Linear and nonlinear filtering for state space models

Master course notes

UdelaR 2010

Thierry CHONAVEL
thierry.chonavel@telecom-bretagne.eu

December 2010

Contents

Chapter 1

Introduction

Filtering prediction and smoothing are important classical tasks of signal processing. In the case of linear models, the Kalman filter enables propagating the mean and covariance of the state variable of the model conditional to observations. In addition, in the Gaussian case the Kalman filter which supplies the minimum mean square error linear approximation yields in the minimum mean square error approximation of the state since in the Gaussian case the linear regression matches the conditional expectation.

First attempts to deal with nonlinear models have consisted in considering local first order Taylor approximations of state and observation equations, leading thus to the Extended Kalman Filter (EKF). Alternatively, in the mid nineties Julier and Uhlmann have proposed, instead of using linear approximations of the model functions, to apply the nonlinear functions to a set of points that catch the first and second order statistics of the variable and of its transform. These points are chosen so as to achieve unbiased estimation of the transformed mean and covariance. There work have lead to the unscented Kalman filter. Kalman filter, EKF and UKF will be presented in the next chapter.

However, above approaches rely on linear or Gaussian approximations. In order to address the general problem, one should first consider propagation of prediction and filtering equations for the general case of nonlinear/non Gaussian models. These mechanism, that are based on the Markov property of the state equation, will be presented in chapter 3.

The propagation of prediction and filtering equations cannot be implemented directly since it involves complex integrations. In such situations, one usually resorts to importance sampling. Importance sampling is the basis to build approximate filter distributions which consist in empirical distributions built from sampled state sequences, a mechanism which is known as particle filtering and that will be presented in chapter 4.

Notation As usual in statistics literature, small letters will be used to design random variables or their realization, the meaning depending on the context.

Chapter 2

Kalman filter and non linear extensions

Many systems from nature or man made can be described through a state space model. Filtering is the operation that consists in estimating the inner state of the model once observation at the model output is available. More specifically, for linear models, the Kalman filter enables calculating the mean and covariance of the state conditional to observation from initial to present time. The Kalman procedure is optimal in the minimum mean square error sense when the entries of the system are Gaussian. First, we are going to recall this procedure here.

When the model is nonlinear the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) supply approximations of the Kalman filter through model linearization and exact transformation of a set of points respectively. We shall also present the equations for these approaches.

2.1 State space models and the Kalman filter [Kal60]

We shall denote the set $\{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ of random variables by $\mathbf{z}_{1:t}$. $\mathbf{x}_t|\mathbf{y}_{1:t'}$ will denote the linear MMSE (Minimum Mean Square Error) estimate of $\mathbf{x}_t$ conditional to $\mathbf{y}_{1:t'}$: $\mathbf{x}_t|\mathbf{y}_{1:t'} = \sum_{k=1}^{t'} \mathbf{A}_k \mathbf{y}_k$, where

$$(\mathbf{A}_1, \ldots, \mathbf{A}_{t'}) = \arg \min_{(\mathbf{B}_1,\ldots,\mathbf{B}_n)} \parallel \mathbf{x}_t - \sum_{k=1}^{t'} \mathbf{A}_k \mathbf{y}_k \parallel^2, \tag{2.1}$$

where $\parallel . \parallel$ denotes the L_2 norm. When there is no ambiguity, $\mathbf{x}_t|\mathbf{y}_{1:t'}$ will be denoted in short by $\mathbf{x}_{t|t'}$.

Let us consider a linear state space model of the form

$$\begin{cases} \mathbf{x}_t &= \mathbf{F}_t \mathbf{x}_t + \mathbf{G}_t \mathbf{v}_t \\ \mathbf{y}_t &= \mathbf{H}_t \mathbf{x}_t + \mathbf{U}_t \mathbf{n}_t. \end{cases} \tag{2.2}$$

In general, $\mathbf{G}_t$ and $\mathbf{U}_t$ can be incorporated in the noises $\mathbf{v}_t$ and $\mathbf{n}_t$ respectively. However, with a view to use common notations with linearized nonlinear models, we consider the model (2.2). We assume that $\mathbf{v}_t$ and $\mathbf{n}_t$ are white noises with zero mean and respective covariance matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$.

The Kalman filter is intended to calculate $\mathbf{x}_{t|t} = \mathbf{x}_t | \mathbf{y}_{1:t}$. The solution is given by solving problem (2.1) with $t' = t$. Clearly, letting $\mathbf{Y}_t = [\mathbf{y}_1^T, \ldots, \mathbf{y}_t^T]^T$, we have

$$\mathbf{x}_{t|t} = cov(\mathbf{x}_t, \mathbf{Y}_t) \times cov(\mathbf{Y}_t)^{-1} \times \mathbf{Y}_t. \tag{2.3}$$

However, the complexity of the calculation increases as $O(t^3)$ due to the calculation of the inverse of the covariance matrix in the above formula. Thus, the Kalman filter is a computationaly efficient implementation of the calculation of $\mathbf{x}_{t|t}$. It relies on the fact that the linear space spanned by $\mathbf{y}_1, \ldots, \mathbf{y}_t$, denoted by $Sp\{\mathbf{y}_{1:t}\}$, can be rewritten as the sum of two orthogonal spaces:

$$Sp\{y_{1:t}\} = Sp\{y_{1:t-1}\} \oplus Sp\{y_t - y_{t|t-1}\}, \tag{2.4}$$

with $\mathbf{y}_{t|t-1} = \mathbf{y}_t | \mathbf{y}_{1:t-1} = \mathbf{H}_t \mathbf{x}_{t|t-1}$ the predicted value for $\mathbf{y}_t$. Thus $\mathbf{x}_{t|t}$ can be decomposed as follows:

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{y}_{t|t-1}), \tag{2.5}$$

where

$$\mathbf{K}_t = cov(\mathbf{x}_t - \mathbf{x}_{t|t-1}, \mathbf{y}_t - \mathbf{y}_{t|t-1}) \times cov(\mathbf{y}_t - \mathbf{y}_{t|t-1})^{-1} \tag{2.6}$$

is named the Kalman gain.

Then the Kalman algorithm can be derived after a few more manipulations and is summarized as follows:

1. initialization: set $\mathbf{x}_0 = \mathbf{x}_{0|0}$ and $\mathbf{P}_0 = \mathbf{P}_{0|0}$
2. iterations: for $t \geq 1$,

$$\begin{array}{ll} \mathbf{x}_{t|t-1} & = \mathbf{F}_t \mathbf{x}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} & = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^T \\ \mathbf{K}_t & = \mathbf{P}_{t|t-1} \mathbf{H}_t^T [\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{U}_t \mathbf{R}_t \mathbf{U}_t^T]^{-1} \\ \mathbf{x}_{t|t} & = \mathbf{x}_{t|t-1} + \mathbf{K}_t [\mathbf{y}_t - \mathbf{H}_t \mathbf{x}_{t|t-1}] \\ \mathbf{P}_{t|t} & = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1} \\ & = \mathbf{P}_{t|t-1} - \mathbf{K}_t (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{U}_t \mathbf{R}_t \mathbf{U}_t^T) \mathbf{K}_t^T \end{array} \tag{2.7}$$

where $\mathbf{P}_{t|t-1} = cov(\mathbf{x}_t - \mathbf{x}_{t|t-1})$ and $\mathbf{P}_{t|t} = cov(\mathbf{x}_t - \mathbf{x}_{t|t})$. In addition,

$$cov(\mathbf{y}_t - \mathbf{y}_{t|t-1}) = \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{U}_t\mathbf{R}_t\mathbf{U}_t^T \tag{2.8}$$

is the covariance matrix of observation innovation. Let us simply recall here the calculation of $\mathbf{P}_{t|t}$, other equations being straightforward:

$$\begin{aligned}
\mathbf{P}_{t|t} &= cov(\mathbf{x}_t - \mathbf{x}_{t|t}) \\
&= cov(\mathbf{x}_t - \mathbf{x}_{t|t-1} - \mathbf{K}_t[\mathbf{y}_t - \mathbf{y}_{t|t-1}]) \\
&= \mathbf{P}_{t|t-1} + \mathbf{K}_t(\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{U}_t\mathbf{R}_t\mathbf{U}_t^T)\mathbf{K}_t^T \\
&\quad - cov(\mathbf{x}_t - \mathbf{x}_{t|t-1}, \mathbf{y}_t - \mathbf{y}_{t|t-1})\mathbf{K}_t^T - \mathbf{K}_t cov(\mathbf{y}_t - \mathbf{y}_{t|t-1}, \mathbf{x}_t - \mathbf{x}_{t|t-1}) \\
&= \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{H}_t\mathbf{P}_{t|t-1} \\
&= \mathbf{P}_{t|t-1} - \mathbf{K}_t(\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{U}_t\mathbf{R}_t\mathbf{U}_t^T)\mathbf{K}_t^T.
\end{aligned} \tag{2.9}$$

The last inequalities hold because, from Eq. (2.6) and Eq. (2.8) we have

$$\begin{aligned}
cov(\mathbf{x}_t - \mathbf{x}_{t|t-1}, \mathbf{y}_t - \mathbf{y}_{t|t-1}) &= cov(\mathbf{x}_t, \mathbf{y}_t - \mathbf{y}_{t|t-1}) \\
&= \mathbf{K}_t cov(\mathbf{y}_t - \mathbf{y}_{t|t-1}) \\
&= \mathbf{P}_{t|t-1}\mathbf{H}_t^T.
\end{aligned} \tag{2.10}$$

2.2 The Extended Kalman Filter (EKF) [AM79]

Let us now consider a nonlinear state space model of the form

$$\begin{cases} \mathbf{x}_{t+1} &= \mathbf{f}_{t+1}(\mathbf{x}_t, \mathbf{v}_{t+1}) \\ \mathbf{y}_t &= \mathbf{h}_t(\mathbf{x}_t, \mathbf{n}_t). \end{cases} \tag{2.11}$$

Such systems can be met even in the case of linear models when the matrices depend on some unknown parameter, as in the following model:

$$\begin{cases} \mathbf{x}_{t+1} &= \mathbf{F}_t(\theta)\mathbf{x}_t + \mathbf{v}_{t+1} \\ \mathbf{y}_t &= \mathbf{H}_t(\theta)\mathbf{x}_t + \mathbf{n}_t. \end{cases} \tag{2.12}$$

In this case, the state $\mathbf{x}$ can be augmented by inserting an additional equation to the state equation, of the form $\theta_{t+1} = \theta_t + \mathbf{v}_{\theta,t+1}$,where $\mathbf{v}_\theta$ is some noise that enables the parameter θ to be updated from its initialization value.

The Extended Kalman Filter (EKF) is based on a first order expansion of state equation and followed by application of the Kalman loop on linear equations. The state equation

$\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{v}_t)$ and of observation equation $\mathbf{y}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{n}_t)$ are linearized around points $(\mathbf{x}, \mathbf{v}) = (\mathbf{x}_{t-1|t-1}, 0)$ and $(\mathbf{x}, \mathbf{n}) = (\mathbf{x}_{t|t-1}, 0)$ respectively. The reason for these choice is that in the Kalman loop the state equation is used to express $\mathbf{x}_{t|t-1}$ as a function of $\mathbf{x}_{t-1|t-1}$, while the observation equation is used to express $\mathbf{x}_{t|t}$ as a function of $\mathbf{x}_{t|t-1}$, and because the noises are assumed to have zero means. Thus, the linearized model used for the EKF equations is of the form

$$\begin{cases} \mathbf{x}_{t+1} & = \mathbf{f}_{t+1}(\mathbf{x}_{t|t}, 0) + \mathbf{F}_{t+1}(\mathbf{x}_{t+1} - \mathbf{x}_{t|t}) + \mathbf{G}_{t+1}\mathbf{v}_t \\ \mathbf{y}_t & = \mathbf{h}_t(\mathbf{x}_{t|t-1}, 0) + \mathbf{H}_t(\mathbf{x}_t - \mathbf{x}_{t|t-1}) + \mathbf{U}_t\mathbf{n}_t, \end{cases} \tag{2.13}$$

where

$$\mathbf{F}_{t+1} = \frac{\partial \mathbf{f}_{t+1}}{\partial \mathbf{x}}(\mathbf{x}_{t|t}, 0),\ \mathbf{G}_{t+1} = \frac{\partial \mathbf{f}_{t+1}}{\partial \mathbf{v}}(\mathbf{x}_{t|t}, 0),\ \mathbf{H}_t = \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{x}}(\mathbf{x}_{t|t-1}, 0),\ \mathbf{U}_t = \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{n}}(\mathbf{x}_{t|t-1}, 0). \tag{2.14}$$

Then, the EKF equations are given by

$$\begin{aligned} \mathbf{x}_{t|t-1} &= \mathbf{f}_t(\mathbf{x}_{t-1|t-1}, 0) \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^T + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^T \\ \mathbf{K}_t &= \mathbf{P}_{t|t-1}\mathbf{H}_t^T[\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{U}_t\mathbf{R}_t\mathbf{U}_t^T]^{-1} \\ \mathbf{x}_{t|t} &= \mathbf{x}_{t|t-1} + \mathbf{K}_t[\mathbf{y}_t - \mathbf{h}_t(\mathbf{x}_{t|t-1}, 0)] \\ \mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{H}_t\mathbf{P}_{t|t-1} = \mathbf{P}_{t|t-1} - \mathbf{K}_t[\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{U}_t\mathbf{R}_t\mathbf{U}_t^T]\mathbf{K}_t^T. \end{aligned} \tag{2.15}$$

The problem with the EKF is that the linearization of the model yields biased estimate for $\mathbf{x}_{t|t}$ and erroneous covariance. The unscented Kalman filter is intended to solve this problem.

2.3 Unscented Kalman filter [JU96]

2.3.1 The unscented transform

Let $\mathbf{x} \sim \mathcal{N}(\overline{\mathbf{x}}, \mathbf{P}_\mathbf{x})$ be a random variable and $\mathbf{g}$ a function. The problem addressed by the unscented transform is to calculate unbiased estimates of the mean $\overline{\mathbf{y}}$ and covariance $\mathbf{P}_\mathbf{y}$ of the transformed variable $\mathbf{y} = \mathbf{g}(\mathbf{x})$. Instead of considering some linear approximation of $\mathbf{g}$ as in the EKF, Julier and Uhlmann have proposed to evaluate the first two moments $\overline{\mathbf{y}}$ and $\mathbf{P}_\mathbf{y}$ of $\mathbf{y}$ from weighted empirical estimates calculated from the exact transformation of a set of well chosen points around $\overline{\mathbf{x}}$. They have called these points σ-points. Letting

n denote the dimension of the random vector $\mathbf{x}$, the σ-points and there corresponding weights are in the form

$$\begin{array}{llll} \mathcal{X}_0 & = \overline{\mathbf{x}} & W_0 = \dfrac{\kappa}{\kappa+n} & \\ \mathcal{X}_i & = \overline{\mathbf{x}} + \left(\sqrt{(\kappa+n)P_{\mathbf{x}}}\right)_i & W_i = \dfrac{1}{2(\kappa+n)} & i = 1,\dots,n \\ \mathcal{X}_i & = \overline{\mathbf{x}} - \left(\sqrt{(\kappa+n)P_{\mathbf{x}}}\right)_{i-n} & W_i = \dfrac{1}{2(\kappa+n)} & i = n+1,\dots,2n. \end{array} \tag{2.16}$$

where κ is a scalar coefficient and $\sqrt{\mathbf{P}_{\mathbf{x}}}_i$ represents the i-th column of some factorization of $\mathbf{P}_{\mathbf{x}}$, such as the Cholesky factorization [GL96]. Thus,

$$\mathbf{P}_{\mathbf{x}} = \sqrt{\mathbf{P}_{\mathbf{x}}} \times \sqrt{\mathbf{P}_{\mathbf{x}}}^T = \sum_{i=1}^{n} (\sqrt{\mathbf{P}_{\mathbf{x}}})_i (\sqrt{\mathbf{P}_{\mathbf{x}}})_i^T. \tag{2.17}$$

Wer are going to see the reason for these choices [JU96]. Empirical statistics of $\mathbf{x}$ and of $\mathbf{y}$ can be calculated from the σ-points and the transformed σ-points respectively: letting

$$\mathcal{Y}_i = g(\mathcal{X}_i), \tag{2.18}$$

we define

$$\begin{array}{ll} \overline{\mathcal{X}} & = \sum_{i=1}^{2n+1} W_i \mathcal{X}_i \\ \mathbf{P}_{\mathcal{X}} & = \sum_{i=1}^{2n+1} W_i (\mathcal{X}_i - \overline{\mathcal{X}})(\mathcal{X}_i - \overline{\mathcal{X}})^T \\ \overline{\mathcal{Y}} & = \sum_{i=1}^{2n+1} W_i \mathcal{Y}_i \\ \mathbf{P}_{\mathcal{Y}} & = \sum_{i=1}^{2n+1} W_i (\mathcal{Y}_i - \overline{\mathcal{Y}})(\mathcal{Y}_i - \overline{\mathcal{Y}})^T. \end{array} \tag{2.19}$$

Then, it can be shown that these empirical means and covariances match the true mean and covariances of $\mathbf{x}$ exactly and those of $\mathbf{y}$ up to error terms of higher orders.

To check this, let us first consider $\mathcal{X} = (\mathcal{X}_i)_{i=1,\dots,2n+1}$. Clearly, $\overline{\mathbf{x}} = \overline{\mathcal{X}}$ and

$$\mathbf{P}_{\mathcal{X}} = \frac{1}{2(n+\kappa)} \times 2 \sum_{i=1}^{n} (n+\kappa)(\sqrt{\mathbf{P}_{\mathbf{x}}})_i (\sqrt{\mathbf{P}_{\mathbf{x}}})_i^T = \mathbf{P}_{\mathbf{x}}. \tag{2.20}$$

Now, assuming the Taylor development of the vector function $g = [g_1, \dots, g_m]^T$ $g : \mathbb{R}^n \to \mathbb{R}^m$ exists about $\overline{\mathbf{x}}$, we get

$$\mathbf{g}(\overline{\mathbf{x}} + \delta) = \mathbf{g}(\overline{x}) + D_\delta \mathbf{g}(\overline{x}) + D_\delta^2 \mathbf{g}(\overline{x}) + ... \tag{2.21}$$

where

$$\begin{array}{ll} D_\delta \mathbf{g}(\overline{x}) & = \nabla^T \mathbf{g}(\overline{\mathbf{x}})\delta \\ D_\delta^2 \mathbf{g}(\overline{x}) & = D_\delta[\nabla^T \mathbf{g}(\overline{\mathbf{x}})\delta] \\ & = \nabla^T[\nabla^T g(\overline{\mathbf{x}})\delta]\delta \\ & = [Tr(\nabla^2 \mathbf{g}_1(\overline{\mathbf{x}})\delta\delta^T), \dots, Tr(\nabla^2 \mathbf{g}_m(\overline{\mathbf{x}})\delta\delta^T)]^T, \end{array} \tag{2.22}$$

where ∇g is the transpose of the Jacobian matrix of $\mathbf{g}$ (the gradient of $\mathbf{g}$ in the case where $m = 1$) and $\nabla^2\mathbf{g}$ the Hessian matrix of $\mathbf{g}$. Thus, it comes that

$$\begin{aligned}\overline{\mathbf{y}} &= \mathbb{E}[\mathbf{g}(\mathbf{x})] \\ &= \mathbf{g}(\overline{\mathbf{x}}) + \mathbb{E}[\nabla^T g(\overline{x})\delta] + \frac{1}{2}\mathbb{E}[[Tr(\nabla^2\mathbf{g}_1(\overline{\mathbf{x}})\delta\delta^T), \ldots, Tr(\nabla^2\mathbf{g}_m(\overline{\mathbf{x}})\delta\delta^T)]^T] + ... \\ &= \mathbf{g}(\overline{\mathbf{x}}) + \frac{1}{2}[Tr(\nabla^2\mathbf{g}_1(\overline{\mathbf{x}})\mathbf{P}_\mathbf{x}), \ldots, Tr(\nabla^2\mathbf{g}_m(\overline{\mathbf{x}})\mathbf{P}_\mathbf{x})]^T + ...\end{aligned} \tag{2.23}$$

Note that since $\mathbf{x} = \overline{\mathbf{x}} + \delta$, $\delta \sim \mathcal{N}(0, \mathbf{P}_\mathbf{x})$ and odd terms in the right hand of Eq. (2.23) vanish. On another hand, letting $\sigma_i = (\sqrt{(\kappa + n)P_\mathbf{x}})_i$,

$$\begin{aligned}\overline{\mathcal{Y}} &= W_0\mathbf{g}(\overline{\mathbf{x}}) + \textstyle\sum_{i=1}^n W_i[\mathbf{g}(\overline{\mathbf{x}} + \sigma_i) + \mathbf{g}(\overline{\mathbf{x}} - \sigma_i)] \\ &= \mathbf{g}(\overline{\mathbf{x}}) + 0 + \frac{1}{2}\mathbb{E}[[Tr(\nabla^2\mathbf{g}_1(\overline{\mathbf{x}})\textstyle\sum_{i=1}^n W_i\sigma_i\sigma_i^T), \ldots, Tr(\nabla^2\mathbf{g}_m(\overline{\mathbf{x}})\textstyle\sum_{i=1}^n W_i\sigma_i\sigma_i^T)]^T] + ... \\ &= \mathbf{g}(\overline{\mathbf{x}}) + \frac{1}{2}[Tr(\nabla^2\mathbf{g}_1(\overline{\mathbf{x}})\mathbf{P}_\mathbf{x}), \ldots, Tr(\nabla^2\mathbf{g}_m(\overline{\mathbf{x}})\mathbf{P}_\mathbf{x})]^T + ...\end{aligned} \tag{2.24}$$

Here again, due to the symetry of σ-points about $\overline{\mathbf{x}}$, odd terms vanish. Thus, the weighted empirical mean of σ-points $\mathcal{Y}$ matches $\overline{\mathbf{y}}$ up to third orders terms.

As far as $\mathbf{P}_\mathbf{y}$ and $\mathbf{P}_\mathcal{Y}$ are concerned, we have

$$\begin{aligned}\mathbf{P}_\mathbf{y} &= \mathbb{E}[(\mathbf{g}(\overline{\mathbf{x}} + \delta) - \mathbf{g}(\overline{\mathbf{x}}))(\mathbf{g}(\overline{\mathbf{x}} + \delta) - \mathbf{g}(\overline{\mathbf{x}}))^T] \\ &= \mathbb{E}[(\nabla^T\mathbf{g}(\overline{\mathbf{x}})\delta)(\nabla^T\mathbf{g}(\overline{\mathbf{x}})\delta)^T] + ... \\ &= \nabla^T\mathbf{g}(\overline{\mathbf{x}})\mathbf{P}_\mathbf{x}\nabla\mathbf{g}(\overline{\mathbf{x}}).\end{aligned} \tag{2.25}$$

and

$$\begin{aligned}\mathbf{P}_\mathcal{Y} &= \textstyle\sum_{i=1}^{2n+1} W_i(\mathcal{Y}_i - \overline{\mathcal{Y}})(\mathcal{Y}_i - \overline{\mathcal{Y}})^T \\ &= \tfrac{1}{2(n+\kappa)}2\textstyle\sum_{i=1}^n W_i(\nabla^T\mathbf{g}(\overline{\mathbf{x}})\sigma_i + ...)(\nabla^T\mathbf{g}(\overline{\mathbf{x}})\sigma_i + ...)^T \\ &= \nabla^T\mathbf{g}(\overline{\mathbf{x}})\mathbf{P}_\mathbf{x}\nabla\mathbf{g}(\overline{\mathbf{x}}).\end{aligned} \tag{2.26}$$

Thus we see that $\mathbf{P}_\mathcal{Y}$ catches well the first order term of $\mathbf{P}_\mathbf{y}$. Thus, the unscented transform is unbiased.

By adjusting κ so that $n+\kappa = 3$, it can be shown [JU96] that the fourth order terms of the mean can be caught by $\mathcal{Y}$ too. However, in this case the weighted empirical covariance of $\mathbf{y}$ may become non-positive. This problem has been addressed by Julier who introduced scaled unscented transformations in [Jul02]. The idea is to scale the distance of the σ-points par replacing $\mathcal{X}_i$ by

$$\mathcal{X}_i' = \mathcal{X}_0 + \alpha(\mathcal{X}_i - \mathcal{X}_0) \tag{2.27}$$

where α is the scaling factor. In this case the weight of $\mathcal{X}_0$ for calculating the empirical mean and covariance must be different. Letting $\lambda = \alpha^2(\kappa + n) + n$ the scaled unscented transform can be summarized as follows:

$$\begin{array}{llll}
\mathcal{X}_0 & = \overline{\mathbf{x}} & W_0^{(m)} = \dfrac{\lambda}{\lambda + n}, & W_0^{(c)} = \dfrac{\lambda}{\lambda + n} + (1 - \alpha^2 + \beta) \\
\mathcal{X}_i & = \overline{\mathbf{x}} + \sqrt{(\lambda + n)P_{\mathbf{x}}}_i & W_i^{(m)} = W_i^{(c)} = \dfrac{1}{2(\kappa + n)}, & i = 1, \dots, n \\
\mathcal{X}_i & = \overline{\mathbf{x}} - \sqrt{(\kappa + n)P_{\mathbf{x}}}_{i-n} & W_i^{(m)} = W_i^{(c)} = \dfrac{1}{2(\kappa + n)}, & i = n + 1, \dots, 2n,
\end{array} \tag{2.28}$$

with $0 \leq \alpha \leq 1$, κ can be set to 0 and $\beta = 2$ is an optimal choice in the Gaussian case. Then, empirical means and covariances of $\mathbf{x}$ and $\mathbf{y}$ are given by

$$\begin{aligned}
\overline{\mathcal{X}} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(m)} \mathcal{X}_i \\
\mathbf{P}_{\mathcal{X}} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(c)} (\mathcal{X}_i - \overline{\mathcal{X}})(\mathcal{X}_i - \overline{\mathcal{X}})^T \\
\overline{\mathcal{Y}} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(m)} \mathcal{Y}_i \\
\mathbf{P}_{\mathcal{Y}} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(c)} (\mathcal{Y}_i - \overline{\mathcal{Y}})(\mathcal{Y}_i - \overline{\mathcal{Y}})^T.
\end{aligned} \tag{2.29}$$

2.3.2 The Unscented Kalman Filter (UKF)

Putting together formulas of the Kalman filter and the unscented transform algorithm yields the UKF that propagates approximate first and second order moments for the distribution of the mean and covariance of predicted and filtered state of the model. Considering again the nonlinear model (2.11), the Unscented transform in the UKF makes use of an augmented state defined by $\mathbf{x}_t^a = [\mathbf{x}_t^T \ \mathbf{v}_t^T \ \mathbf{n}_t^T]^T$ together with its extended covariance matrix. Then, the UKF algorithm is summarized on next page

1. initialization: set $\mathbf{x}_{0|0}$ and $\mathbf{P}_{0|0}$
2. iterations: for $t \geq 1$,

$$
\begin{aligned}
\mathbf{P}^a &= \begin{bmatrix} \mathbf{P}_{t-1|t-1} & 0 & 0 \\ 0 & \mathbf{Q}_t & 0 \\ 0 & 0 & \mathbf{R}_t \end{bmatrix}, \\
\mathcal{X}^a &= [\mathbf{x}_{t-1|t-1} \; \mathbf{x}_{t-1|t-1} \pm \sqrt{(n_a + \lambda)\mathbf{P}^a}] = \begin{bmatrix} \mathcal{X}^x_{t-1|t-1} \\ \mathcal{X}^v_t \\ \mathcal{X}^n_t \end{bmatrix} \\
\mathcal{X}_{t|t-1} &= \mathbf{f}_t(\mathcal{X}^x_{t|t-1}, \mathcal{X}^v_t) \\
\mathbf{x}_{t|t-1} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(m)} \mathcal{X}_{t|t-1,i} \\
\mathbf{P}_{t|t-1} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(c)} (\mathcal{X}_{t|t-1,i} - \mathbf{x}_{t|t-1})(\mathcal{X}_{t|t-1,i} - \mathbf{x}_{t|t-1})^T \\
\mathcal{Y}_{t|t-1} &= \mathbf{h}_t(\mathcal{X}_{t|t-1}, \mathcal{X}^v_t) \\
\mathbf{y}_{t|t-1} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(m)} \mathcal{Y}^x_{t|t-1,i} \\
\mathbf{P}^{\mathbf{y}}_{t|t-1} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(c)} (\mathcal{Y}_{t|t-1,i} - \mathbf{y}_{t|t-1})(\mathcal{Y}_{t|t-1,i} - \mathbf{y}_{t|t-1})^T \\
\mathbf{P}^{\mathbf{xy}}_{t|t-1} &= \textstyle\sum_{i=1}^{2n+1} W_i^{(c)} (\mathcal{X}^x_{t|t-1,i} - \mathbf{x}_{t|t-1})(\mathcal{Y}_{t|t-1,i} - \mathbf{y}_{t|t-1})^T \\
\mathbf{K}_t &= \mathbf{P}^{\mathbf{xy}}_{t|t-1}[\mathbf{P}^{\mathbf{y}}_{t|t-1}]^{-1} \\
\mathbf{x}_{t|t} &= \mathbf{x}_{t|t-1} + \mathbf{K}_t[\mathbf{y}_t - \mathbf{y}_{t|t-1}] \\
\mathbf{P}_{t|t} &= \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{P}^{\mathbf{y}}_{t|t-1} \mathbf{K}_t^T.
\end{aligned}
\tag{2.30}
$$

Chapter 3

Propagation of distributions in state space models

In the case of a nonlinear and/or non Gaussian state space model, it is possible to propagate prediction, filtering or smoothing distributions in an exact form [PS], generalizing thus the Kalman equations that perform these tasks for mean and covariance in the linear case.

Let us consider again the state space model defined by

$$\begin{cases} \mathbf{x}_{t+1} & = \mathbf{f}_{t+1}(\mathbf{x}_t, \mathbf{v}_{t+1}) \\ \\ \mathbf{y}_t & = \mathbf{h}_t(\mathbf{x}_t, \mathbf{n}_t). \end{cases} \tag{3.1}$$

We assume that $(\mathbf{v}_{t+1}, \mathbf{n}_t)_{t \geq 0}$ is a white noise. We also assume that the distributions $p(\mathbf{x}_0)$ and $p(\mathbf{x}_{t+1}, \mathbf{y}_t | \mathbf{x}_t)$ are known. When $\mathbf{v}_{t+1}$ and $\mathbf{n}_t$ are independent, then $p(\mathbf{x}_{t+1}, \mathbf{y}_t | x_t) = p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{y}_t|\mathbf{x}_t)$. We will consider that processes $\mathbf{v}$ and $\mathbf{n}$ are independent in what follows, although this is not necessary to propagate distributions.

The distributions that we would like to characterize are of the form $p(\mathbf{x}_k|\mathbf{y}_{1:t})$. We have th following definitions

- if $k > t$, $p(\mathbf{x}_k|\mathbf{y}_{1:t})$ is the $t-k$ steps predictor of $\mathbf{x}$,
- if $k = t$, $p(\mathbf{x}_k|\mathbf{y}_{1:t})$ is the filter of $\mathbf{x}$,
- if $k < t$, $p(\mathbf{x}_k|\mathbf{y}_{1:t})$ is the smoother of $\mathbf{x}$.

These distributions are also called normalized predictor, filter or smoother. The corresponding un-normalized distributions are defined respectively by $p(\mathbf{x}_k,\ \mathbf{y}_{1:t})$, $p(\mathbf{x}_k,\ \mathbf{y}_{1:t})$

and $p(\mathbf{x}_k,\ \mathbf{y}_{1:t})$. We can go from the un-normalized distributions to the normalized ones by dividing them by $p(\mathbf{y}_{1:t})$. The coefficient $p(\mathbf{y}_{1:t})$ can be expressed directly in terms of the known distributions $p(\mathbf{x}_0)$ and $p(\mathbf{x}_{t+1}, \mathbf{y}_t|\mathbf{x}_t)$:

$$\begin{aligned} p(\mathbf{y}_{1:t}) &= \int_{x_{0:t}} p(\mathbf{y}_{1:t}, x_{0:t}) dx_{0:t} \\ &= \int_{x_{0:t}} p(\mathbf{y}_t|\mathbf{x}_t) \prod_{i=1:t} p(\mathbf{y}_{i-1}, \mathbf{x}_i|\mathbf{x}_{i-1}) p(\mathbf{x}_0) dx_{0:t}. \end{aligned} \tag{3.2}$$

The capability of propagating these distributions relies on the markovian nature of the process $(\mathbf{x}_{t+1}, \mathbf{y}_t)$:

$$p(\mathbf{x}_{t+1}, \mathbf{y}_t|(\mathbf{x}_{i+1}, \mathbf{y}_i)_{i=1:t-1}) = p(\mathbf{x}_{t+1}, \mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{t-1}) = p(\mathbf{x}_{t+1}, \mathbf{y}_t|\mathbf{x}_t). \tag{3.3}$$

The one step predictor can be written as

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \int_{\mathbf{x}_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t}) d\mathbf{x}_t. \tag{3.4}$$

Thus we see that the one step predictor $p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})$ can be obtained from the filter $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, extending thus the prediction equations of the Kalman algorithm where predicted mean and covariance are calculated from filtered mean and covariance calculated in the previous iteration. In the same way, the filter can be updated from the predictor:

$$\begin{aligned} p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) &= \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})} \\ &= \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})}{\displaystyle\int_{\mathbf{x}_{t+1}} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) d\mathbf{x}_{t+1}}. \end{aligned} \tag{3.5}$$

By propagating these equations it is possible to calculate the one step predictor and the filter at any time. Note however that these nice formulas are of little help if we are not able to calculate involved integrals. In the chapter devoted to particle filtering we shall see that how approximate empirical distributions can be propagated by using importance sampling.. Note that combining the prediction and filtering steps it is possible to update the filter directly from time t to time $t+1$: indeed, putting Eq. (3.4) and Eq. (3.5) together, we get

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) = \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) \int_{\mathbf{x}_t} p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t}) d\mathbf{x}_t}{\displaystyle\int_{\mathbf{x}_t, \mathbf{x}_{t+1}} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t}) d\mathbf{x}_t d\mathbf{x}_{t+1}}. \tag{3.6}$$

Let us give a few more examples of equations propagation. The k-steps predictor can be obtained by marginalizing $p(\mathbf{x}_{t:t+k}|\mathbf{y}_{1:t})$:

$$\begin{aligned} p(\mathbf{x}_{t+k}|\mathbf{y}_{1:t+1}) &= \int_{\mathbf{x}_{t+1:t+k-1}} p(\mathbf{x}_{t+1:t+k}|\mathbf{y}_{1:t}) d\mathbf{x}_{t+1:t+k-1} \\ &= \int_{\mathbf{x}_{t+1:t+k-1}} \prod_{i=2}^{k} p(\mathbf{x}_{t+i}|\mathbf{x}_{t+i-1}) p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) d\mathbf{x}_{t+1:t+k-1}. \end{aligned} \tag{3.7}$$

Smoothing distributions can also be propagated. For fixed t, let us consider the unnormalized smoother $p(\mathbf{x}_k, \mathbf{y}_{1:t})$, with $k < t$. This distribution can be calculated iteratively as follows:

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{y}_{1:t}) &= \int_{\mathbf{x}_{0:k-1}, \mathbf{x}_{k+1:t}} p(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) d\mathbf{x}_{0:k-1} d\mathbf{x}_{k+1:t} \\ &= \int_{\mathbf{x}_{0:k-1}, \mathbf{x}_{k+1:t}} p(\mathbf{y}_{k:t}, \mathbf{x}_{k+1:t}, |\mathbf{x}_{k:t}, \mathbf{y}_{1:k-1}) p(\mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}) d\mathbf{x}_{0:k-1} d\mathbf{x}_{k+1:t} \\ &= \int_{\mathbf{x}_{0:k-1}} p(\mathbf{x}_0) p(\mathbf{x}_1|\mathbf{x}_0) \prod_{i=1}^{k-1} p(\mathbf{y}_i, \mathbf{x}_{i+1}|\mathbf{x}_i) d\mathbf{x}_{0:k-1} \\ &\quad \times \int_{\mathbf{x}_{k+1:t}} p(\mathbf{y}_t|\mathbf{x}_t) \prod_{i=k}^{t-1} p(\mathbf{y}_i, \mathbf{x}_{i+1}|\mathbf{x}_i) d\mathbf{x}_{k+1:t} \\ &= \alpha_k(\mathbf{x}_k) \times \beta_k(\mathbf{x}_k) \end{aligned} \tag{3.8}$$

It is easy to check that the two integral factors $\alpha_k(\mathbf{x}_k)$ and $\beta_k(\mathbf{x}_k)$ can be rewritten as $\alpha_k(\mathbf{x}_k) = p(\mathbf{x}_k, \mathbf{y}_{1,k-1})$ and $\beta_k(\mathbf{x}_k) = p(\mathbf{y}_{k:t}|\mathbf{x}_k)$. $\alpha_k(\mathbf{x}_k)$ and $\beta_k(\mathbf{x}_k)$ are called forward and backward filter respectively. In addition, one can see that these filter satisfy the following recurrence equations

$$\begin{aligned} \alpha_{k+1}(\mathbf{x}_{k+1}) &= \int_{\mathbf{x}_k} p(\mathbf{y}_k, \mathbf{x}_{k+1}|\mathbf{x}_k) \alpha_k(\mathbf{x}_k) d\mathbf{x}_k \\ \beta_k(\mathbf{x}_k) &= \int_{\mathbf{x}_{k+1}} p(\mathbf{y}_k, \mathbf{x}_{k+1}|\mathbf{x}_k) \beta_{k+1}(\mathbf{x}_{k+1}) d\mathbf{x}_{k+1}. \end{aligned} \tag{3.9}$$

Thus, we see that it is possible to express recursively the forward and backward filters, going forward and backward respectively and with respective initialization $\alpha_1(\mathbf{x}_1) = p(\mathbf{x}_1) = \int_{\mathbf{x}_0} p(\mathbf{x}_0) p(\mathbf{x}_1|\mathbf{x}_0) d\mathbf{x}_0$ and $\beta_t(\mathbf{x}_t) = p(\mathbf{y}_t|\mathbf{x}_t)$.

Alternatively, the normalized smoothing at time u can be expressed in terms of the smoothing at time $u+1$ and of the filter and one step predictor at time u:

$$\begin{aligned}
p(\mathbf{x}_u|\mathbf{y}_{1:t}) &= \int_{\mathbf{x}_{u+1}} p(\mathbf{x}_u, \mathbf{x}_{u+1}|\mathbf{y}_{1:t}) d\mathbf{x}_{u+1} \\
&= \int_{\mathbf{x}_{u+1}} p(\mathbf{x}_u|\mathbf{x}_{u+1}\mathbf{y}_{1:t}) p(\mathbf{x}_{u+1}|\mathbf{y}_{1:t}) d\mathbf{x}_{u+1} \\
&= \int_{\mathbf{x}_{u+1}} p(\mathbf{x}_u|\mathbf{x}_{u+1}\mathbf{y}_{1:u}) p(\mathbf{x}_{u+1}|\mathbf{y}_{1:t}) d\mathbf{x}_{u+1} \qquad (3.10) \\
&= \int_{\mathbf{x}_{u+1}} \frac{p(\mathbf{x}_{u+1}|\mathbf{x}_u\mathbf{y}_{1:u}) p(\mathbf{x}_u|\mathbf{y}_{1:u})}{p(\mathbf{x}_{u+1}|\mathbf{y}_{1:u})} p(\mathbf{x}_{u+1}|\mathbf{y}_{1:t}) d\mathbf{x}_{u+1} \\
&= \int_{\mathbf{x}_{u+1}} \frac{p(\mathbf{x}_{u+1}|\mathbf{x}_u) p(\mathbf{x}_u|\mathbf{y}_{1:u})}{p(\mathbf{x}_{u+1}|\mathbf{y}_{1:u})} p(\mathbf{x}_{u+1}|\mathbf{y}_{1:t}) d\mathbf{x}_{u+1}.
\end{aligned}$$

As an example of this forward backward procedure, let us consider the smoothing for a linear Gaussian model. In this case, the forward filter is supplied by the prediction filter of Kalman equations. To propagate the backward equations, we propagate the distribution $p(\mathbf{x}_u|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{x}_u; \mathbf{m}_u, \mathbf{R}_u)$ from $p(\mathbf{x}_{u+1}|\mathbf{y}_{1:t})$. Note that for $u = t$, $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is supplied by the Kalman filter: $p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t|t}, \mathbf{P}_{t|t})$, providing thus initialization for the backward calculation of $p(\mathbf{x}_u|\mathbf{y}_{1:t})$.

Now, in order to use formula (3.10), let us note that up to constant terms

$$\begin{aligned}
&-2\log(p(\mathbf{x}_{u+1}|\mathbf{x}_u)p(\mathbf{x}_u|\mathbf{y}_{1:u})) \\
&= (\mathbf{x}_{u+1} - \mathbf{F}_{u+1}\mathbf{x}_u)^T \mathbf{Q}_v^{-1} (\mathbf{x}_{u+1} - \mathbf{F}_{u+1}\mathbf{x}_u) + (\mathbf{x}_u - \mathbf{x}_{u|u})^T \mathbf{P}^{[}_{u|u} - 1 (\mathbf{x}_u - \mathbf{x}_{u|u}) \qquad (3.11) \\
&\propto \mathbf{x}_u^T (\mathbf{F}_{u+1}^T \mathbf{Q}_v^{-1} \mathbf{F}_{u+1} + \mathbf{P}_{u|u}^{-1}) \mathbf{x}_u - 2\mathbf{x}_u^T (\mathbf{F}_{u+1}^T \mathbf{Q}_v^{-1} \mathbf{x}_{u+1} + \mathbf{P}_{u|u}^{-1} \mathbf{x}_{u|u})
\end{aligned}$$

Identifying the covariance and the mean of $p(\mathbf{x}_u|\mathbf{x}_{u+1}\mathbf{y}_{1:t})$ leads directly to

$$\begin{aligned}
&p(\mathbf{x}_u|\mathbf{x}_{u+1}\mathbf{y}_{1:t}) = \\
&\mathcal{N}(\mathbf{x}_u; [\mathbf{F}_{u+1}^T \mathbf{Q}_v^{-1} \mathbf{F}_{u+1} + \mathbf{P}_{u|u}^{-1}]^{-1}]^{-1} [\mathbf{F}_{u+1}^T \mathbf{Q}_v^{-1} \mathbf{x}_{u+1} + \mathbf{P}_{u|u}^{-1} \mathbf{x}_{u|u}], [\mathbf{F}_{u+1}^T \mathbf{Q}_v^{-1} \mathbf{F}_{u+1} + \mathbf{P}_{u|u}^{-1}]^{-1}).
\end{aligned} \qquad (3.12)$$

Integrating this density w.r.t. $p(\mathbf{x}_{u+1}|\mathbf{y}_{1:t})$ yields

$$\begin{aligned}
&p(\mathbf{x}_u|\mathbf{y}_{1:t}) = \\
&\mathcal{N}(\mathbf{x}_u; [\mathbf{F}_{u+1}^T \mathbf{Q}_v^{-1} \mathbf{F}_{u+1} + \mathbf{P}_{u|u}^{-1}]^{-1} [\mathbf{F}_{u+1}^T \mathbf{Q}_v^{-1} \mathbf{m}_{u+1} + \mathbf{P}_{u|u}^{-1} \mathbf{x}_{u|u}], [\mathbf{F}_{u+1}^T \mathbf{Q}_v^{-1} \mathbf{F}_{u+1} + \mathbf{P}_{u|u}^{-1}]^{-1}).
\end{aligned} \qquad (3.13)$$

In order to get nicer formulas, let us introduce a backward Kalman gain, defined by

$$\mathbf{K}_u^b = \mathbf{P}_{u|u} \mathbf{F}_{u+1}^T [\mathbf{Q}_v + \mathbf{F}_{u+1} \mathbf{P}_{u|u} \mathbf{F}_{u+1}^T]^{-1} \qquad (3.14)$$

Then, using the matrix inversion lemma (see Appendix), we get

$$\begin{aligned}
\mathbf{R}_u &= [\mathbf{F}_{u+1}\mathbf{Q}_v^{-1}\mathbf{F}_{u+1}^T + \mathbf{P}_{u|u}^{-1}]^{-1} \\
&= \mathbf{P}_{u|u} - \mathbf{P}_{u|u}\mathbf{F}_{u+1}^T[\mathbf{Q}_v + \mathbf{F}_{u+1}\mathbf{P}_{u|u}\mathbf{F}_{u+1}^T]^{-1}\mathbf{F}_{u+1}\mathbf{P}_{u|u} \qquad (3.15) \\
&= \mathbf{P}_{u|u} - \mathbf{K}_u^b\mathbf{F}_{u+1}\mathbf{P}_{u|u}
\end{aligned}$$

and $\mathbf{m}_u$ can be expressed as

$$\begin{aligned}
\mathbf{m}_u &= (\mathbf{P}_{u|u} - \mathbf{K}_u^b\mathbf{F}_{u+1}\mathbf{P}_{u|u})[\mathbf{F}_{u+1}^T\mathbf{Q}_v^{-1}\mathbf{m}_{u+1} + \mathbf{P}_{u|u}^{-1}\mathbf{x}_{u|u}] \\
&= \mathbf{x}_{u|u} - \mathbf{K}_u^b\mathbf{F}_{u+1}\mathbf{x}_{u|u} + \mathbf{P}_{u|u}\mathbf{F}_{u+1}^T\mathbf{Q}_v^{-1}\mathbf{m}_{u+1} - \mathbf{K}_u^b F_{u+1}\mathbf{P}_{u|u}\mathbf{F}_{u+1}^T\mathbf{Q}_v^{-1}\mathbf{m}_{u+1} \\
&= \mathbf{x}_{u|u} - \mathbf{K}_u^b\mathbf{F}_{u+1}\mathbf{x}_{u|u} + \mathbf{K}_u^b[\mathbf{Q}_v + \mathbf{F}_{u+1}\mathbf{P}_{u|u}\mathbf{F}_{u+1}^T]\mathbf{Q}_v^{-1}\mathbf{m}_{u+1} - \mathbf{K}_u^b F_{u+1}\mathbf{P}_{u|u}\mathbf{F}_{u+1}^T\mathbf{Q}_v^{-1}\mathbf{m}_{u+1} \\
&= \mathbf{x}_{u|u} + \mathbf{K}_u^b[\mathbf{m}_{u+1} - \mathbf{F}_{u+1}\mathbf{x}_{u|u}]
\end{aligned} \qquad (3.16)$$

Thus calculating iteratively $(\mathbf{K}_u^b, \mathbf{R}_u, \mathbf{m}_u)$ from $u = t$ downto $u = k$ leads to $p(\mathbf{x}_k|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_k, \mathbf{R}_k)$.

Chapter 4

Particle filters

We have seen in chapter 2 that the EKF and UKF supply approximate Gauss approximations of the state conditional to the observations. Often, the EKF works quite poorly due to first order Taylor approximations of state and observation equations. On another hand, the UKF propagates a Gaussian approximation of the state and observation transformation equations. But, in general the transformed distributions are not Gaussian, even if the inputs of the state and observation equations are Gaussian. To get rid of the Gaussian approximation, researches have been made to propagate directly the filter predictor and filter distributions. As we have seen in the previous chapter these propagation equations involve integrations that are often complex. This suggests using importance sampling techniques to approximate these equations. We are going to present how this can be done. A convenient choice of the importance distribution enables propagation of approximate predictor and filter distributions. However, this propagation suffers degeneracy phenomenon. The resampling technique has been introduced in this context in 1993 [GSS93], making thus possible efficient implementation of particle filter equations.

4.1 Importance sampling for filter equation

In the previous chapter, we have recalled recurrence equations in state space models among the distributions of the state variable conditional to observations. In a general form, we can write that we are interested in calculating $p(\mathbf{x}_{0:t}|y_{1:t})$. Then it is possible, from Monte carlo samples $\mathbf{x}_{0:t}^{(i)}$ $(i = 1, \ldots, N)$ drawn from $p(\mathbf{x}_{0:t}|y_{1:t})$, to estimate quantities such as $\mathbb{E}[k(\mathbf{x}_{0:t})]$ by approximations of the form

$$\frac{1}{N}\sum_{i=1}^{N} k(\mathbf{x}_{0:t}^{(i)}). \tag{4.1}$$

For independent samples $\mathbf{x}_{0:t}^{(i)}$ the law of large number and central limit theorems tell us about the convergence of this approximation towards $\mathbb{E}[k(\mathbf{x}_{0:t})]$.

But unfortunately, as we have seen it in the previous chapter, $p(\mathbf{x}_{0:t}|y_{1:t})$ is not known in closed form in general and sampling from it is not straightforward. Thus the problem will be solved by resorting to importance sampling.

Let us recall here that importance sampling is based on the simple remark that

$$\begin{aligned}\mathbb{E}_f[k(X)] &= \int k(x)f(x)dx \\ &= \int k(x)\frac{f(x)}{g(x)}g(x)dx \\ &= \mathbb{E}_g[\frac{f(X)}{g(X)}\ k(X)]\end{aligned} \tag{4.2}$$

for any pdf g which has a support containing that of f. In particular, if $(x^{(i)})_{i=1,\ldots,N}$ are independent samples drawn from g, then

$$\mathbb{E}_f[k(X)] \approx \sum_{i=1}^{N} \frac{f(x^{(i)})}{g(x^{(i)})}\ k(x^{(i)}). \tag{4.3}$$

In other words, distribution with pdf f is approximated by the weighted empirical distribution

$$\hat{P}(dx) = \sum_{i=1}^{N} \frac{f(x^{(i)})}{g(x^{(i)})}\ \delta_{x^{(i)}}(dx), \tag{4.4}$$

where the weights $f(x^{(i)})/g(x^{(i)})$ "compensate" the fact that samples are drawn from the pdf g instead of f.

The particle filter is based on this idea of an auxiliary importance distribution. In the particle filter there are however two specificities that must also be accounted for: (i) the necessity to propagate filter equations at each instant and (ii) a degeneracy phenomenon that makes most of the weights of the estimate of $p(\mathbf{x}_{0:t}|y_{1:t})$ go to zero. Equations propagation has been studied for long while the second problem has only been solved in a satisfactory way in the early nineties [GSS93].

4.2 Sequential importance sampling

If $\mathbf{x}_{0:t} \sim p(\mathbf{x}_{0:t}|y_{1:t})$, then letting $q(\mathbf{x}_{0:t}|y_{1:t})$ denote an importance distribution and defining unnormalized importance weights by

$$w(\mathbf{x}_{0:t}) = \frac{p(y_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{q(\mathbf{x}_{0:t}|y_{1:t})}, \tag{4.5}$$

we have

$$
\begin{aligned}
\mathbb{E}[k(\mathbf{x}_{0:t})] &= \int k(\mathbf{x}_{0:t})\frac{p(\mathbf{x}_{0:t}|y_{1:t})}{q(\mathbf{x}_{0:t}|y_{1:t})}q(\mathbf{x}_{0:t}|y_{1:t})d\mathbf{x}_{0:t} \\
&= \int k(\mathbf{x}_{0:t})\frac{p(y_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(y_{1:t})q(\mathbf{x}_{0:t}|y_{1:t})}q(\mathbf{x}_{0:t}|y_{1:t})d\mathbf{x}_{0:t} \\
&= \frac{1}{p(y_{1:t})}\int k(\mathbf{x}_{0:t})w(\mathbf{x}_{0:t})q(\mathbf{x}_{0:t}|y_{1:t})d\mathbf{x}_{0:t} \\
&= \frac{\displaystyle\int k(\mathbf{x}_{0:t})w(\mathbf{x}_{0:t})q(\mathbf{x}_{0:t}|y_{1:t})d\mathbf{x}_{0:t}}{\displaystyle\int w(\mathbf{x}_{0:t})q(\mathbf{x}_{0:t}|y_{1:t})d\mathbf{x}_{0:t}}.
\end{aligned} \tag{4.6}
$$

Then, if $\mathbf{x}_{0:t}^{(i)} \sim q(\mathbf{x}_{0:t}|y_{1:t})$ for $i = 1, \ldots, N$, then

$$
\begin{aligned}
\mathbb{E}[k(\mathbf{x}_{0:t})] &\approx \sum_{i=1}^{N} \frac{w(\mathbf{x}_{0:t}^{(i)})}{\sum_{j=1}^{N} w(\mathbf{x}_{0:t}^{(j)})} k(\mathbf{x}_{0:t}^{(j)}) \\
&= \textstyle\sum_{i=1}^{N} \tilde{w}(\mathbf{x}_{0:t}^{(i)})k(\mathbf{x}_{0:t}^{(i)}),
\end{aligned} \tag{4.7}
$$

where the normalized weights $\tilde{w}(\mathbf{x}_{0:t}^{(i)})$ sum to one: $\sum_{i=1}^{N} \tilde{w}(\mathbf{x}_{0:t}^{(i)}) = 1$. Clearly, here $p(\mathbf{x}_{0:t}|y_{1:t})$ is approximated by

$$
\hat{P}(d\mathbf{x}_{0:t}|y_{1:t}) = \sum_{i=1}^{N} \tilde{w}(\mathbf{x}_{0:t}^{(i)})\delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t}). \tag{4.8}
$$

Note that although the above approximation involves a ratio of sums, it is possible to derive convergence properties and in particular a central limit theorem, provided that the $\mathbf{x}_{0:t}^{(i)} \sim q(\mathbf{x}_{0:t}|y_{1:t})$ $(i = 1, \ldots, N)$ are independent samples, $supp(p(\mathbf{x}_{0:t}|y_{1:t})) \subset supp(q(\mathbf{x}_{0:t}|y_{1:t}))$, and functions $k(\mathbf{x}_{0:t})$, $w(\mathbf{x}_{0:t})$ and $w(\mathbf{x}_{0:t})k^2(\mathbf{x}_{0:t})$ have finite means under distribution $p(\mathbf{x}_{0:t}|y_{1:t})$ [Gew89].

With a view to propagate the approximate distribution $\hat{P}(d\mathbf{x}_{0:t}|y_{1:t})$, we shall put some asumption upon q. From Bayes formula, we have $q(\mathbf{x}_{0:t}|y_{1:t}) = q(\mathbf{x}_t|\mathbf{x}_{0:t-1}y_{1:t})q(\mathbf{x}_{0:t-1}|y_{1:t})$. In fact, we shall choose q such that $q(\mathbf{x}_{0:t-1}|y_{1:t}) = q(\mathbf{x}_{0:t-1}|y_{1:t-1})$. Thus,

$$
q(\mathbf{x}_{0:t}|y_{1:t}) = q(\mathbf{x}_t|\mathbf{x}_{0:t-1}y_{1:t})q(\mathbf{x}_{0:t-1}|y_{1:t-1}). \tag{4.9}
$$

Then, noting that

$$
\begin{aligned}
p(\mathbf{x}_{0:t}) &= p(\mathbf{x}_0)\textstyle\prod_{j=1}^{t} p(\mathbf{x}_j|\mathbf{x}_{j-1}) \\
p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) &= \textstyle\prod_{j=1}^{t} p(\mathbf{y}_j|\mathbf{x}_j),
\end{aligned} \tag{4.10}
$$

we get

$$\begin{aligned} w(\mathbf{x}_{0:t}) &= \frac{p(\mathbf{x}_{0:t})p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}y_{1:t})q(\mathbf{x}_{0:t-1}|y_{1:t-1})} \\ &= w(\mathbf{x}_{0:t-1})\frac{p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{y}_t|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}y_{1:t})}. \end{aligned} \tag{4.11}$$

Note that high variance of the weights samples $w(\mathbf{x}_{0:t}^{(i)})$ would yield normalized weights being close to zero, preventing thus from benefiting from the asymptotic convergence approximation brought by averaging, unless using a very high value for the number N of samples.

It has been proven in [DdFG01] that choosing $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, y_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, y_{1:t})$ would be the optimum choice to minimize the variance of $w(\mathbf{x}_{0:t})$. However, simulating from $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, y_{1:t})$ is infeasible in general, which is precisely the reason why we are resorting to importance sampling. Very often people consider $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, y_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ which is generally the easiest way to sample $\mathbf{x}_t$. Then we get the following strategy for sampling and weights updating:

- $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}) \to \mathbf{x}_{0:t}^i = (\mathbf{x}_{0:t-1}^i, \mathbf{x}_t^{(i)})$, $i = 1, \ldots, N$
- $w(\mathbf{x}_{0:t}^{(i)}) = w(\mathbf{x}_{0:t}^i)p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$.

A sequence $\mathbf{x}_{0:t}^{(i)}$ is called a particle. For this choice of q, the particle filter is also called **bootstrap filter**.

Note that the above simulation/weight-update iteration is closely related to the prediction/filtering met in the Kalman filter. Simply, for each particle, prediction is replaced by simulation from the prediction distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$, while the update of state prediction from data innovation in Kalman filter is replaced by weights update by multiplication by the likelihood $p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$.

4.3 Resampling

Resampling is intended to avoid degeneracy. The degeneracy can be measured by

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} \tilde{w}_i^2}. \tag{4.12}$$

Clearly, $N_{eff} \in [1, N]$ with $N_{eff} = N$ when $\tilde{w}_t^{(i)} = 1/N$ for all i and $N_{eff} = 1$ when all the weights but one are equal to 0. In order not to resample at each iteration of the particle filter, we can fix some threshold N_{thr} (for instance $0.9N$ or $0.7N$) and resample only when $N_{eff} < N_{thr}$.

Re-sampling can be implemented by randomly selecting N times one particle among the initial particles $\hat{\mathbf{x}}_{0:t}^{(i)}$ with probabilities equal to their weights $\tilde{w}_t^{(i)}$ and then to set to $1/N$ all weights of the new particles. Note that with this procedure initial particles with high weights will be multiplicated while those will small weights tend to disappear.

A simpler and more efficient technique[Kit96, CML99], in terms of variance of the re-sampled weights, consists first in drawing randomly a point m_0 in $[0, 1/N]$, uniformly. Then we consider the points $m_i = m_0 + i/N$ for $i = 0, \dots, N-1$ and resample the k-th particle N_k times, where N_k is the number of points m_i that fall inside k-th interval of the cumulative function of $(\tilde{w}_t^{(i)})_{i=1,\dots,N}$, that is inside $[\sum_{i=1}^{k-1} \tilde{w}_t^{(i)}, \sum_{i=1}^{k} \tilde{w}_t^{(i)}]$. Clearly $N_k \approx \tilde{w}_t^{(k)} N$.

The Matlab code for this resampling procedure is given down here, where `Wn` is the vector of normalized weights, `part` the matrix of particles and `N_thr` the threshold for resampling:

```
N_eff        = 1/sum(Wn.^2);
if N_eff<N_thr
     Fw           = [0; cumsum(Wn)];
     pts          = rand/nb + (0:(nb-1))'/nb;   % regularly spaced points
     aux          = 1;                          % index parameter
     part_aux     = zeros(nb,1);
     for m=1:N_part,
       nb_pts                      = sum((pts>Fw(m)).*(pts<Fw(m+1)));
       part_aux(aux:aux+nb_pts-1) = part(m,t)*ones(nb_pts,1);
       aux                         = aux + nb_pts;
     end;
     part(:,t)    = part_aux;                  % resampled particles
     Wn           = ones(nb,1)/nb;              % updated weights
end;
```

To illustrate the interest of particle filters compared to EKF and UKF, let us consider the following non linear state space model:

$$\begin{cases} \mathbf{x}_{t+1} &= 0.5\mathbf{x}_t + 25\mathbf{x}_t(1+\mathbf{x}_t^2)^{-1} + 8\cos(1.2t) + \mathbf{v}_t, \\ \mathbf{y}_t &= 0.05\mathbf{x}_t^2 + \mathbf{w}_t, \end{cases}$$

with $\mathbf{v}_t \sim \mathcal{N}(0, 0.1)$, $\mathbf{w}_t \sim \mathcal{L}(\sqrt{2})$, where $\mathcal{L}(\lambda)$ denotes the Laplace distribution with parameter λ, that is a symmetric exponential ($p(\mathbf{w}_t) = (\lambda/2)\exp(-\lambda|\mathbf{w}_t|)$). This toy example is considered in several works. Its interest lies in the difficulty to estimate the sign of the state at zero crossings, since the dependence of the observation upon the state is quadratic.

The figure 4.1 shows the improvement brought by particle filters compared to the EKF and UKF filters. The mean square errors for the state estimate, carried from a data vector

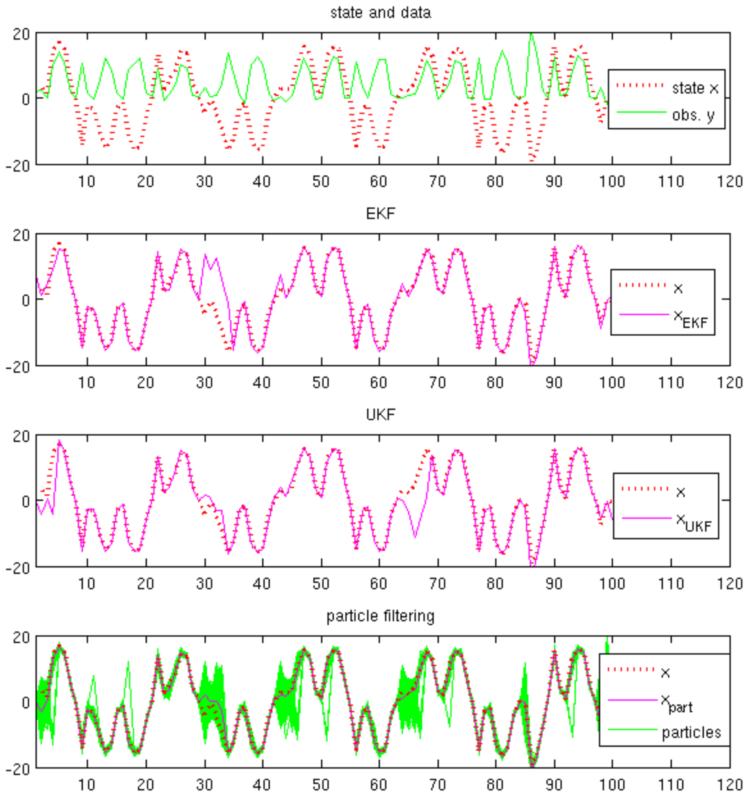

Figure 4.1: Comparison of state filtering with EKF, UKF and particle filter.

of length 10^4 are 48, 8 and 2 respectively for EKF, UKF and the particle filter. With 10 , 10^2, 10^3 and 10^4 particles, we get repective MSEs of 100, 98, 82 and 2 while standard deviations, calculated for 10^2, 10^3 and 10^4 particles, are 9.9, 9.8, and

4.4 Improvements of particle filters

Many techniques have been proposed to improve the behavior of particle filters. Here, we consider some interesting methods to perform this task.

4.4.1 MCMC move

The optimal choice of the importance distribution in the sense of minimizing the variance of the estimates would be to take for it the target distribution. Here, this would mean

$q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$. But since we are not able to draw $\mathbf{x}_t$ from p we have chosen $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$ which can be simulated quite easily in general. However, to obtain samples drawn from $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$ is feasible by resorting to the Metropolis-Hastings algorithm [Bré99, RC04, Cho02]. Letting $\hat{x}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$ denote the candidate value for $\mathbf{x}_t^{(i)}$ supplied by the particle filter, we can move toward a point with distribution closer to $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$ by applying one step (or several) of the Metropolis-Hastings algorithm. Letting $\mathbf{x}_t^{(i)*} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$ denote the new candidate, the Metropolis ratio is given by

$$\alpha = \min\left\{1, \frac{p(\mathbf{x}_t^{(i)*}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})p(\hat{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{p(\hat{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})p(\mathbf{x}_t^{(i)*}|\mathbf{x}_{t-1}^{(i)})}\right\} \tag{4.13}$$

This formula simplifies readily to

$$\alpha = \min\left\{1, \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)*})}{p(\mathbf{y}_t|\hat{x}_t^{(i)})}\right\}. \tag{4.14}$$

Then letting $u \sim \mathbf{U}_{[0,1]}$, we take

$$\begin{array}{ll} \mathbf{x}_t^{(i)} = \mathbf{x}_t^{(i)*} & \text{if } u < \alpha \\ \mathbf{x}_t^{(i)} = \hat{\mathbf{x}}_t^{(i)} & \text{if } u > \alpha. \end{array} \tag{4.15}$$

Smoothing

When it is possible to wait before estimating the distribution of $\mathbf{x}_t$, smoothing often provides significant improvements compared to filtering. We are now going to present smoothing techniques for particle filtering [GDW04]. To perform smoothing let us remark first that

$$\begin{aligned} p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) &= p(\mathbf{x}_t|\mathbf{y}_{0:t}) \textstyle\prod_{k=0}^{t-1} p(\mathbf{x}_k|\mathbf{x}_{k+1:t}, \mathbf{y}_{0:t}) \\ &= p(\mathbf{x}_t|\mathbf{y}_{0:t}) \textstyle\prod_{k=0}^{t-1} p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{y}_{0:k}) \end{aligned} \tag{4.16}$$

and that

$$p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{y}_{0:T}) \propto p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{0:k}). \tag{4.17}$$

Then, starting from knowledge of propagated approximate filter distributions $\hat{P}(d\mathbf{x}_u|\mathbf{y}_{0:u}) = \sum_{i=1}^N w_u^{(i)} \delta_{\mathbf{x}_u^{(i)}}(d\mathbf{x}_u)$ for $u = 1, \ldots, t$, it is possible to approximate $p(\mathbf{x}_u|\mathbf{x}_{u+1}, \mathbf{y}_{0:k})$ by

$$\begin{aligned} \hat{P}(d\mathbf{x}_u|\mathbf{x}_{u+1}, \mathbf{y}_{0:k}) &= \sum_{i=1}^N \frac{p(\mathbf{x}_{u+1}^{(i)}|\mathbf{x}_u^{(i)})w_u^{(i)}}{\sum_{i=1}^N p(\mathbf{x}_{u+1}^{(j)}|\mathbf{x}_u^{(j)})w_u^{(j)}} \delta_{\mathbf{x}_u^{(i)}}(d\mathbf{x}_u) \\ &= \textstyle\sum_{i=1}^N \rho_u^{(i)} \delta_{\mathbf{x}_u^{(i)}}(d\mathbf{x}_u). \end{aligned} \tag{4.18}$$

To generate a particle in the backward procedure, starting form $\tilde{\mathbf{x}}_t^{(i)} = \mathbf{x}_t(i)$, we iteratively generate $\tilde{\mathbf{x}}_u^{(i)} \sim \hat{P}(d\mathbf{x}_u|\tilde{\mathbf{x}}_{u+1}^{(i)}, \mathbf{y}_{0:T})$. The algorithm can be summarized as follows [GDW04]:

1. initialization: set $\tilde{\mathbf{x}}_t^{(i)} = x_t(i)$, $i = 1, \ldots, N$
2. iterations: for $u = t-1, t-2, \ldots, k$,
   - calculate $\rho_u^{(i)} \propto p(\tilde{\mathbf{x}}_{u+1}^{(i)} | \mathbf{x}_u^{(i)}) w_u^{(i)}$, $i = 1, \ldots, N$
   - choose $\tilde{\mathbf{x}}_u^{(i)} \sim \sum_{i=1}^{N} \rho_u^{(i)} \delta_{\mathbf{x}_u^{(i)}}(dx_u)$
3. set $\hat{p}(d\mathbf{x}_k | \tilde{x}_{\mathbf{k+1}}^{(i)}, \mathbf{y}_{0:T}) = \sum_{i=1}^{N} \rho_k^{(i)} \delta_{\tilde{\mathbf{x}}_k^{(i)}}(d\mathbf{x}_k)$.

In order to show the improvement brought by smoothing, we consider again the example of the previous section but with $\mathbf{v}_t \sim \mathcal{N}(0, 10)$ and $\mathbf{w}_t \sim \mathcal{N}(0, 1)$. This is quite a difficult example as we can see it in Fig. 4.2 by considering the large dispersion of particles supplied by the bootstrap particle filter. Here, the bootstrap filter is implemented with systematic re-sampling. 10^4 particles have been used for the filter. 10 smoothed particles have been simulated and plotted, showing that smoothing enables high concentration of the particles around the true state trajectory. Here standard deviation for the particle filter is 4.87, while it is only 1.37 for smoothed particles.

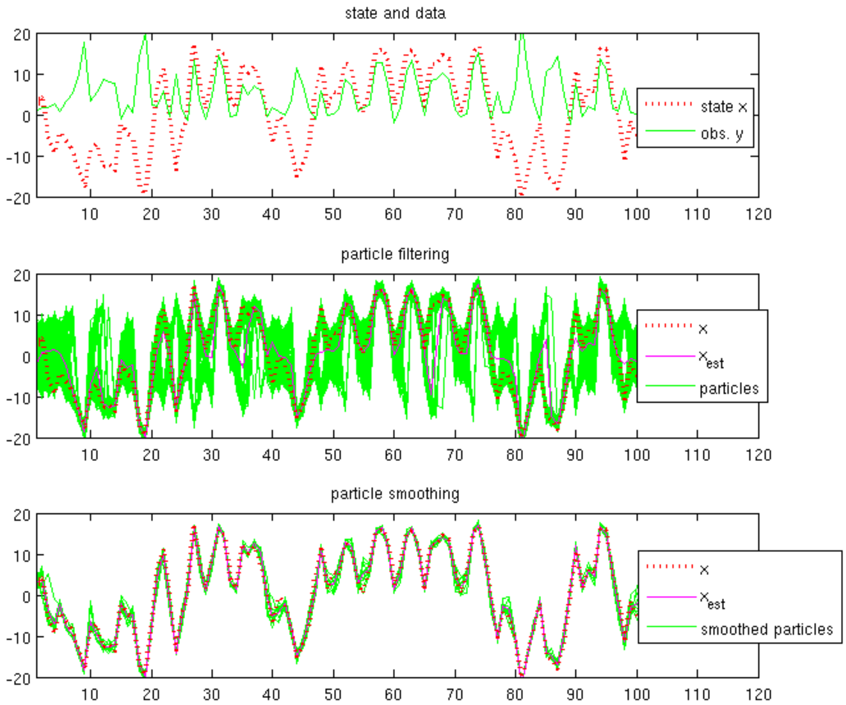

Figure 4.2: Improvement brought by particle smoothing.

4.4.2 EKF particle filter and UPF

Since generating particles from $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$ yields high variance of weights, it would be interesting to improve simulation by incorporating knowledge of $\mathbf{y}_t$ in the generating mechanism of $\mathbf{x}_t^{(i)}$. For nonlinear models, EKF and UKF yield Gaussian approximations of $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, of the form $\mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \hat{\mathbf{P}}_{t,t})$. Starting from particles for the EKF or σ-points for the UKF at time $t-1$, that are denoted by $\hat{\mathbf{x}}_{t|t-1}^{(i)}$, we can build an estimate of the form $\mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \hat{\mathbf{P}}_{t,t})$ for the distribution of $\mathbf{x}_t$. This estimate can be used as the sampling distribution $q(\mathbf{x}_t|, \mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})$ for the particle filter. These ideas are detailed in [MDFW00].

Bibliography

[AM79] B.D. Anderson and J.B. Moore. *Optimal filtering*. Springer, 1979.

[Bré99] P. Brémaud. *Markov Chains - Gibbs Fields, Monte Carlo Simulation, And Queues*. Springer, 1999.

[Cho02] T. Chonavel. *Statistical Signal Processing*. Springer, 2002.

[CML99] D. Crisan, P. Del Moral, and T. Lyons. Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields*, 5(3):293–318, 1999.

[DdFG01] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[GDW04] S.J. Godsill, A. Doucet, and M. West. Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, March 2004.

[Gew89] J. Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57(6):1317–1339, Nov. 1989.

[GL96] G.H. Golub and C.F. Van Loan. *Matrix computation*. The Johns Hopkins University Press; 3-rd edition, 3 edition, 1996.

[GSS93] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE proceedings*, 140(2):107–113, 1993.

[JU96] S. Julier and J. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Oxford University, november 1996.

[Jul02] S.J. Julier. The scaled unscented transformation. In *Proceedings of the 2002 In American Control Conference*, pages 4555–4559, november 2002.

[Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45, 1960.

[Kit96] G. Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *American Statistical Association*, 5(1):1–25, march 1996.

[MDFW00] R. Van Der Merwe, Arnaud Doucet, N. De Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/F-ENFING/TR380, Cambridge University Engineering Department, august 2000.

[PS] V.S. Pougachev and I.N. Sinitsyn. *Stochastic Differential Systems, Analysis and Filtering*.

[RC04] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

Appendix A

Matrix inversion lemma

Letting $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{E}$ denote matrices such that

$$\mathbf{A} = \mathbf{B} + \mathbf{CDE} \tag{A.1}$$

and assuming that $\mathbf{B}$ and $\mathbf{D}$ are invertible, we have

$$(\mathbf{B} + \mathbf{CDE})^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{C}(\mathbf{D}^{-1} + \mathbf{E}\mathbf{B}^{-1}\mathbf{C})^{-1}\mathbf{E}\mathbf{B}^{-1}. \tag{A.2}$$

The proof is straightforward.

Index