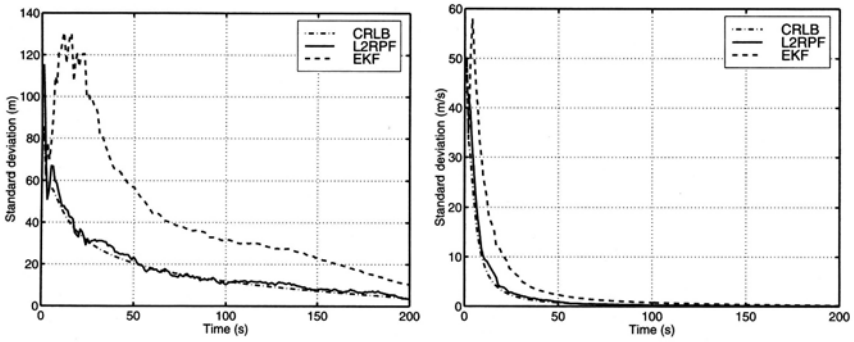

Figure 12.6. Horizontal position and velocity std.

the origin with USM (50m/s, 0m/s), then maneuvers at time $t = 100s$ to follow another USM (-50m/s, 50m/s) until $T = 200s$. The observer measures every Δ=1s a noisy angle with std 0.5 degree.

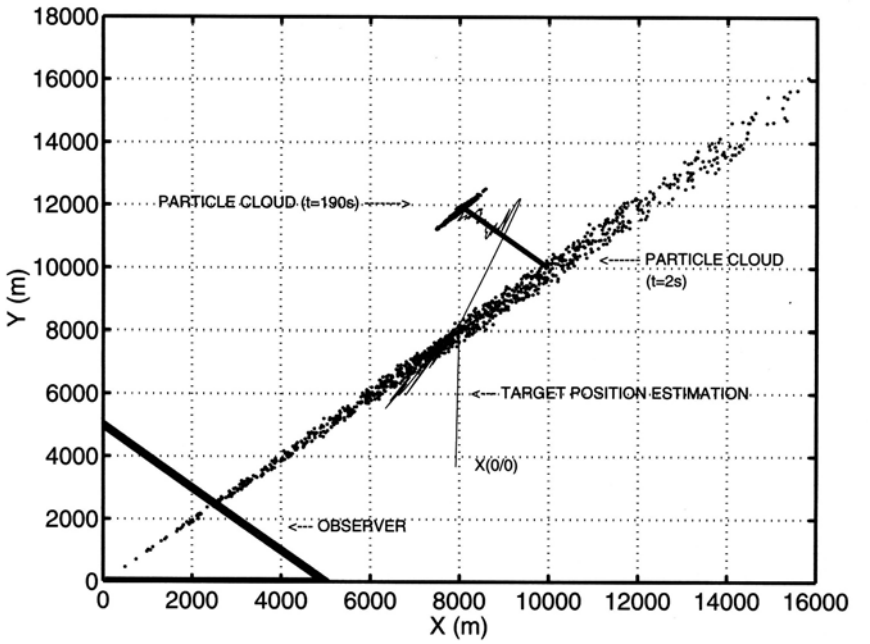

Figure 12.7. Real and estimated paths.

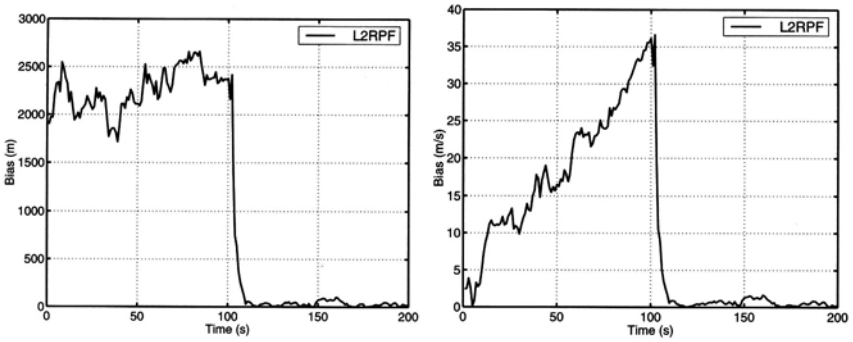

Figure 12.8. Horizontal position and velocity bias.

Figure 12.7 shows the configuration and the trajectory estimate (center of the cloud) of the L2RPF. As can be seen in Figure 12.9, the std of the horizontal position and velocity of the target is close to the CRLB. The corresponding bias is null after the observer maneuvers, see Figure 12.8.

Recall that this model is not identifiable until the observer maneuvers, see (Nardone and Aidala 1981), and indeed we have observed the divergence of the EKF in 5 runs out of 50, hence results for the EKF are not reported here.

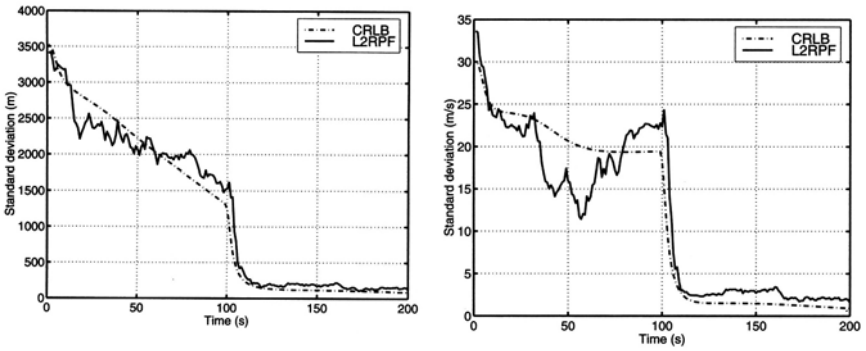

Figure 12.9. Horizontal position and velocity std.

12.5.3 Multiple model particle filter (MMPF)

By means of the formalism of interacting multiple model (IMM) (Blom and Bar-Shalom 1988), the dynamic model of the target is estimated among some given models. This is a case of multi–modality. We suppose that $(\theta_t)_{t\geq 0}$ is a Markov chain with finite state-space S, and transition probability matrix $p = (p(k,\ell))_{k,\ell\in S}$, which determines the dynamic model between time t and $(t+1)$. Therefore, we can apply the theory of particle filters with the new augmented state $(\mathbf{x}_t, \theta_t)$. Assuming that θ_t is independent of $\mathbf{x}_{t-1}$, given θ_{t-1}, and introducing the transition probability kernel

$$\mathbb{P}(\mathbf{x}_t \in \mathrm{d}\mathbf{x}' \mid \theta_t = \ell, \mathbf{x}_{t-1} = \mathbf{x}) = Q_t(\ell, \mathbf{x}, \mathrm{d}\mathbf{x}') \ ,$$

the prediction step is given by

$$\begin{aligned} &\mathbb{P}(\mathbf{x}_t \in \mathrm{d}\mathbf{x}', \theta_t = \ell \mid \mathbf{y}_{0:t-1}) \\ &= \sum_{k\in S} p(k,\ell) \int Q_t(\ell, \mathbf{x}, \mathrm{d}\mathbf{x}')\, \mathbb{P}(\mathbf{x}_{t-1} \in \mathrm{d}\mathbf{x}, \theta_{t-1} = k \mid \mathbf{y}_{0:t-1}) \ . \end{aligned} \tag{12.5.30}$$

If we have a sample $(\mathbf{x}_{t-1}^{(1)}, \theta_{t-1}^{(1)}, \cdots, \mathbf{x}_{t-1}^{(N)}, \theta_{t-1}^{(N)})$, the following algorithm produces a predicted sample according to (12.5.30): independently for all i

1. Generate $\theta_t^{(i)} \in S$, with $\mathbb{P}[\theta_t^{(i)} = \ell] = p(\theta_{t-1}^{(i)}, \ell)$.
2. Generate $\mathbf{x}_{t|t-1}^{(i)} \sim Q_t(\theta_t^{(i)}, \mathbf{x}_{t-1}^{(i)}, \cdot)$.

The correction step is done with L2RPF. In our simulation, the target can have 2 dynamic models: USM and turn with constant angular velocity $\Omega = 0.006$ rad/s. Figure 12.10 shows the configuration. The observer located at the origin measures bearing (std=1 degree) and range (std=20m) every 10s. The target is in USM during 600s, then in turn during 800s, and finally in USM until T=2000s. The initial state is set as follows: $\mathbf{x}_0$=(50km,0m/s,50km,5m/s) with $P_{0|0}$=[diag(450m, 63m/s, 42m,

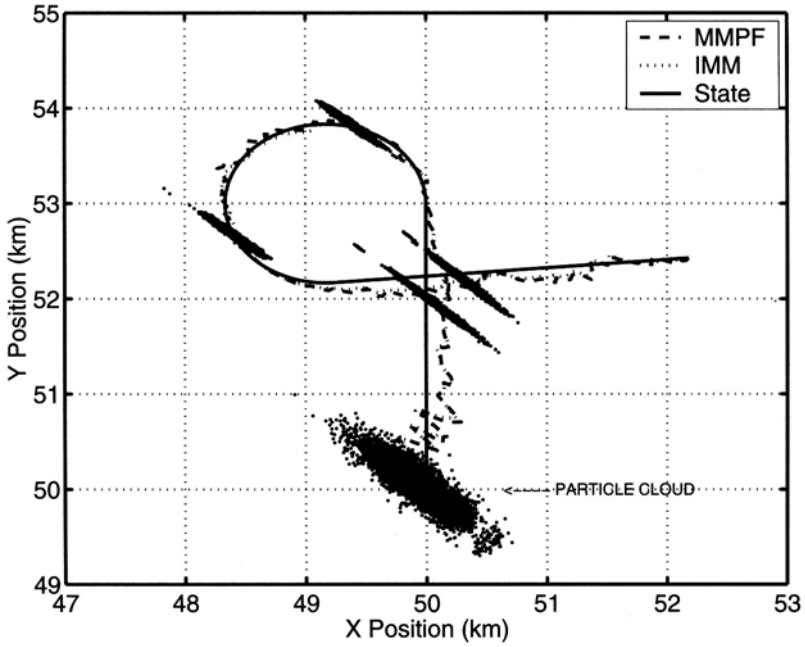

Figure 12.10. Real and estimated paths.

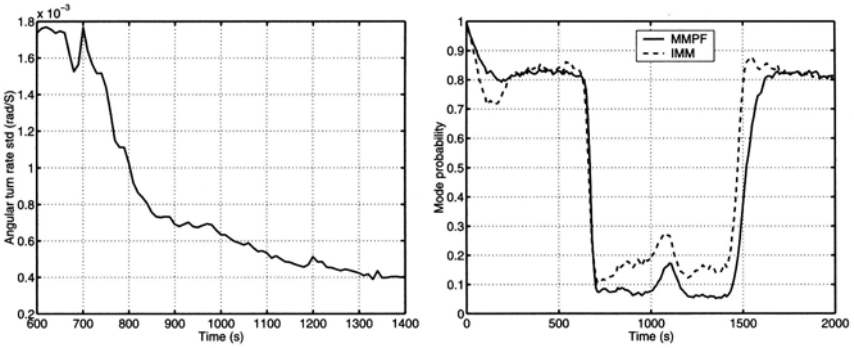

Figure 12.11. Angular turn rate and mode probability estimates.

60m/s)]^2. The initial USM mode probability is $\mathbb{P}(\theta_0 = 1) = 0.99$, and the transition probability matrix is $\begin{pmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{pmatrix}$. Classic IMM filter and MMPF are compared. MMPF estimates the angular turn rate (dimension of the state is 5). The IMM knows this rate (otherwise, for this context, IMM is not stable). Nevertheless, the behavior of the two filters is similar.

We can see in Figure 12.11 that the probability of the USM mode and the angular turn rate are well estimated by MMPF.