UNIVERSITÄT
MANNHEIM

Generalised Annealed Particle Filter
Mathematical Framework, Algorithms and Applications

Diplomarbeit am Lehrstuhl für Mathematik V
Prof. Dr. J. Potthoff
Fakultät für Mathematik und Informatik
Universität Mannheim

von

Jürgen Gall

Betreuer:
Prof. Dr. J. Potthoff
Prof. Dr. C. Schnörr

Tag der Abgabe: 22. Dezember 2005

Fakultät Mathematik & Informatik

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Mannheim, den 22. Dezember 2005

Acknowledgment

I would like to thank Prof. Dr. Jürgen Potthoff and Prof. Dr. Christoph Schnörr for helpful comments and stimulating discussions. My special thanks go to Dahlia de Candido and André Gall for providing reading material and computing power. Last but not least, I want to thank Jan Kästle and Frederic Mantlik for proof reading.

> *" 'Where shall I begin, please your Majesty?' he asked. 'Begin at the beginning,' the King said, gravely, 'and go on till you come to the end: then stop.' "*
>
> Lewis Carroll

Abstract

Jonathan Deutscher et al. introduced a new algorithm, termed annealed particle filter, for articulated body motion tracking. It is a modified particle filter that uses a continuation principle, based on annealing, to introduce the influence of narrow peaks in the fitness function, gradually. However, neither an analytical expression for the quality of the estimates nor restrictions, that are necessary for the stability of the algorithm, are given. We develop a mathematical framework, termed generalised annealed particle filter, based on the same ideas as the heuristic annealed particle filter. As a result, we are able to give estimates for the error and state conditions that are sufficient for the convergence.

Contents

List of Figures

List of Tables

List of Algorithms

1. Introduction

Many real-world applications require estimating the unknown state of a system from some given observations at each time step. In the field of computer vision, the observations are usually image sequences captured by one or more cameras, and the discrete time steps are given by the frame rate of the cameras. Even though the dynamics of the system are not known exactly in many situations, prior knowledge is generally available to construct a suitable model. In the Bayesian approach to dynamic state estimation, prior distributions for the states and likelihood functions relating these states with the observations are derived from the model. In this context, estimates of the states base on the posterior distribution obtained from Bayes' theorem. In order to avoid storing the complete data and to enable sequential processing of the observations, recursive filters are suitable for this task. These filters consist essentially of a prediction step, where the state is predicted for the next time step according to the dynamical model, and an update step, where the prediction is updated according to the latest observation.

If the model is linear and Gaussian, then the Kalman filter [Kalm60] is the optimal recursive filter in order to minimise the mean square error between the true state and its estimate. However, there are many situations where these assumptions do not hold. Various filters, such as extended Kalman filter ([Jazw70], [Gelb01]), unscented Kalman filter [JuUh97], Gaussian sum approximations [AlSo72] and grid-based filters [PoWH88], have been developed to deal with this problem. An overview is given in [AMGC02]. But these, besides the last one, are only suboptimal solutions since they approximate the nonlinearity and non-Gaussianity of the model, for example by Taylor expansion or by using a sum of Gaussian distributions. When the set of states is uncountable, the grid-based filters are computationally expensive in high dimensions.

Sequential Monte Carlo methods are recursive Bayesian filters that base on Monte Carlo simulations [HaHa67] and provide a convenient approach to approximate the posterior distributions. This technique is known as bootstrap filtering [GoSS93], condensation [IsBl96], Monte Carlo filters [KiGe96], interacting particle approximations [Mora04], survival of the fittest [KaKR95] and particle filters [DoFG01] depending on the area of research. Though the ideas go back to the 70s, these methods have be-come very popular due to the extraordinary increase of computational power only in the last few years. During this time, the methods have been successfully applied to a wide range of applications, where several examples are discussed in [DoFG01]. The mathematical fundamentals, including convergence results, have been developed further by Pierre del Moral in [Mora98] and [MoMi00]. A survey of convergence results is given in [CrDo02].

Various improvements of the particle filters have been proposed such as the regularised particle filter [DoFG01, Chapter 12] and the auxiliary particle filter [DoFG01, Chapter 13]. Another modified particle filter, termed annealed particle filter, was introduced for articulated body motion tracking by Jonathan Deutscher et al. [DeBR00]. It uses a continuation principle, based on annealing, to introduce the influence of narrow peaks in the fitness function, gradually. The algorithm is motivated by simulated annealing [KiJV83], which is a Markov chain based method for optimisation. In contrast to [BlIs00] and [HSFl00], the dynamics are very simply modelled. Godsill and Clapp [DoFG01, Chapter 7] suggested to use a similar idea for the filtering problem.

1.1 Purpose

Jonathan Deutscher et al. showed that the annealed particle filter works well for articulated body motion tracking. However, neither an analytical expression for the quality of the estimates nor restrictions, that are necessary for the stability of the algorithm, are given. Such results are very helpful for improvements, comparison with other approaches and use in different applications, as the filtering problem. Thus, we develop a mathematical framework, termed generalised annealed particle filter, based on the same ideas as the heuristic annealed particle filter. As a result, we are able to give estimates for the error and state conditions that are sufficient for the convergence. Moreover, we evaluate the various parameters of the algorithm and compare the annealed particle filter with the generic particle filter.

1.2 Outline

The thesis is divided into two parts. The first part containing Chapter 2 - 6 treats the mathematical framework of the annealed particle filter. A compact overview over the mathematical foundations, which are essential for the succeeding chapters, is given in Chapter 2. The filtering problem is stated in Chapter 3. Furthermore, a theorem is proved that establishes necessary and sufficient conditions for the convergence of an approximating sequence to the posterior distribution. The generic particle filter and its mathematical properties, such as convergence and rate of convergence, are discussed in Chapter 4. In Chapter 5, an interacting annealing algorithm is derived that combines the idea of annealing with particle filtering. Moreover, we prove the convergence of the algorithm. The mathematical framework is completed by connecting the generic particle filter with the interacting annealing algorithm in Chapter 6. In the second part, Chapter 7, two applications are used to evaluate the annealed particle filter with various parameter settings and compare this with the generic particle filter.

2. Preliminaries

We give a minimalistic summary of important probabilistic tools needed in the following chapters, where we assume that the basics of measure and probability theory, particularly conditional expectation, are known. We recommend the books [Baue90] and [Baue91], [Bill95] or [Shir84] to readers who are unfamiliar with this subject. In the following, we will often cite from these books.

2.1 Notation

The notation is similar to that used in [Mora04]. Let $(\Omega, \mathcal{F}, P)$ be a probability space and let $(E, \mathcal{E})$ be a measurable space, where $\mathcal{E}$ denotes the σ-field of Borel subsets of E.

First, we introduce some standard notations:

- $B(E)$ - set of bounded $\mathcal{E}$-measurable functions $f : E \to \mathbb{R}$;
- $C_b(E)$ - set of bounded continuous functions $f : E \to \mathbb{R}$;
- $C_c(E)$ - set of compactly supported continuous functions $f : E \to \mathbb{R}$;
- $\mathcal{P}(E)$ - set of probability measures on $\mathcal{E}$.

We have $C_c(E) \subset C_b(E)$ since a continuous function on a compact set attains its maximum and minimum values on the set.

We will also use the *supremum norm*

$$\|f\|_\infty := \sup_{x \in E} |f(x)| ,$$

for all $f \in C_b(E)$, and the *total variation distance*

$$\|\mu_1 - \mu_2\|_{TV} := \sup_{A \in \mathcal{E}} |\mu_1(A) - \mu_2(A)| ,$$

for all $\mu_1, \mu_2 \in \mathcal{P}(E)$.

The following condition is necessary so that a measure μ possesses a density f with respect to a measure ν, i.e.

$$\mu(B) = \int_B f d\nu \qquad \forall B \in \mathcal{E}.$$

Definition 2.1.1. [Baue90, Definition 17.7] A measure μ on $\mathcal{E}$ is called *absolutely continuous* with respect to a measure ν on $\mathcal{E}$ if

$$\nu(B) = 0 \Rightarrow \mu(B) = 0$$

for all $B \in \mathcal{E}$.

If $E = \mathbb{R}^d$ and ν is the Lebesgue measure on $\mathcal{B}(\mathbb{R}^d)$, or if ν is a probability measure, then the Radon-Nikodym theorem [Baue90, Theorem 17.10] shows that the condition is also sufficient. The density f is ν-almost surely unique and also called *Radon-Nikodym derivative* written as

$$\frac{d\mu}{d\nu} = f \qquad \nu - a.s.$$

2.2 Weak Convergence

Definition 2.2.1. [Baue90, Definition 30.7] Let $(\mu_n)_{n\in\mathbb{N}}$ be a sequence of probability measures on $\big(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d)\big)$ and let μ be a probability measure on $\big(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d)\big)$. Then $(\mu_n)_{n\in\mathbb{N}}$ *converges weakly* to μ if

$$\lim_{n\to\infty} \int f \, d\mu_n = \int f \, d\mu \qquad \forall f \in C_b(\mathbb{R}^d).$$

The weak convergence is commonly denoted by $\mu_n \xrightarrow{w} \mu$.

2.3 Markov processes

We consider an E-valued stochastic process $(X_t)_{t\in\mathbb{N}_0}$, that means a stochastic process with state space $(E, \mathcal{E})$.

Definition 2.3.1. [Baue91, Definition 36.1] A *kernel* on E is a function

$$K : E \times \mathcal{E} \to [0, \infty]$$

such that

1. $x \mapsto K(x, B)$ is $\mathcal{E}$-measurable $\quad \forall B \in \mathcal{E}$;
2. $B \mapsto K(x, B)$ is a measure on $\mathcal{E} \quad \forall x \in E$.

The kernel is called *Markov* if $K(x, E) = 1$.

Let $f \in B(E)$, $\mu \in \mathcal{P}(E)$ and let K and L be kernels on E. We will use the following notations:

$$\begin{aligned}
\langle \mu, f \rangle &= \int_E f(x)\, \mu(dx), \\
\langle K, f \rangle(x) &= \int_E f(y)\, K(x, dy), \\
\langle \mu, K \rangle(B) &= \int_E K(x, B)\, \mu(dx), \\
(KL)(x, B) &= \int_E L(y, B)\, K(x, dy),
\end{aligned}$$

for all $x \in E$ and $B \in \mathcal{E}$.

Definition 2.3.2. [Mora04, Definition 4.2.1] The *Dobrushin contraction* or *ergodic coefficient* $\beta(K) \in [0, 1]$ of a Markov kernel K on E is defined by

$$\beta(K) := \sup_{x_1, x_2 \in E} \|K(x_1, \cdot) - K(x_2, \cdot)\|_{TV} .$$

Let K_1 and K_2 be Markov kernels, then we have

$$\beta(K_1 K_2) \leq \beta(K_1)\, \beta(K_2), \tag{2.3.1}$$

as given in [SeVa05] and [Gida95, Section 3.2].

We define a Markov process according to [Baue91, Chapter 42] as follows:

Definition 2.3.3. An E-valued *Markov process* $X = (\Omega, \mathcal{A}, P, (X_t)_{t \in \mathbb{N}_0})$ is an E-valued stochastic process such that for $s, t \in \mathbb{N}_0$, $0 \leq s \leq t$ and $B \in \mathcal{E}$

$$P\left(X_t \in B \mid \mathcal{F}_s\right) = P\left(X_t \in B \mid X_s\right) \qquad \text{a.s.}, \tag{2.3.2}$$

where $\mathcal{F}_s := \sigma(X_s, \ldots, X_0)$. The family of Markov kernels $(K_t)_{t \in \mathbb{N}_0}$ defined by

$$K_t(x, B) := P(X_{t+1} \in B \mid X_t = x),$$

for all $x \in E$, $B \in \mathcal{E}$ and $t \in \mathbb{N}_0$, is called *family of transition kernels*. In the case where $K_t = K$ for all $t \in \mathbb{N}_0$, the Markov process is called *time-homogeneous*, otherwise *time-inhomogeneous*.

For time-homogeneous processes, we simplify the notation, and we denote by K^n the kernel for n transitions, i.e. $K^n = KK^{n-1}$ with $K^0 = Id$.

Definition 2.3.4. [RoWi01, Chapter III.6] A family of transition kernels on E is said to satisfy the *Feller property* if

$$\langle K_t, f \rangle \in C_b(E) \qquad \forall f \in C_b(E), \tag{2.3.3}$$

for all $t \in \mathbb{N}_0$.

We note that equation (2.3.2) can be rewritten as

$$E\left[1_B \circ X_t \mid \mathcal{F}_s\right] = E\left[1_B \circ X_t \mid X_s\right] \qquad \text{a.s.},$$

and therefore we get by a limit argument

$$E\left[f(X_t) \mid \mathcal{F}_s\right] = E\left[f(X_t) \mid X_s\right] \qquad \text{a.s.}, \tag{2.3.4}$$

for all $f \in B(E)$ and $0 \leq s \leq t$.

Furthermore, we deduce from equation (2.3.4) a useful result. If $A \in \mathcal{E}$, the definition of a transition kernel yields

$$\begin{aligned}
\int_{\{X_t \in A\}} f(X_{t+1})\, dP &= \int_A E\left[f(X_{t+1}) \mid X_t = x\right] P_{X_t}(dx) \\
&= \int_A \int_E f(y)\, K_t(x, dy)\, P_{X_t}(dx) \\
&= \int_A \langle K_t, f\rangle(x)\, P_{X_t}(dx) \\
&= \int_{\{X_t \in A\}} \langle K_t, f\rangle(X_t)\, dP.
\end{aligned}$$

Hence, we obtain

$$E\left[f(X_{t+1}) \mid \mathcal{F}_t\right] = E\left[f(X_{t+1}) \mid X_t\right] = \langle K_t, f\rangle(X_t) \qquad \text{a.s.}, \tag{2.3.5}$$

for all $f \in B(E)$ and $t \in \mathbb{N}_0$.

Finally, we remark that the distribution of the Markov process $(X_t)_{t\in\mathbb{N}_0}$ is uniquely determined by the family of transition kernels $(K_t)_{t\in\mathbb{N}_0}$ and an initial distribution μ since

$$P\left(X_t \in B\right) = \int_E \int_E \ldots \int_E K_{t-1}(x_{t-1}, B)\, K_{t-2}(x_{t-2}, dx_{t-1}) \ldots K_0(x_0, dx_1)\, \mu(dx_0),$$

for all $B \in \mathcal{E}$.

Definition 2.3.5. [MeTw93, Chapter 10] A probability measure ν on $\mathcal{E}$ is *invariant* for the transition kernel K if

$$\nu(B) = \langle \nu, K\rangle(B) \qquad \forall B \in \mathcal{E}.$$

In the case that the probability measure possesses a density f, we also say that K leaves f invariant.

3. Filtering problem

We state the filtering problem as discussed in [DoFG01, Chapter 2] and [CrGr99]. Furthermore, we prove a theorem that establishes necessary and sufficient conditions for the convergence of an approximating sequence to the distribution of interest, which is given by the problem, in some sense.

3.1 Filtering Problem

Let $X = (X_t)_{t\in\mathbb{N}_0}$ be an $\mathbb{R}^d$-valued Markov process, called *signal process*, with a family of transition kernels $(K_t)_{t\in\mathbb{N}_0}$ satisfying the Feller property and initial distribution η_0. Let $Y = (Y_t)_{t\in\mathbb{N}_0}$ be an $\mathbb{R}^m$-valued stochastic process, called *observation process*, defined as

$$Y_t = h_t(X_t) + W_t \qquad \text{for } t > 0,$$

and $Y_0 = 0$, where

1. $h_t : \mathbb{R}^d \to \mathbb{R}^m$ is a continuous function for all $t \in \mathbb{N}$,
2. W_t are independent m-dimensional random vectors and their distributions possess a density g_t w.r.t. the Lebesgue measure λ,
3. $g_t \in C_b(\mathbb{R}^m)$.

The filtering problem consists of computing the conditional distribution

$$\eta_t(B) := P\left(X_t \in B \mid \mathcal{G}_t\right), \tag{3.1.1}$$

for all $B \in \mathcal{B}(\mathbb{R}^d)$ or, alternatively,

$$\langle \eta_t, f\rangle = E\left[f(X_t) \mid \mathcal{G}_t\right] \qquad \text{a.s.},$$

for any function $f \in B(\mathbb{R}^d)$, where $\mathcal{G}_t := \sigma(Y_t, \ldots, Y_0)$. We also introduce the predicted conditional probability measure

$$\hat{\eta}_t(B) := P\left(X_t \in B \mid \mathcal{G}_{t-1}\right), \tag{3.1.2}$$

for all $B \in \mathcal{B}(\mathbb{R}^d)$ and $t > 0$.

We have the following recurrence relations

$$\begin{aligned} \frac{d\eta_t}{d\hat{\eta}_t} &= \frac{g_t(Y_t - h_t)}{\int_{\mathbb{R}^d} g_t(Y_t - h_t(x))\,\hat{\eta}_t(dx)} \qquad \hat{\eta}_t - a.s. &(3.1.3)\\ \hat{\eta}_{t+1} &= \langle \eta_t, K_t \rangle &(3.1.4)\end{aligned}$$

almost surely, as shown in [DoFG01, p. 20].

3.2 Convergence

Let $(\Omega, \mathcal{F}, P)$ be a probability space and let $(\mu^n)_{n\in\mathbb{N}}$ be a sequence of random probability measures, that means $\mu^n : \Omega \to \mathcal{P}(\mathbb{R}^d)$. Let either $\mu : \Omega \to \mathcal{P}(\mathbb{R}^d)$ be another random probability measure or $\mu \in \mathcal{P}(\mathbb{R}^d)$ be a deterministic probability measure. We say that the sequence $(\mu^n)_{n\in\mathbb{N}}$ *converges almost surely* to μ if

$$P\left(\omega : \mu^n(\omega) \xrightarrow{w} \mu(\omega)\right) = 1 \tag{3.2.1}$$

and

$$P\left(\omega : \mu^n(\omega) \xrightarrow{w} \mu\right) = 1, \tag{3.2.2}$$

respectively.

We note that the equations (3.2.1) or (3.2.2) imply, for any $1 \leq p < \infty$,

$$\lim_{n\to\infty} E\left[|\langle \mu^n, f\rangle - \langle \mu, f\rangle|^p\right] = 0,$$

for all $f \in C_b(\mathbb{R}^d)$. Let $f \in C_b(\mathbb{R}^d)$ and consider $\langle \mu^n, f\rangle$ and $\langle \mu, f\rangle$ as functions on Ω. Then $\langle \mu^n, f\rangle \in \mathcal{L}^p(P)$ converges almost surely to $\langle \mu, f\rangle \in \mathcal{L}^p(P)$. Since $|\langle \mu^n(\omega), f\rangle| \leq \|f\|_\infty$ for all $\omega \in \Omega$, the dominated convergence theorem [Baue90, Theorem 15.6] yields $\lim_{n\to\infty} \int |\langle \mu^n, f\rangle - \langle \mu, f\rangle|^p dP = 0$.

It is known, cf. [Baue90, Chapter 31], that there exists a countable set $\mathcal{M}$ that is dense in $C_c(\mathbb{R}^d)$, with respect to uniform convergence. By ordering the functions $\varphi \in \mathcal{M} \setminus \{0\}$ as $\varphi_1, \varphi_2, \ldots$,

$$d_{\mathcal{M}}(\mu, \nu) := \sum_{k=1}^{\infty} \frac{|\langle \mu, \varphi_k\rangle - \langle \nu, \varphi_k\rangle|}{2^k \|\varphi_k\|_\infty}$$

defines a metric on $\mathcal{P}(\mathbb{R}^d)$, which generates the weak topology

$$\lim_{n\to\infty} \nu^n = \nu \Leftrightarrow \lim_{n\to\infty} d_{\mathcal{M}}(\nu^n, \nu) = 0.$$

Thus equations (3.2.1) and (3.2.2) are equivalent to

$$P\left(\omega : \lim_{n\to\infty} d_{\mathcal{M}}(\mu^n(\omega), \mu(\omega)) = 0\right) = 1$$

and

$$P\left(\omega : \lim_{n\to\infty} d_{\mathcal{M}}(\mu^n(\omega), \mu) = 0\right) = 1,$$

respectively.

Remark 3.2.1. It is easily seen that if

$$d_k(\nu^n, \nu) := |\langle \nu^n, \varphi_k \rangle - \langle \nu, \varphi_k \rangle| \to 0 \qquad \text{as } n \to \infty,$$

for all $k \in \mathbb{N}$, then

$$\lim_{n\to\infty} d_{\mathcal{M}}(\nu^n, \nu) \to 0.$$

Let $0 < \varepsilon \leq 4$ and put $l = \lceil 2 - \log_2(\varepsilon) \rceil$, where $\lceil a \rceil$ denotes the smallest integer greater than or equal to a. Then we have

$$d_{\mathcal{M}}(\nu^n, \nu) = \sum_{k=1}^{\infty} \frac{1}{2^{k-1}} \frac{d_k(\nu^n, \nu)}{2\,\|\varphi_k\|_\infty} \leq \sum_{k=1}^{l} \frac{1}{2^k} \frac{d_k(\nu^n, \nu)}{\|\varphi_k\|_\infty} + \sum_{k=l+1}^{\infty} \frac{1}{2^{k-1}}.$$

By assumption, there exists an $n \in \mathbb{N}$ such that

$$\sum_{k=1}^{l} \frac{1}{2^k} \frac{d_k(\nu^n, \nu)}{\|\varphi_k\|_\infty} < \frac{\varepsilon}{2}.$$

Thus, we have

$$d_{\mathcal{M}}(\nu^n, \nu) < \frac{\varepsilon}{2} + \frac{1}{2^{l-1}} \leq \varepsilon.$$

Remark 3.2.2. Let us assume that

$$P\left(\omega : \lim_{n\to\infty} \langle \mu^n(\omega), \varphi \rangle = \langle \mu(\omega), \varphi \rangle\right) = 1 \qquad \forall \varphi \in \mathcal{M} \subset C_b(\mathbb{R}^d). \tag{3.2.3}$$

Then we have that for all $\varphi \in \mathcal{M}$ exists $N_\varphi \in \mathcal{F}$ such that $P(N_\varphi) = 0$ and $\langle \mu^n(\omega), \varphi \rangle \to \langle \mu(\omega), \varphi \rangle$ for all $\omega \in N_\varphi^{\complement}$. Using Remark 3.2.1, we get

$$d_{\mathcal{M}}(\mu^n(\omega), \mu(\omega)) \to 0 \qquad \forall \omega \in \bigcap_{\varphi \in \mathcal{M}} N_\varphi^{\complement},$$

which yields $P\left(\omega : d_{\mathcal{M}}(\mu^n(\omega), \mu(\omega)) \to 0\right) = 1$. This shows that it is sufficient to prove (3.2.3) in order to obtain (3.2.1). We get the same result in the case of a deterministic probability measure μ.

We will state conditions for the convergence of the approximating sequences $(\eta_t^n)_{n\in\mathbb{N}}$ and $(\hat{\eta}_t^n)_{n\in\mathbb{N}}$ to η_t and $\hat{\eta}_t$, respectively. We assume that η_t^n and $\hat{\eta}_t^n$ are random probability measures, such that

$$\int_{\mathbb{R}^d} g_t(Y_t - h_t(x))\, \hat{\eta}_t^n(dx) > 0,$$

for all $n, t \in \mathbb{N}$. Let also $\bar{\eta}_t^n$ be defined as random probability measure that is absolutely continuous with respect to $\hat{\eta}_t^n$ and such that

$$\frac{d\bar{\eta}_t^n}{d\hat{\eta}_t^n} = \frac{g_t(Y_t - h_t)}{\int_{\mathbb{R}^d} g_t(Y_t - h_t(x))\, \hat{\eta}_t^n(dx)} \qquad \hat{\eta}_t^n - a.s. \tag{3.2.4}$$

almost surely, for $n, t \in \mathbb{N}$.

Theorem 3.2.3. *The sequences $(\eta_t^n)_{n\in\mathbb{N}}$ and $(\hat{\eta}_{t+1}^n)_{n\in\mathbb{N}}$ converge almost surely to η_t and $\hat{\eta}_{t+1}$, respectively, i.e. in the sense of (3.2.1), for all $t \in \mathbb{N}_0$ if and only if*

$$P\left(\omega : \eta_0^n(\omega) \xrightarrow{w} \eta_0\right) = 1, \tag{3.2.5}$$

$$P\left(\omega : \lim_{n\to\infty} d_{\mathcal{M}}\left(\hat{\eta}_{t+1}^n(\omega), \langle \eta_t^n(\omega), K_t\rangle\right) = 0\right) = 1, \tag{3.2.6}$$

$$P\left(\omega : \lim_{n\to\infty} d_{\mathcal{M}}\left(\eta_{t+1}^n(\omega), \bar{\eta}_{t+1}^n(\omega)\right) = 0\right) = 1, \tag{3.2.7}$$

for all $t \in \mathbb{N}_0$.

Proof: *"⇐"*: We will show that $(\eta_t^n)_{n\in\mathbb{N}}$ converges to η_t by induction on t. The convergence of the sequence $(\hat{\eta}_{t+1}^n)_{n\in\mathbb{N}}$ to $\hat{\eta}_{t+1}$ also follows from the proof.

Let $t = 0$. Then the convergence of $(\eta_0^n)_{n\in\mathbb{N}}$ is given by condition (3.2.5). Assume that for a fixed $t \in \mathbb{N}_0$

$$P\left(\omega : \eta_t^n(\omega) \xrightarrow{w} \eta_t(\omega)\right) = 1.$$

Let $\varphi \in C_b(\mathbb{R}^d)$. Then the Feller property (2.3.3) yields $\langle K_t, \varphi\rangle \in C_b(\mathbb{R}^d)$. Using Fubini's theorem and the induction hypothesis, we obtain

$$|\langle\langle \eta_t^n, K_t\rangle, \varphi\rangle - \langle\langle \eta_t, K_t\rangle, \varphi\rangle| = |\langle \eta_t^n, \langle K_t, \varphi\rangle\rangle - \langle \eta_t, \langle K_t, \varphi\rangle\rangle| \to 0 \tag{3.2.8}$$

almost surely, as $n \to \infty$.

Since $\hat{\eta}_{t+1} = \langle \eta_t, K_t\rangle$ by (3.1.4), we get

$$d_{\mathcal{M}}\left(\hat{\eta}_{t+1}^n, \hat{\eta}_{t+1}\right) \le d_{\mathcal{M}}\left(\hat{\eta}_{t+1}^n, \langle \eta_t^n, K_t\rangle\right) + d_{\mathcal{M}}\left(\langle \eta_t^n, K_t\rangle, \langle \eta_t, K_t\rangle\right) \qquad \text{a.s.}$$

The first term of the right hand side converges almost surely to zero according to (3.2.6), and also the second term converges to zero, which follows from (3.2.8) and Remark 3.2.2. Hence, we have

$$P\left(\omega : \hat{\eta}_{t+1}^n(\omega) \xrightarrow{w} \hat{\eta}_{t+1}(\omega)\right) = 1. \tag{3.2.9}$$

Let $\varphi \in C_b(\mathbb{R}^d)$. We write $g_{t+1}^{Y_{t+1}}(\omega)$ instead of $g_{t+1}(Y_{t+1}(\omega) - h_{t+1})$ and note that $g_{t+1}^{Y_{t+1}}(\omega) \in C_b(\mathbb{R}^d)$ for all $\omega \in \Omega$. Using equations (3.1.3) and (3.2.4), we obtain

$$\begin{aligned}
\left|\langle \bar{\eta}_{t+1}^n, \varphi\rangle - \langle \eta_{t+1}, \varphi\rangle\right| &= \left|\frac{\langle \hat{\eta}_{t+1}^n, \varphi g_{t+1}^{Y_{t+1}}\rangle}{\langle \hat{\eta}_{t+1}^n, g_{t+1}^{Y_{t+1}}\rangle} - \frac{\langle \hat{\eta}_{t+1}, \varphi g_{t+1}^{Y_{t+1}}\rangle}{\langle \hat{\eta}_{t+1}, g_{t+1}^{Y_{t+1}}\rangle}\right| \\
&\le \left|\frac{\langle \hat{\eta}_{t+1}^n, \varphi g_{t+1}^{Y_{t+1}}\rangle}{\langle \hat{\eta}_{t+1}^n, g_{t+1}^{Y_{t+1}}\rangle} - \frac{\langle \hat{\eta}_{t+1}^n, \varphi g_{t+1}^{Y_{t+1}}\rangle}{\langle \hat{\eta}_{t+1}, g_{t+1}^{Y_{t+1}}\rangle}\right| \\
&\quad + \left|\frac{\langle \hat{\eta}_{t+1}^n, \varphi g_{t+1}^{Y_{t+1}}\rangle}{\langle \hat{\eta}_{t+1}, g_{t+1}^{Y_{t+1}}\rangle} - \frac{\langle \hat{\eta}_{t+1}, \varphi g_{t+1}^{Y_{t+1}}\rangle}{\langle \hat{\eta}_{t+1}, g_{t+1}^{Y_{t+1}}\rangle}\right| \\
&\le \frac{\|\varphi\|_\infty}{\langle \hat{\eta}_{t+1}, g_{t+1}^{Y_{t+1}}\rangle}\left|\langle \hat{\eta}_{t+1}^n, g_{t+1}^{Y_{t+1}}\rangle - \langle \hat{\eta}_{t+1}, g_{t+1}^{Y_{t+1}}\rangle\right| \\
&\quad + \frac{1}{\langle \hat{\eta}_{t+1}, g_{t+1}^{Y_{t+1}}\rangle}\left|\langle \hat{\eta}_{t+1}^n, \varphi g_{t+1}^{Y_{t+1}}\rangle - \langle \hat{\eta}_{t+1}, \varphi g_{t+1}^{Y_{t+1}}\rangle\right|
\end{aligned} \tag{3.2.10}$$

almost surely. Since the right hand side converges to zero almost surely as $n \to \infty$ due to (3.2.9), condition (3.2.7) gives

$$d_{\mathcal{M}}\left(\eta_{t+1}^{n}, \eta_{t+1}\right) \leq d_{\mathcal{M}}\left(\eta_{t+1}^{n}, \bar{\eta}_{t+1}^{n}\right) + d_{\mathcal{M}}\left(\bar{\eta}_{t+1}^{n}, \eta_{t+1}\right) \to 0$$

almost surely, as $n \to \infty$.

"$\Rightarrow$": We now assume that $d_{\mathcal{M}}\left(\eta_t^n, \eta_t\right)$ and $d_{\mathcal{M}}\left(\hat{\eta}_{t+1}^{n}, \hat{\eta}_{t+1}\right)$ converge to zero almost surely, for all $t \in \mathbb{N}_0$. This implies in particular condition (3.2.5). Since $\hat{\eta}_{t+1} = \langle \eta_t, K_t \rangle$, we have furthermore

$$d_{\mathcal{M}}\left(\hat{\eta}_{t+1}^{n}, \langle \eta_t^n, K_t \rangle\right) \leq d_{\mathcal{M}}\left(\hat{\eta}_{t+1}^{n}, \hat{\eta}_{t+1}\right) + d_{\mathcal{M}}\left(\langle \eta_t^n, K_t \rangle, \langle \eta_t, K_t \rangle\right) \qquad \text{a.s.}$$

Applying again (3.2.8) and Remark 3.2.2, we obtain condition (3.2.6). Finally, we get condition (3.2.7) along the lines of (3.2.10) by using the triangle inequality

$$d_{\mathcal{M}}(\eta_{t+1}^{n}, \bar{\eta}_{t+1}^{n}) \leq d_{\mathcal{M}}(\eta_{t+1}^{n}, \eta_{t+1}) + d_{\mathcal{M}}(\bar{\eta}_{t+1}^{n}, \eta_{t+1}).$$

□

4. Particle Filters

Particle filters provide a convenient approach to approximate the distribution of interest without restriction to the linear and Gaussian case. This technique is also known as bootstrap filtering [GoSS93], condensation [IsBl98], Monte Carlo filters [KiGe96], interacting particle approximations [Mora04] and survival of the fittest [KaKR95] depending on the area of research. In the following, we introduce a basic particle filter and discuss its mathematical properties, such as convergence and rate of convergence.

4.1 Generic Particle Filter

As in Chapter 3, we assume that the signal process $X = (X_t)_{t\in\mathbb{N}_0}$ is an $\mathbb{R}^d$-valued Markov process with a family of transition kernels $(K_t)_{t\in\mathbb{N}_0}$ satisfying the Feller property and initial distribution η_0 and that the observation process $Y = (Y_t)_{t\in\mathbb{N}_0}$ is an $\mathbb{R}^m$-valued stochastic process.

The generic particle filter (Algorithm 4.1) is a commonly used particle filter that provides a basis for further developments and modifications for diverse applications. The algorithm consists of the four steps "Initialisation", "Prediction", "Updating" and "Resampling". During the initialisation, we sample n times from the initial distribution η_0. By saying that we sample $x^{(i)}$ from a distribution μ, for $i = 1, \ldots, n$, we mean that we simulate n independent random samples, also named particles, according to μ. Hence, the n random variables $(X_{0,0}^{(i)})_{1\le i\le n}$ are independent and identically distributed (i.i.d.) according to η_0. Afterwards, the values of the particles are predicted for the next time step according to the dynamics of the signal process. During the "Updating" step, each predicted particle is weighted by the likelihood function $g_t(y_t - h_t(\cdot))$, which is determined by the observation process. The "Resampling" step can be regarded as a special case of a "Selection" step. The particles are selected in accordance with the weighting function g_t. This step gives birth to some particles at the expense of light particles which die. The "Resampling" step is not unique for the particle filters, for example, branching procedures are also used, see for instance [CrGr99], [CrML99] or [DoFG01]. We restrict ourselves to algorithms using the stated resampling procedure. The particle system is also called

Algorithm 4.1 Generic Particle Filter

Requires: n number of particles and η_0, $(K_t)_{t\in\mathbb{N}_0}$, $(g_t)_{t\in\mathbb{N}}$, $(h_t)_{t\in\mathbb{N}}$ as defined in Chapter 3

1. Initialisation
   - $t \leftarrow 0$
   - For $i = 1, \ldots, n$, sample $x_{0,0}^{(i)}$ from η_0
2. Prediction
   - For $i = 1, \ldots, n$, sample $\bar{x}_{t+1,0}^{(i)}$ from $K_t(x_{t,0}^{(i)}, \cdot)$
3. Updating
   - For $i = 1, \ldots, n$, set $\pi_{t+1,0}^{(i)} \leftarrow g_{t+1}(y_{t+1} - h_{t+1}(\bar{x}_{t+1,0}^{(i)}))$
   - For $i = 1, \ldots, n$, set $\pi_{t+1,0}^{(i)} \leftarrow \frac{\pi_{t+1,0}^{(i)}}{\sum_{j=1}^{n} \pi_{t+1,0}^{(j)}}$
4. Resampling
   - For $i = 1, \ldots, n$, set $x_{t+1,0}^{(i)} \leftarrow \bar{x}_{t+1,0}^{(j)}$ with probability $\pi_{t+1,0}^{(j)}$
   - $t \leftarrow t + 1$ and go to step 2

interacting particle system [Mora98] since the particles are not independent after resampling.

For the case of a one-dimensional signal process, the operation of the algorithm is illustrated in Figure 4.1, where the grey circles represent the unweighted particles after the "Prediction" step and the black circles represent the weighted particles after the "Updating" step. While the horizontal positions of the particles indicate their values in the state space of the signal process, the diameters of the black circles indicate the particle weights, that is the larger the diameter the greater the weight. As illustrated, the particles with great weight generate more offspring than particles with lower weight during the "Resampling" step. In order to discuss the mathematical properties of the algorithm, we use the following definitions.

Definition 4.1.1. [MacC00] A *weighted particle* is a pair (x, π) where $x \in \mathbb{R}^d$ and $\pi \in [0, 1]$. A *weighted particle set* S is a sequence of finite sets of random variables whose values are weighted particles: the nth member of the sequence is a set of n random variables

$$S^{(n)} = \{(X^{(1)}, \Pi^{(1)}), \ldots, (X^{(n)}, \Pi^{(n)})\} ,$$

such that $\sum_{i=1}^{n} \Pi_i^{(n)} = 1$.

It is clear that every weighted particle set determines a sequence of random probability measures by

$$\sum_{i=1}^{n} \Pi^{(i)} \delta_{X^{(i)}} ,$$

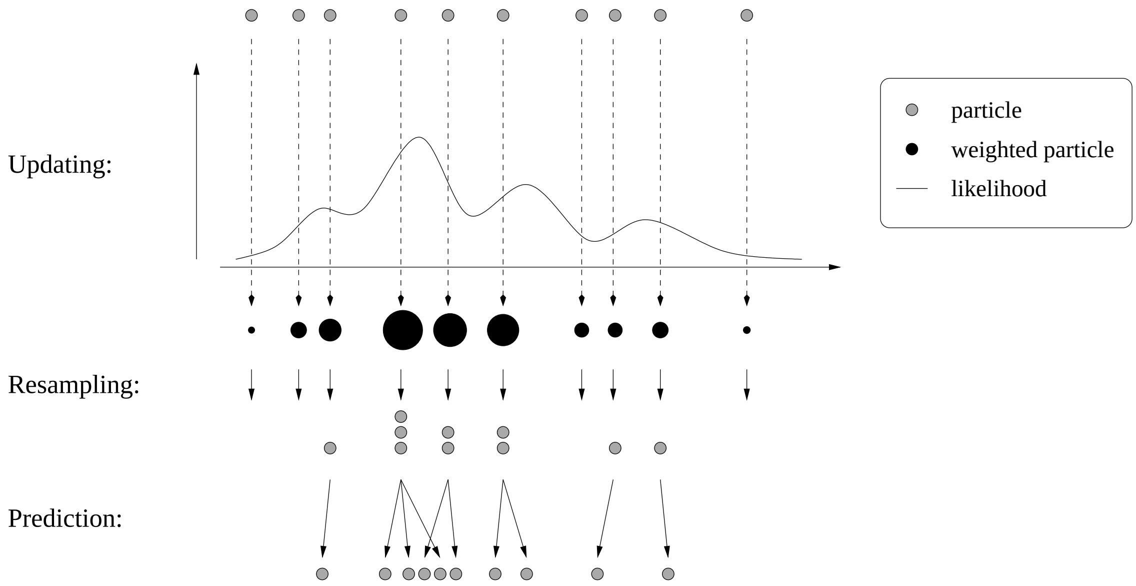


Figure 4.1: Operation of the generic particle filter.

for $n \in \mathbb{N}$.

The idea now is to approximate the conditional distribution η_t (3.1.1) by an appropriate weighted particle set. We note that each step of the generic particle filter defines a particle set and consequently a random probability measure:

$$\begin{aligned}
\eta_0^n(\omega) &:= \frac{1}{n}\sum_{i=1}^{n} \delta_{X_{0,0}^{(i)}(\omega)} && \text{(Initialisation);}\\
\hat{\eta}_t^n(\omega) &:= \frac{1}{n}\sum_{i=1}^{n} \delta_{\bar{X}_{t,0}^{(i)}(\omega)} && \text{(Prediction);}\\
\bar{\eta}_t^n(\omega) &:= \sum_{i=1}^{n} \Pi_{t,0}^{(i)}(\omega)\delta_{\bar{X}_{t,0}^{(i)}(\omega)} && \text{(Updating);}\\
\eta_t^n(\omega) &:= \frac{1}{n}\sum_{i=1}^{n} \delta_{X_{t,0}^{(i)}(\omega)} && \text{(Resampling).}
\end{aligned}$$

Using these definitions, the algorithm is illustrated by the three separate steps

$$\eta_t^n \xrightarrow{Prediction} \hat{\eta}_{t+1}^n \xrightarrow{Updating} \bar{\eta}_{t+1}^n \xrightarrow{Resampling} \eta_{t+1}^n.$$

4.2 Convergence

Before we use the results from Section 3.2 to prove the convergence of the generic particle filter, we need to establish that $\bar{\eta}_t^n$ satisfies (3.2.4). Obviously, $\bar{\eta}_t^n$ is absolutely continuous with respect to $\hat{\eta}_t^n$ for $t, n \in \mathbb{N}$. Let $A \in \mathcal{B}(\mathbb{R}^d)$. Then we have

$$\begin{aligned}
\bar{\eta}_t^n(\omega)(A) &= \frac{\frac{1}{n}\sum_{i=1}^{n} g_t(Y_t(\omega) - h_t(\bar{X}_{t,0}^{(i)}(\omega)))\delta_{\bar{X}_{t,0}^{(i)}(\omega))}(A)}{\frac{1}{n}\sum_{i=1}^{n} g_t(Y_t(\omega) - h_t(\bar{X}_{t,0}^{(i)}(\omega)))}\\
&= \frac{\int_A g_t(Y_t(\omega) - h_t(x))\, \hat{\eta}_t^n(\omega)(dx)}{\int_{\mathbb{R}^d} g_t(Y_t(\omega) - h_t(x))\, \hat{\eta}_t^n(\omega)(dx)},
\end{aligned}$$

for all $\omega \in \Omega$, which yields (3.2.4).

Since the assumptions for Theorem 3.2.3 are satisfied, we have

$$P\left(\omega : \eta_t^n(\omega) \xrightarrow{w} \eta_t(\omega)\right) = 1,$$

for all $t \in \mathbb{N}_0$, if the equations (3.2.5), (3.2.6) and (3.2.7) hold for all $t \in \mathbb{N}_0$. We will prove by the following three lemmas that these equations are achieved by the generic particle filter.

Lemma 4.2.1.

$$P\left(\omega : \eta_0^n(\omega) \xrightarrow{w} \eta_0\right) = 1.$$

Proof: $(X_{0,0}^{(i)})_{1\leq i\leq n}$ are n i.i.d. random variables, which are distributed according to η_0 as mentioned above. Let $\varphi \in C_b(\mathbb{R}^d)$. Then by the strong law of large numbers [Baue91, Theorem 12.1], we have

$$P\left(\omega \ : \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \varphi\left(X_{0,0}^{(i)}(\omega)\right) = \langle \eta_0, \varphi\rangle\right) = 1,$$

and thus

$$P\left(\omega \ : \lim_{n\to\infty} \langle \eta_0^n(\omega), \varphi\rangle = \langle \eta_0, \varphi\rangle\right) = 1.$$

Hence, the lemma follows from Remark 3.2.2. □

Lemma 4.2.2.

$$P\left(\omega : \lim_{n\to\infty} d_{\mathcal{M}}\left(\hat{\eta}_{t+1}^n(\omega), \langle \eta_t^n(\omega), K_t\rangle\right) = 0\right) = 1 \qquad \forall t \in \mathbb{N}_0.$$

Proof: We note that the value of $\bar{x}_{t+1,0}^{(i)}$ during the prediction step depends only on $x_{t,0}^{(i)}$ and not on the values of $x_{t,0}^{(j)}$ for $j \neq i$, i.e.

$$E\left[\bar{X}_{t+1,0}^{(i)} \,\middle|\, X_{t,0}^{(1)} = x_{t,0}^{(1)}, \ldots, X_{t,0}^{(n)} = x_{t,0}^{(n)}\right] = E\left[\bar{X}_{t+1,0}^{(i)} \,\middle|\, X_{t,0}^{(i)} = x_{t,0}^{(i)}\right],$$

for all i. We define $\mathcal{H}_t := \sigma(X_t^{(j)}; 1 \leq j \leq n)$, and let $A \in \mathcal{B}(\mathbb{R}^d) \otimes \ldots \otimes \mathcal{B}(\mathbb{R}^d)$ and $\varphi \in C_b(\mathbb{R}^d)$. Then we obtain, similarly to (2.3.5),

$$\begin{aligned}
&\int_{\{(X_{t,0}^{(1)},\ldots,X_{t,0}^{(n)})\in A\}} \varphi(\bar{X}_{t+1,0}^{(i)})\, dP \\
&= \int_A E\left[\varphi(\bar{X}_{t+1,0}^{(i)}) \,\middle|\, X_{t,0}^{(1)} = x_{t,0}^{(1)}, \ldots, X_{t,0}^{(n)} = x_{t,0}^{(n)}\right] P_{X_{t,0}^{(1)}\otimes\ldots\otimes X_{t,0}^{(n)}}(d(x_{t,0}^{(1)}, \ldots, x_{t,0}^{(n)})) \\
&= \int_A \langle K_t, \varphi\rangle(x_{t,0}^{(i)})\, P_{X_{t,0}^{(1)}\otimes\ldots\otimes X_{t,0}^{(n)}}(d(x_{t,0}^{(1)}, \ldots, x_{t,0}^{(n)})) \\
&= \int_{\{(X_{t,0}^{(1)},\ldots,X_{t,0}^{(n)})\in A\}} \langle K_t, \varphi\rangle(X_{t,0}^{(i)})\, dP.
\end{aligned}$$

Hence, we have

$$E\left[\varphi(\bar{X}_{t+1,0}^{(i)}) \mid \mathcal{H}_t\right] = \langle K_t, \varphi\rangle(X_{t,0}^{(i)}) \qquad \text{a.s.,} \tag{4.2.1}$$

for all $1 \leq i \leq n$, $\varphi \in C_b(\mathbb{R}^d)$ and $t \in \mathbb{N}_0$.

Since the random variables $\bar{X}^{(i)}_{t+1,0}$ are conditionally independent w.r.t. $\mathcal{H}_t$, we get

$$\begin{aligned}
&E\left[\left(\langle \hat{\eta}^n_{t+1}, \varphi\rangle - \langle\langle \eta^n_t, K_t\rangle, \varphi\rangle\right)^4\right] \\
&\quad = \frac{1}{n^4} E\left[\left(\sum_{i=1}^n \left(\varphi(\bar{X}^{(i)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(i)}_{t,0})\right)\right)^4\right] \\
&\quad = \frac{1}{n^4} \left\{ \sum_{i=1}^n E\left[\left(\varphi(\bar{X}^{(i)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(i)}_{t,0})\right)^4\right] \right. \\
&\qquad + 6 \sum_{1\leq i<j\leq n} E\left[\left(\varphi(\bar{X}^{(i)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(i)}_{t,0})\right)^2 \left(\varphi(\bar{X}^{(j)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(j)}_{t,0})\right)^2\right] \\
&\qquad \left. + \sum_{i=1}^n E\left[\left(\varphi(\bar{X}^{(i)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(i)}_{t,0})\right) A_i\right] \right\},
\end{aligned}$$

where A_i can be written as the sum of the terms

$$c \prod_{\substack{j=1 \\ j\neq i}}^n \left(\varphi(\bar{X}^{(j)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(j)}_{t,0})\right)^{k_j}$$

with $c \in \mathbb{N}$ and $k_j \in \mathbb{N}_0$, such that $\sum_{j\neq i} k_j = 3$. Since $\bar{X}^{(i)}_{t+1,0}$ and A_i are conditionally independent w.r.t. $\mathcal{H}_t$ for all i, (4.2.1) yields

$$\begin{aligned}
&E\left[\left(\varphi(\bar{X}^{(i)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(i)}_{t,0})\right) A_i\right] \\
&\quad = E\left[E\left[\left(\varphi(\bar{X}^{(i)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(i)}_{t,0})\right) A_i \,\middle|\, \mathcal{H}_t\right]\right] \\
&\quad = E\left[E\left[\varphi(\bar{X}^{(i)}_{t+1,0}) - \langle K_t, \varphi\rangle(X^{(i)}_{t,0}) \,\middle|\, \mathcal{H}_t\right] E\left[A_i \,\middle|\, \mathcal{H}_t\right]\right] = 0.
\end{aligned}$$

Therefore we have

$$E\left[\left(\langle \hat{\eta}^n_{t+1}, \varphi\rangle - \langle\langle \eta^n_t, K_t\rangle, \varphi\rangle\right)^4\right] \leq \frac{1}{n^4}\left(48\, n^2 \parallel \varphi \parallel^4_\infty - 32\, n \parallel \varphi \parallel^4_\infty\right) \leq \frac{48 \parallel \varphi \parallel^4_\infty}{n^2}.$$

Let $\varepsilon > 0$. Markov's inequality [Bill95, Section 21] gives

$$\sum_{n=1}^\infty P\left(\mid \langle \hat{\eta}^n_{t+1}, \varphi\rangle - \langle\langle \eta^n_t, K_t\rangle, \varphi\rangle \mid \geq \varepsilon\right) \leq \frac{1}{\varepsilon^4} \sum_{n=1}^\infty E\left[\left(\langle \hat{\eta}^n_{t+1}, \varphi\rangle - \langle\langle \eta^n_t, K_t\rangle, \varphi\rangle\right)^4\right] < \infty,$$

and thus the Borel-Cantelli lemma [Shir84, Chapter II.10] yields

$$P\left(\limsup_n \left\{\mid \langle \hat{\eta}^n_{t+1}, \varphi\rangle - \langle\langle \eta^n_t, K_t\rangle, \varphi\rangle \mid \geq \varepsilon\right\}\right) = 0.$$

Consequently, the lemma is proved according to [Shir84, Corollary II.10.1] and Remark 3.2.2. □

The "Resampling" step can be modelled as follows. Let $(\Omega', \mathcal{F}', Q)$ be another probability space on which a sequence of i.i.d. real random variables $(\kappa_i)_{1\leq i\leq n}$, uniformly distributed on the interval $[0,1]$, is defined. We set on the extended probability space $(\tilde{\Omega}, \tilde{\mathcal{F}}, \tilde{P}) := (\Omega \times \Omega', \mathcal{F} \otimes \mathcal{F}', P \otimes Q)$ the resampling process

$$X_{t,0}^{(i)}(\omega, \omega') := \bar{X}_{t,0}^{(J_t^i(\omega,\omega'))}(\omega),$$

for $1 \le i \le n$, where

$$J_t^i(\omega, \omega') := \sum_{l=1}^{n} l \times 1_{\iota_t^l(\omega)}(\kappa_i(\omega'))$$

and

$$\begin{aligned}
\iota_t^1(\omega) &:= \left[0, \Pi_{t,0}^{(1)}(\omega)\right), \\
\iota_t^l(\omega) &:= \left[\sum_{i=1}^{l-1} \Pi_{t,0}^{(i)}(\omega), \sum_{i=1}^{l} \Pi_{t,0}^{(i)}(\omega)\right) \quad \text{for } 1 < l < n, \\
\iota_t^n(\omega) &:= \left[\sum_{i=1}^{n-1} \Pi_{t,0}^{(i)}(\omega), 1\right].
\end{aligned}$$

Without loss of generality, we assume that the random variables $(\kappa_i)_{1 \le i \le n}$ are defined on $(\Omega, \mathcal{F}, P)$, i.e.

$$\eta_t^n(\omega) = \frac{1}{n} \sum_{i=1}^{n} \delta_{X_{t,0}^{(i)}(\omega)} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\bar{X}_{t,0}^{(J_t^i(\omega))}(\omega)}.$$

Lemma 4.2.3.

$$P\left(\omega : \lim_{n\to\infty} d_{\mathcal{M}}\left(\eta_{t+1}^n(\omega), \bar{\eta}_{t+1}^n(\omega)\right) = 0\right) = 1 \qquad \forall t \in \mathbb{N}_0.$$

Proof: First, we note that

$$P\left(J_{t+1}^i = l \mid \bar{\mathcal{H}}_{t+1}\right) = \Pi_{t+1,0}^{(l)} \qquad \text{a.s.},$$

for $1 \le i, l \le n$, where $\bar{\mathcal{H}}_{t+1} := \sigma(\bar{X}_{t+1,0}^{(i)}; 1 \le i \le n)$. Let $\varphi \in C_b(\mathbb{R}^d)$. Then we have

$$\begin{aligned}
E\left[\varphi(\bar{X}_{t+1,0}^{(J_{t+1}^i)}) \,\middle|\, \bar{\mathcal{H}}_{t+1}\right] &= \sum_{l=1}^{n} E\left[\varphi(\bar{X}_{t+1,0}^{(l)}) \,\middle|\, \bar{\mathcal{H}}_{t+1}\right] P\left(J_{t+1}^i = l \,\middle|\, \bar{\mathcal{H}}_{t+1}\right) \\
&= \sum_{l=1}^{n} \varphi(\bar{X}_{t+1,0}^{(l)}) \Pi_{t+1,0}^{(l)} \\
&= \langle \bar{\eta}_{t+1}^n, \varphi \rangle \qquad \text{a.s.} \qquad (4.2.2)
\end{aligned}$$

Since the random variables $\bar{X}_{t+1,0}^{(J_{t+1}^i)}$ are conditionally independent w.r.t. $\bar{\mathcal{H}}_{t+1}$, we get

$$\begin{aligned}
&E\left[\left(\langle \eta_{t+1}^n, \varphi \rangle - \langle \bar{\eta}_{t+1}^n, \varphi \rangle\right)^4\right] \\
&= \frac{1}{n^4} \left\{ \sum_{i=1}^{n} E\left[\left(\varphi(\bar{X}_{t+1,0}^{(J_{t+1}^i)}) - \langle \bar{\eta}_{t+1}^n, \varphi \rangle\right)^4\right] \right. \\
&\quad + 6 \sum_{1 \le i < j \le n} E\left[\left(\varphi(\bar{X}_{t+1,0}^{(J_{t+1}^i)}) - \langle \bar{\eta}_{t+1}^n, \varphi \rangle\right)^2 \left(\varphi(\bar{X}_{t+1,0}^{(J_{t+1}^j)}) - \langle \bar{\eta}_{t+1}^n, \varphi \rangle\right)^2\right] \\
&\quad \left. + \sum_{i=1}^{n} E\left[\left(\varphi(\bar{X}_{t+1,0}^{(J_{t+1}^i)}) - \langle \bar{\eta}_{t+1}^n, \varphi \rangle\right) A_i\right] \right\},
\end{aligned}$$

where $\bar{X}_{t+1,0}^{(J_{t+1}^i)}$ and A_i are conditionally independent w.r.t. $\bar{\mathcal{H}}_{t+1}$ for all i. Thus (4.2.2) yields

$$\begin{aligned}
&E\left[\left(\varphi(\bar{X}_{t+1,0}^{(J_{t+1}^i)}) - \langle \bar{\eta}_{t+1}^n, \varphi\rangle\right) A_i\right] \\
&\quad = E\left[E\left[\varphi(\bar{X}_{t+1,0}^{(J_{t+1}^i)}) - \langle \bar{\eta}_{t+1}^n, \varphi\rangle \,\middle|\, \bar{\mathcal{H}}_{t+1}\right] E\left[A_i \,\middle|\, \bar{\mathcal{H}}_{t+1}\right]\right] = 0.
\end{aligned}$$

Therefore we have

$$E\left[\left(\langle \eta_{t+1}^n, \varphi\rangle - \langle \bar{\eta}_{t+1}^n, \varphi\rangle\right)^4\right] \le \frac{48 \parallel \varphi \parallel_\infty^4}{n^2}.$$

The lemma follows by the same arguments as used in Lemma 4.2.2. □

In summary, we have proved the convergence of the generic particle filter in the sense of

Theorem 4.2.4.

$$P\left(\omega : \eta_t^n(\omega) \xrightarrow{w} \eta_t(\omega)\right) = 1.$$

4.3 Rate of Convergence

The convergence of the mean square error toward zero follows directly from Theorem 4.2.4, as seen from equation (3.2). When applying, however, one is not only interested in the convergence of the generic particle filter but also in the rate of convergence. The latter additionally provides an estimate for the number of particles needed to achieve a certain level of error.

We omit the proof for the following theorem, which gives an estimate for the convergence rate and can be found in [CrDo02], since similar arguments as in Section 4.2 are used.

Theorem 4.3.1. *For all $t \in \mathbb{N}_0$, there exists c_t independent of n such that*

$$E\left[\left(\langle \eta_t^n, \varphi\rangle - \langle \eta_t, \varphi\rangle\right)^2\right] \le c_t \frac{\|\varphi\|_\infty^2}{n}, \tag{4.3.1}$$

for all $\varphi \in B(\mathbb{R}^d)$.

Equation (4.3.1) shows that the rate of convergence of the mean square error is of order $1/n$. However, c_t depends on t and, without any additional assumption, c_t actually increases over time. This is not very satisfactory in applications as this implies that one needs an increasingly larger number of particles as time t increases to ensure a given precision. We will establish an in time uniform convergence result under additional assumptions on the filtering problem. The idea of preventing an increasing error is to ensure that any error is forgotten fast enough. For this purpose, we define a so-called mixing condition in accordance with [GlOu04] and [MoGu01].

Definition 4.3.2. A kernel on E is called *mixing* if there exists a constant $0 < \varepsilon \le 1$ and a measure μ on E such that

$$\varepsilon\mu(B) \le K(x, B) \le \frac{1}{\varepsilon}\,\mu(B), \tag{4.3.2}$$

for all $x \in E$ and $B \in \mathcal{E}$.

This strong assumption means that the measure $K(x,\cdot)$ depends very "weakly" on x. It can typically only be established when $E \subset \mathbb{R}^d$ is a bounded subset. We give two examples where the kernels are not mixing.

Example 4.3.3. Let $E = \{a,b\}$ and $K(x,B) := \varepsilon_x(B)$ for all $x \in E$ and $B \in \mathcal{E}$, where $\mathcal{E} = \{\emptyset, \{a\}, \{b\}, E\}$. Assume that K is mixing. Then from inequality (4.3.2), we get the following contradiction

$$\begin{aligned} K(a,\{b\}) = \varepsilon_a(\{b\}) = 0 \;&\Rightarrow\; \mu(\{b\}) = 0, \\ K(b,\{b\}) = \varepsilon_b(\{b\}) = 1 \;&\Rightarrow\; \mu(\{b\}) > 0. \end{aligned}$$

Example 4.3.4. Let $E = \mathbb{R}$ and

$$K(x,B) := \frac{1}{\sqrt{2\pi\sigma^2}} \int_B \exp\left(\frac{-(x-y)^2}{2\sigma^2}\right) dy.$$

Suppose there exists an $\varepsilon > 0$ and a measure μ such that the inequality (4.3.2) is satisfied. We note that for any $x \in \mathbb{R}$

$$K(x,B) = 0 \Leftrightarrow \mu(B) = 0,$$

where $B \in \mathcal{B}(\mathbb{R})$. If, in particular, $B = [0,1]$, then $K(x,B) > 0$ yields $\mu(B) > 0$. Hence, there exists $\varepsilon_* > 0$ such that for all $x \in \mathbb{R}$

$$\varepsilon_* := \varepsilon\,\mu(B) \le K(x,B).$$

By putting $x < -\sqrt{-\ln(2\pi\sigma^2\varepsilon_*^2)}\sigma$, however, we obtain

$$K(x,B) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_0^1 \exp\left(\frac{-(x-y)^2}{2\sigma^2}\right) dy \le \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-x^2}{2\sigma^2}\right) < \varepsilon_*.$$

This example demonstrates that if E is not bounded then the kernel is not mixing even in the Gaussian case.

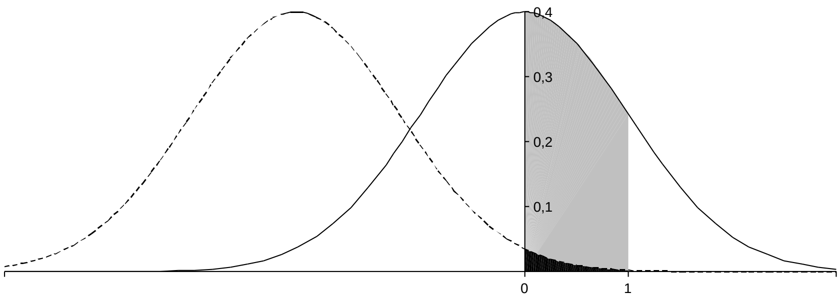


Figure 4.2: A Gaussian kernel $K(x,B)$ satisfies the mixing condition only if the set is bounded. When the set is $\mathbb{R}$ and thus unbounded, one simply moves x, which is the mean, to $-\infty$ such that the integral over $B = [0,1]$ gets smaller than any given $\varepsilon_* > 0$.

Le Gland and Oudjane [GlOu04] showed the uniform convergence of the generic particle filter (Theorem 4.3.5) by using the mixing condition as follows. Let us consider for all $\omega \in \Omega$ the family of random kernels $(R_t)_{t\in\mathbb{N}_0}$ defined by

$$R_t(x,B)(\omega) := \int_B g_{t+1}(Y_{t+1}(\omega) - h_{t+1}(y))\, K_t(x,dy),$$

for all $x \in E \subset \mathbb{R}^d$, $B \in \mathcal{E}$ and $t \in \mathbb{N}_0$.

Theorem 4.3.5. *If the family of random kernels $(R_t)_{t\in\mathbb{N}_0}$ is mixing with $\varepsilon_t \geq \varepsilon > 0$, then there exists a constant $c(\varepsilon)$ independent of n such that*

$$E\left[\left(\langle \eta_t^n, \varphi\rangle - \langle \eta_t, \varphi\rangle\right)^2\right] \leq c(\varepsilon)\frac{\|\varphi\|_\infty^2}{n},$$

for all $t \in \mathbb{N}_0$ and $\varphi \in B(\mathbb{R}^d)$.

Remark 4.3.6. The mixing condition (4.3.2) can be relaxed such that the density $dK(x,\cdot)/d\mu$ is not μ-almost surely greater than or equal to $\varepsilon > 0$ but may vanish on a part of the state space, as shown in [ChLi04].

5. Interacting Annealing Algorithm

We derive an interacting annealing algorithm that combines the idea of annealing with particle filtering. For this purpose, we introduce the Metropolis-Hastings algorithm that is relevant for annealed importance sampling, which is an algorithm that uses annealing for sampling. The latter forms the basis of the interacting annealing algorithm. Moreover, we draw up a mathematical model and prove the convergence of the two algorithms.

5.1 Introduction

Before we go into detail, we sketch the idea of annealing. As seen in the top left image of Figure 5.1, it may happen that the predicted particles differ significantly from the "true" state resulting in a poor estimate for the signal. This could be caused by a rare event in the context of the filtering problem or by a fast movement of the observed object in the context of tracking. In order to obtain a better estimate in this situation, the idea is to move the particles towards the global maximum of the weighting function. One approach is to repeat the procedure, that means diffusing the particles, weighting the particles and resampling, several times before the next time step, as illustrated in Figure 5.1. We refer to it as *repetition effect*. This may work well as long as there are no local maxima. However, as seen on the left hand side of Figure 5.2, the particles might get stuck in a local maximum. Then the repetition effect fails since the particles are misguided by the local maximum. For avoiding this misbehaviour, the particles are weighted by the weighting function only in a final step. Previously, the particles are weighted by smoothed versions of the original weighting function, where the influence of the local maxima is reduced first but increases gradually. This approach helps to overcome the problem with the local maxima, as demonstrated on the right hand side of Figure 5.2. We call it *annealing effect*. In the following sections, we discuss how to take advantage of this effect in the context of the filtering problem.

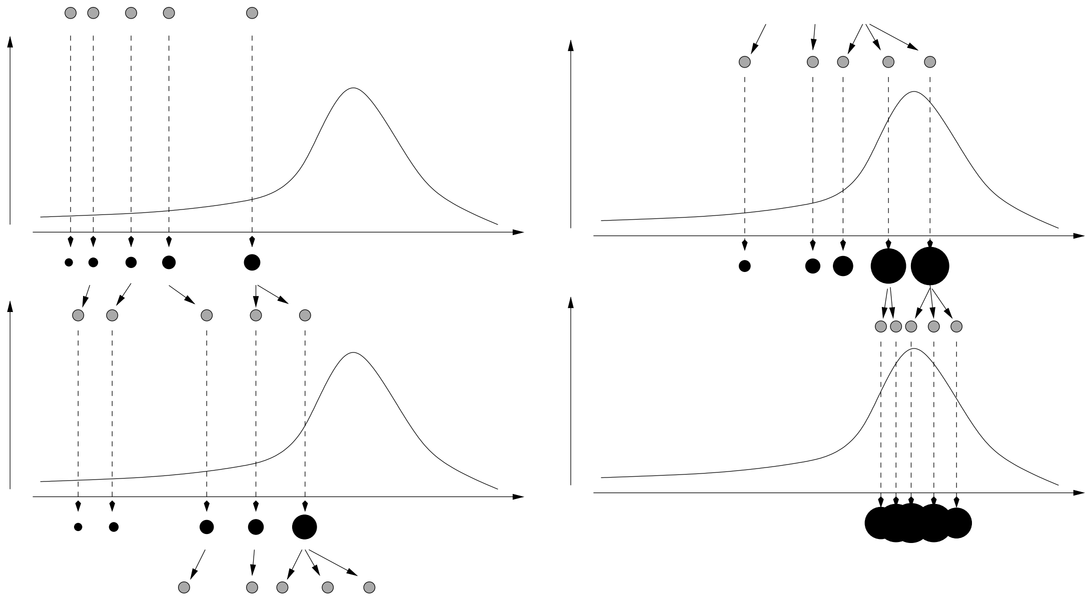


Figure 5.1: Repetition effect: the particles move to the maximum.

5.2 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a method for sampling from a *target distribution* $\nu(dx) = f(x)\lambda(dx)$ defined on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, where $\mathcal{X} \subseteq \mathbb{R}^k$. It was introduced by Metropolis [MRRT⁺53] and generalised by Hastings [Hast70]. The basic principle is to use a Markov process $(Y_k)_{k\in\mathbb{N}_0}$, such that the probability measure ν is invariant for the transitions. The initial distribution μ of the Markov process is usually the Dirac measure δ_x for a given $x \in \mathcal{X}$. The transitions are determined by the following procedure:

Algorithm 5.1 Metropolis-Hastings Algorithm

Requires: target distribution ν, initial distribution μ and proposal distribution T

1.
   - $k \leftarrow 0$
   - Sample y_0 from μ
2.
   - Sample y'_k from $T(y_k, \cdot)$
   - Take
   $$y_{k+1} = \begin{cases} y'_k & \text{with probability } p(y_k, y'_k) \\ y_k & \text{otherwise,} \end{cases}$$
   where
   $$p(x, y) = \min\left\{\frac{f(y)}{f(x)}\frac{g(y,x)}{g(x,y)}, 1\right\}.$$
   - $k \leftarrow k + 1$ and go to step 2

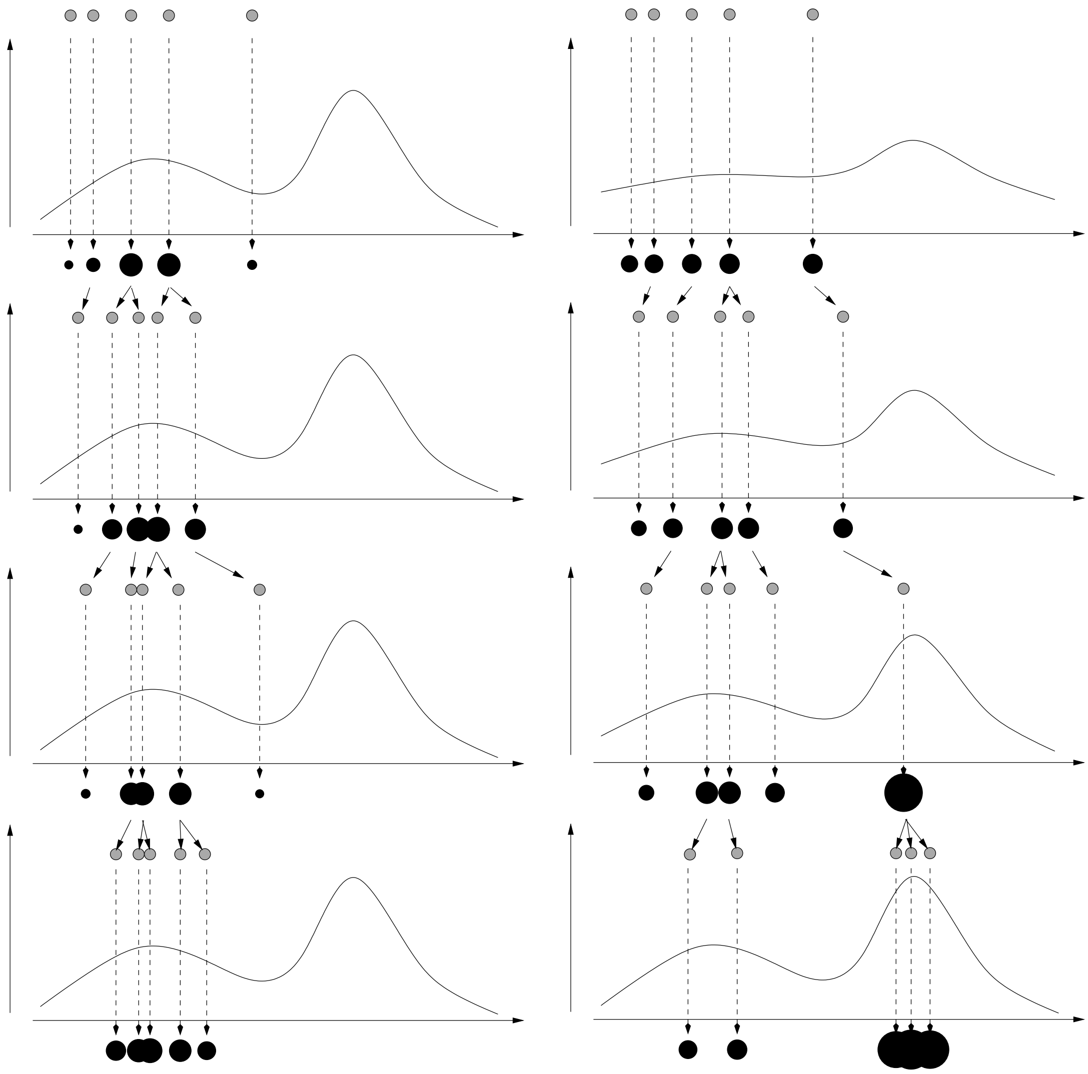


Figure 5.2: Without an annealing effect, the particles get stuck in the local maximum (left). In order that the particles escape from the local maximum, the annealing effect is used (right).

Suppose that $y_k \in \mathcal{X}$ is generated by the algorithm. Then y'_k is sampled from the distribution $T(y_k, \cdot)$, termed *proposal distribution*, where we assume that the distribution possesses a density, that is $T(x, dy) = g(x, y)\lambda(dy)$. The new value y'_k is always accepted if the ratio $f(y'_k)/g(y_k, y'_k)$ is greater than or equal to the previous value $f(y_k)/g(y'_k, y_k)$. If g is symmetric, the acceptance is controlled by the likelihood ratio $f(y'_k)/f(y_k)$. However, if the ratio decreases, y'_k is not automatically rejected. Instead, it is possible that the new value is accepted. An useful feature of the method is that if the process forgets its initial distribution μ, for a sufficient large k_*, Y_{k_*} can be considered as distributed from ν.

The algorithm generates a Markov process with initial distribution μ and transition kernel

$$K(x, dy) = p(x, y)\, T(x, dy) + (1 - r(x))\, \delta_x(dy), \tag{5.2.1}$$

where $r(x) = \int p(x, y)T(x, dy)$. Under the assumption that f and g are strictly positive, it can be shown that the process is reversible and f is invariant. If, in addition, f is bounded on every compact set and there exist $\varepsilon > 0$ and $\delta > 0$ such that $g(x, y) > \varepsilon$ for all x and y with $|x - y| < \delta$, then

$$\lim_{n\to\infty} \left\| \int K^n(x, \cdot)\, \mu(dx) - \nu \right\|_{TV} = 0.$$

For a more extensive discussion and proofs, see [RoCa02, Chapter 6] for instance.

Example 5.2.1. Let us suppose that the target distribution is a *Boltzmann-Gibbs measure*, which is the case in many applications. It is defined in terms of a common "energy" function $V \geq 0$ and an inverse "temperature" $\beta \geq 0$ by

$$\nu(dx) := \frac{1}{Z} \exp\left(-\beta V(x)\right)\, \lambda(dx),$$

where $Z := \langle \lambda, \exp(-\beta V) \rangle$. Whenever g is symmetric we find that

$$p(x, y) = \exp\left(-\beta\, (V(y) - V(x))^+\right),$$

where $(a)^+$ denotes $\max\{a, 0\}$.

5.3 Annealed Importance Sampling

The annealed importance sampling method was proposed by Neal [Neal98] and is motivated by simulated annealing ([KiJV83], [AaKo89]). A sequence of densities $(f_s)_{1\leq s\leq t}$ is used to interpolate between some initial distribution μ_0 with density f_0 and the target density f_{t+1} in order to generate weighted particles $(y^{(i)}_{t+1}, \pi^{(i)})$. It is assumed that $\langle \lambda, f_{t+1} \rangle < \infty$ and $\mathrm{supp}(f_s) \subset \mathrm{supp}(f_{s+1})$, i.e. $f_{s+1}(x) > 0$ whenever $f_s(x) > 0$. Furthermore, we denote by f^{norm} the normalised density, i.e.

$$f^{norm}(x) = \frac{1}{\langle \lambda, f \rangle} f(x).$$

In order to sample from the sequence of densities, transition kernels $T_s(y_s, \cdot)$ with densities g_s are used that leave f^{norm}_{s+1} invariant for $0 \leq s \leq t$. This can be achieved by Gibbs sampling [GeGe84], see [RoCa02, chapter 7] for instance, or by the Metropolis-Hastings algorithm with target density f_{s+1} and initial distribution δ_{y_s}.

Algorithm 5.2 Annealed Importance Sampling

Requires: number of particles n, number of runs t, initial distribution μ_0, densities $(f_s)_{1\leq s\leq t}$ and transitions $(T_s)_{0\leq s\leq t}$

1.
   - $s \leftarrow 0$
   - For $i = 1, \ldots, n$, set $\pi^{(i)} \leftarrow 1$
   - For $i = 1, \ldots, n$, sample $y_0^{(i)}$ from μ_0
2.
   - For $i = 1, \ldots, n$, set $\pi^{(i)} \leftarrow \frac{f_{s+1}(y_s^{(i)})}{f_s(y_s^{(i)})}\pi^{(i)}$
   - For $i = 1, \ldots, n$, sample $y_{s+1}^{(i)}$ from $T_s(y_s^{(i)}, \cdot)$
   - Until $s < t$: $s \leftarrow s + 1$ and go to step 2
3.
   - For $i = 1, \ldots, n$, set $\pi^{(i)} \leftarrow \frac{\pi^{(i)}}{\sum_{j=1}^{n}\pi^{(j)}}$

The sequence $(f_s)_{1\leq s\leq t}$ is defined as

$$f_s(x) = f_{t+1}(x)^{\beta_s} f_0(x)^{1-\beta_s}$$

according to some appropriate schedule $0 < \beta_1 < \beta_2 < \ldots < \beta_t < 1$. We note that $\langle \lambda, f_s\rangle < \infty$ for all s. Furthermore, we will use the notation $d(x_1, \ldots, x_t)$ to denote an infinitesimal neighbourhood of $(x_1, \ldots, x_t) \in \mathcal{X} \times \ldots \times \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}^d$ is the state space.

To validate the sampling scheme, we extend the state space to $\prod_{s=0}^{t+1}\mathcal{X}$ and define on $\bigotimes_{s=0}^{t+1}\mathcal{B}(\mathcal{X})$ the measure

$$P_{f_{t+1}}(d(y_0, \ldots, y_{t+1})) := \psi(y_0, \ldots, y_{t+1})\, \lambda^{t+2}(d(y_0, \ldots, y_{t+1})),$$

where $\lambda^{t+2} := \bigotimes_{s=0}^{t+1}\lambda$,

$$\psi(y_0, \ldots, y_{t+1}) := g_0'(y_1, y_0)\, g_1'(y_2, y_1) \ldots g_t'(y_{t+1}, y_t)\, f_{t+1}^{norm}(y_{t+1})$$

and $T_s'(x, dy) := g_s'(x, y)\lambda(dy)$ is the reversal of the transition kernel T_s. That is

$$g_s'(x, y) = \frac{g_s(y, x)\, f_{s+1}^{norm}(y)}{f_{s+1}^{norm}(x)} = \frac{g_s(y, x)\, f_{s+1}(y)}{f_{s+1}(x)}. \tag{5.3.1}$$

The invariance of f_{s+1}^{norm} with respect to T_s ensures that

$$\int T_s'(x, dy) = \int \frac{g_s(y, x)\, f_{s+1}^{norm}(y)}{f_{s+1}^{norm}(x)}\, \lambda(dy) = 1.$$

Thus we have that $P_{f_{t+1}}$ is a probability measure.

The weighted particle set defined by the algorithm determines a sequence of random probability measures by

$$p_{t+1}^n(\omega) := \sum_{i=1}^{n} \Pi^{(i)}(\omega)\delta_{Y_{t+1}^{(i)}(\omega)}$$

that converges to the probability measure $p_{t+1}(dx) := f_{t+1}^{norm}(x)\lambda(dx)$ as the following lemma shows.

Lemma 5.3.1.
$$P\left(\omega : p_{t+1}^n(\omega) \xrightarrow{w} p_{t+1}\right) = 1.$$

Proof: The joint distribution P_{f_0} of $(Y_0, \ldots, Y_{t+1})$, which is defined by the annealed importance sampling procedure, possesses a density

$$\phi(y_0, \ldots, y_{t+1}) = g_t(y_t, y_{t+1})\, g_{t-1}(y_{t-1}, y_t) \ldots g_0(y_0, y_1)\, f_0(y_0)$$

with respect to λ^{t+2}. Using (5.3.1) on the other hand, we obtain

$$\psi(y_0, \ldots, y_{t+1}) = \frac{f_{t+1}^{norm}(y_{t+1})}{f_{t+1}(y_{t+1})}\, g_t(y_t, y_{t+1}) \ldots \frac{f_2(y_1)}{f_1(y_1)}\, g_0(y_0, y_1)\, f_1(y_0).$$

The unnormalised weight $\widetilde{\pi}$ can now be expressed as

$$\begin{aligned} \widetilde{\pi}(y_0, \ldots, y_{t+1}) &= \prod_{s=0}^{t} \frac{f_{s+1}(y_s)}{f_s(y_s)} = \langle \lambda, f_{t+1} \rangle \frac{f_{t+1}^{norm}(y_{t+1})}{f_{t+1}(y_{t+1})} \prod_{s=0}^{t} \frac{f_{s+1}(y_s)}{f_s(y_s)} \\ &= \langle \lambda, f_{t+1} \rangle \frac{\psi(y_0, \ldots, y_{t+1})}{\phi(y_0, \ldots, y_{t+1})}. \end{aligned} \tag{5.3.2}$$

Let $\varphi \in C_b(\mathcal{X})$ and $\bar{\varphi} \in C_b(\prod_{s=0}^{t+1} \mathcal{X})$ defined by $\bar{\varphi}(y_0, \ldots, y_{t+1}) := \varphi(y_{t+1})$. Since the random variables $\bar{Y}^{(i)} := (Y_0^{(i)}, \ldots, Y_{t+1}^{(i)})$ are i.i.d. by definition, it follows from the strong law of large numbers [Baue91, Theorem 12.1] that

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \bar{\varphi}\left(\bar{Y}^{(i)}\right) \frac{\psi\left(\bar{Y}^{(i)}\right)}{\phi\left(\bar{Y}^{(i)}\right)} = \int \frac{\bar{\varphi}(\bar{y})\psi(\bar{y})}{\phi(\bar{y})}\, P_{f_0}(d\bar{y}) \qquad \text{a.s.}$$

and

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \frac{\psi\left(\bar{Y}^{(i)}\right)}{\phi\left(\bar{Y}^{(i)}\right)} = \int \frac{\psi(\bar{y})}{\phi(\bar{y})}\, P_{f_0}(d\bar{y}) = \int P_{f_{t+1}}(d\bar{y}) = 1 \qquad \text{a.s.}$$

Using the two limit values, we obtain

$$\begin{aligned} \langle p_{t+1}^n, \varphi \rangle &= \sum_{i=1}^{n} \Pi^{(i)} \varphi\left(Y_{t+1}^{(i)}\right) = \frac{\frac{1}{n}\sum_{i=1}^{n} \frac{1}{\langle \lambda, f_{t+1}\rangle} \widetilde{\pi}\left(\bar{Y}^{(i)}\right) \bar{\varphi}\left(\bar{Y}^{(i)}\right)}{\frac{1}{n}\sum_{i=1}^{n} \frac{1}{\langle \lambda, f_{t+1}\rangle} \widetilde{\pi}\left(\bar{Y}^{(i)}\right)} \\ &= \frac{\frac{1}{n}\sum_{i=1}^{n} \bar{\varphi}\left(\bar{Y}^{(i)}\right) \frac{\psi\left(\bar{Y}^{(i)}\right)}{\phi\left(\bar{Y}^{(i)}\right)}}{\frac{1}{n}\sum_{i=1}^{n} \frac{\psi\left(\bar{Y}^{(i)}\right)}{\phi\left(\bar{Y}^{(i)}\right)}} \\ &\xrightarrow{n\to\infty} \int\int \ldots \int \varphi(y_{t+1})\, T_0'(y_1, dy_0) \ldots T_t'(y_{t+1}, dy_t)\, p_{t+1}(dy_{t+1}) \\ &= \int \varphi(y_{t+1})\, p_{t+1}(dy_{t+1}) \end{aligned}$$

almost surely. Hence, the lemma follows from Remark 3.2.2. □

Remark 5.3.2. The original algorithm introduced by Neal [Neal98] does not sample y_{t+1} from $T_t(y_t, \cdot)$ in the last iteration and uses $p_{t+1}^n(\omega) := \sum_{i=1}^n \Pi^{(i)}(\omega)\delta_{Y_t^{(i)}(\omega)}$ as approximation instead. Indeed, one would omit the last sampling procedure in practice. Moreover, Lemma 5.3.1 would still hold. This can be proved along the lines of the proof above. In this case the probability measure $P_{f_{t+1}}$ on $\bigotimes_{s=0}^{t} \mathcal{B}(\mathcal{X})$ is defined by the density

$$\psi(y_0, \ldots, y_t) := g'_0(y_1, y_0) \ldots g'_{t-1}(y_t, y_{t-1})\, f_{t+1}^{norm}(y_t),$$

and ϕ becomes

$$\phi(y_0, \ldots, y_t) = g_{t-1}(y_{t-1}, y_t) \ldots g_0(y_0, y_1)\, f_0(y_0).$$

Hence, the unnormalised weight satisfies the equation (5.3.2) since

$$\langle \lambda, f_{t+1} \rangle \frac{\psi(y_0, \ldots, y_t)}{\phi(y_0, \ldots, y_t)} = \langle \lambda, f_{t+1} \rangle \frac{f_{t+1}^{norm}(y_t)}{f_t(y_t)} \prod_{s=0}^{t-1} \frac{f_{s+1}(y_s)}{f_s(y_s)} = \widetilde{\pi}(y_0, \ldots, y_t).$$

Example 5.3.3. As in Example 5.2.1, we consider a Boltzmann-Gibbs measure, namely

$$\nu(dx) := \frac{1}{\langle \lambda, g \rangle} g(x)\, \lambda(dx),$$

where $g(x) := \exp(-V(x))$. Let μ_0 be the initial distribution with density h. We set the target density $f_{t+1}(x) := h(x)g(x)$. Then we get for a given schedule $0 = \beta_0 < \beta_1 < \beta_2 < \ldots < \beta_t < \beta_{t+1} = 1$

$$f_s(x) = h(x)g(x)^{\beta_s},$$

for $0 \leq s \leq t+1$. Hence, the unnormalised weight reduces to

$$\widetilde{\pi} = \prod_{s=0}^{t} \exp\left(-(\beta_{s+1} - \beta_s)V(y_s)\right).$$

There can be problems of degeneracy in the particle filter when the conditional distribution η_t (3.1.1) does not overlap significantly with the predicted conditional probability measure $\hat{\eta}_t$ (3.1.2) in the context of the filtering problem. One way to overcome the deficiency is to incorporate annealed importance sampling (Algorithm 5.2) into the generic particle filter (Algorithm 4.1). However as in [DoMo02], we found in our simulations that this is not efficient, which is not astonishing. During the t annealing steps the information of the n particles is not shared and the benefit of the algorithm is devoured by the additional costs. In [DoFG01, chapter 7], the authors suggest to combine the annealed importance sampling with resampling.

5.4 Interacting Metropolis Model

In the following sections of this chapter, we derive an interacting annealing algorithm that combines the idea of annealing, as used for annealed importance sampling, with particle filtering. The mathematical framework of the algorithm is described by an interacting Metropolis model, which is a special Feynman-Kac model ([Mora04], [MoDP04], [MoMi00]). This gives us the possibility to prove the convergence of the algorithm and to estimate the rate of convergence.

5.4.1 Feynman-Kac Model

Let $(X_t)_{t\in\mathbb{N}_0}$ be an E-valued Markov process with family of transition kernels $(K_t)_{t\in\mathbb{N}_0}$ and initial distribution η_0. We denote by P_{η_0} the distribution of the Markov process, i.e., for $t \in \mathbb{N}_0$,

$$P_{\eta_0}\left(d(x_0, x_1, \ldots, x_t)\right) = K_{t-1}(x_{t-1}, dx_t)\, K_{t-2}(x_{t-2}, dx_{t-1}) \ldots K_0(x_0, dx_1)\, \eta_0(dx_0),$$

and by $E\left[\cdot\right]_{\eta_0}$ the expectation with respect to P_{η_0}. Moreover, let $(g_t)_{t\in\mathbb{N}_0}$ be a family of nonnegative, bounded $\mathcal{E}$-measurable functions such that

$$E\left[\prod_{s=0}^{t} g_s\left(X_s\right)\right]_{\eta_0} > 0,$$

for any $t \in \mathbb{N}_0$. We now present the Feynman-Kac model associated with the sequence of pairs (g_t, K_t).

Definition 5.1. The sequence of distributions $(\eta_t)_{t\in\mathbb{N}_0}$ on E defined for any $\varphi \in B(E)$ as

$$\langle \eta_t, \varphi \rangle := \frac{\langle \gamma_t, \varphi \rangle}{\langle \gamma_t, 1 \rangle},$$

where

$$\langle \gamma_t, \varphi \rangle := E\left[\varphi\left(X_t\right) \prod_{s=0}^{t-1} g_s\left(X_s\right)\right]_{\eta_0}, \tag{5.4.1}$$

is called the *Feynman-Kac model* associated with the pair (g_t, K_t).

Example 5.4.1. The functions $(g_t)_{t\in\mathbb{N}_0}$ are often unnormalised Boltzmann-Gibbs measures

$$g_t(x) = \exp\left(-\beta_t\, V_t(x)\right).$$

Equation (5.4.1) then becomes

$$\langle \gamma_t, \varphi \rangle := E\left[\varphi\left(X_t\right) \exp\left(-\sum_{s=0}^{t-1} \beta_s\, V\left(X_s\right)\right)\right]_{\eta_0}.$$

We now show that the Feynman-Kac model, as defined above, satisfies the recursive equation

$$\eta_{t+1} = \langle \Psi_t(\eta_t), K_t \rangle, \tag{5.4.2}$$

where the *Boltzmann-Gibbs transformation* Ψ_t is defined by

$$\Psi_t\left(\eta_t\right)\left(dx_t\right) := \frac{1}{\langle \eta_t, g_t \rangle}\, g_t(x_t)\, \eta_t(dx_t). \tag{5.4.3}$$

Let $\varphi \in B(E)$ and $\mathcal{F}_t = \sigma(X_t, \ldots, X_0)$. Then we get by the Markov property (2.3.5)

$$\begin{aligned}
\langle \gamma_{t+1}, \varphi \rangle &= E\left[E\left[\varphi\left(X_{t+1}\right) | \mathcal{F}_t\right]_{\eta_0} \prod_{s=0}^{t} g_s\left(X_s\right)\right]_{\eta_0} \\
&= E\left[\langle K_t, \varphi \rangle \left(X_t\right) \prod_{s=0}^{t} g_s\left(X_s\right)\right]_{\eta_0} \\
&= E\left[\left(\langle K_t, \varphi \rangle\, g_t\right)\left(X_t\right) \prod_{s=0}^{t-1} g_s\left(X_s\right)\right]_{\eta_0} \\
&= \langle \gamma_t, \langle K_t, \varphi \rangle\, g_t \rangle.
\end{aligned}$$

Therefore $\langle \gamma_{t+1}, 1\rangle = \langle \gamma_t, g_t\rangle$, and we have

$$\begin{aligned}
\langle \eta_{t+1}, \varphi\rangle &= \frac{\langle \gamma_{t+1}, \varphi\rangle}{\langle \gamma_{t+1}, 1\rangle} = \frac{\langle \gamma_t, \langle K_t, \varphi\rangle\, g_t\rangle / \langle \gamma_t, 1\rangle}{\langle \gamma_t, g_t\rangle / \langle \gamma_t, 1\rangle} = \frac{\langle \eta_t, \langle K_t, \varphi\rangle\, g_t\rangle}{\langle \eta_t, g_t\rangle} \\
&= \int_E \langle K_t, \varphi\rangle(x_t)\, \frac{g_t(x_t)}{\langle \eta_t, g_t\rangle}\, \eta_t(dx_t) = \langle \Psi_t(\eta_t), \langle K_t, \varphi\rangle\rangle \\
&= \langle\langle \Psi_t(\eta_t), K_t\rangle, \varphi\rangle.
\end{aligned}$$

The particle approximation of the flow (5.4.2) depends on a chosen family of Markov transition kernels $(K_{t,\eta_t})_{t\in\mathbb{N}_0}$ satisfying the compatibility condition

$$\langle \Psi_t\left(\eta_t\right), K_t\rangle := \langle \eta_t, K_{t,\eta_t}\rangle.$$

These families are not unique, we can choose, as in [Mora04, Chapter 2.5.3],

$$K_{t,\eta_t} = S_{t,\eta_t} K_t, \tag{5.4.4}$$

where

$$S_{t,\eta_t}(x_t, dy_t) = \epsilon_t\, g_t(x_t)\, \delta_{x_t}(dy_t) + (1 - \epsilon_t\, g_t(x_t))\;\Psi_t\left(\eta_t\right)(dy_t), \tag{5.4.5}$$

with $\epsilon_t \geq 0$ and $\epsilon_t \left\|g_t\right\|_\infty \leq 1$. It is interesting to remark that the parameters ϵ_t are allowed to depend on the current distribution η_t.

Example 5.4.2. Let us continue Example 5.4.1. Then the selection kernel becomes

$$S_{t,\eta_t}(x_t, dy_t) = \epsilon_t\, \exp\left(-\beta_t\, V_t(x_t)\right)\, \delta_{x_t}(dy_t) + \left(1 - \epsilon_t\, \exp\left(-\beta_t\, V_t(x_t)\right)\right)\; \Psi_t\left(\eta_t\right)(dy_t),$$

where

$$\Psi_t\left(\eta_t\right)(dy_t) = \frac{E\left[\exp\left(-\sum_{s=0}^{t-1} \beta_s\, V(X_s)\right)\right]_{\eta_0}}{E\left[\exp\left(-\sum_{s=0}^{t} \beta_s\, V(X_s)\right)\right]_{\eta_0}}\, \exp\left(-\beta_t\, V_t(y_t)\right)\, \eta_t(dy_t).$$

Next, we describe the approximation by a particle set using equation (5.4.4). The particle system is initialised by n i.i.d. random variables $X_0^{(i)}$ with common law η_0 determining the random probability measure

$$\eta_0^n(\omega) := \frac{1}{n}\sum_{i=1}^n \delta_{X_0^{(i)}(\omega)}.$$

Since K_{t,η_t} can be regarded as the composition of a pair of selection and mutation Markov kernels, we split the transitions into the following two steps

$$\eta_t^n \xrightarrow{\;Selection\;} \check{\eta}_t^n \xrightarrow{\;Mutation\;} \eta_{t+1}^n, \tag{5.4.6}$$

where

$$\begin{aligned}
\eta_t^n(\omega) &:= \frac{1}{n}\sum_{i=1}^n \delta_{X_t^{(i)}(\omega)}, \\
\check{\eta}_t^n(\omega) &:= \frac{1}{n}\sum_{i=1}^n \delta_{\check{X}_t^{(i)}(\omega)}.
\end{aligned}$$

During the selection step each particle $X_t^{(i)}$ evolves according to the Markov transition kernel $S_{t,\eta_t^n}(X_t^{(i)},\cdot)$. That means $X_t^{(i)}$ is accepted with probability $\epsilon_t g_t(X_t^{(i)})$, and we set $\check{X}_t^{(i)} = X_t^{(i)}$. Otherwise, $\check{X}_t^{(i)}$ is randomly selected with distribution

$$\sum_{i=1}^{n} \frac{g_t(X_t^{(i)})}{\sum_{j=1}^{n} g_t(X_t^{(j)})} \delta_{X_t^{(i)}}.$$

The mutation step consists in evolving each selected particle $\check{X}_t^{(i)}$ according to the Markov transition kernel $K_t(\check{X}_t^{(i)},\cdot)$.

5.4.2 Interacting Metropolis Model

In this section, we establish an algorithm based on annealed importance sampling (Algorithm 5.2) that allows additional interaction of the particles during the steps. We use a Feynman-Kac model as introduced in Section 5.4.1 to describe the mathematical framework of the algorithm. To go about doing it, we enlarge the state space $E = (\mathcal{X} \times \mathcal{X})$. In addition, we associate with a Markov kernel K on $\mathcal{X}$ the Markov kernel $\bar{K}$ on E defined by

$$\bar{K}((x,x'), B \times B') := \int_B K(y, B')\, \delta_{x'}(dy),$$

for $(B \times B') \in \mathcal{E} = \mathcal{B}(\mathcal{X}) \otimes \mathcal{B}(\mathcal{X})$. That means if $(Y_t)_{t\in\mathbb{N}_0}$ is a $\mathcal{X}$-valued Markov process with the family of transition kernels $(K_t)_{t\in\mathbb{N}_0}$, then $(\bar{K}_t)_{t\in\mathbb{N}}$ is the family of transition kernels of the Markov process defined on E by

$$X_t = (Y_t, Y_{t+1}) \in E.$$

Note that $\bar{K}_t$ are the transitions from X_{t-1} to X_t. Finally, we introduce the following notations for any $\mu \in \mathcal{P}(\mathcal{X})$ and any Markov kernel K on $\mathcal{X}$:

$$\begin{aligned}(\mu \times K)_1(B \times B') &= \int_B K(y, B')\, \mu(dy), \\ (\mu \times K)_2(B \times B') &= \int_{B'} K(y', B)\, \mu(dy'),\end{aligned}$$

for all $(B \times B') \in \mathcal{E}$.

We assume that $(T_t)_{t\in\mathbb{N}_0}$ and $(T'_t)_{t\in N_0}$ are families of Markov kernels on $\mathcal{X}$ and $\mu_t \in \mathcal{P}(\mathcal{X})$, for all $t \in \mathbb{N}_0$, such that $(\mu_{t+1} \times T'_t)_2$ is absolutely continuous with respect to $(\mu_t \times T_t)_1$, for all $t \in \mathbb{N}_0$, and the Radon-Nikodym derivatives

$$g_t(y_t, y_{t+1}) := \frac{d(\mu_{t+1} \times T'_t)_2}{d(\mu_t \times T_t)_1}(y_t, y_{t+1}) \tag{5.4.7}$$

are strictly positive and bounded functions on E. The Feynman-Kac model associated with the pair $(g_t, \bar{T}_{t+1})$ is called the *Feynman-Kac-Metropolis model*. The distributions P_{μ_0} and P_{μ_t} are defined as above by

$$\begin{aligned}P_{\mu_0}(d(y_0, y_1, \ldots, y_t)) &= T_{t-1}(y_{t-1}, dy_t)\, T_{t-2}(y_{t-2}, dy_{t-1}) \ldots T_0(y_0, dy_1)\, \mu_0(dy_0), \\ P_{\mu_t}(d(y_0, y_1, \ldots, y_t)) &= T'_0(y_1, dy_0)\, T'_1(y_2, dy_1) \ldots T'_{t-1}(y_t, dy_{t-1})\, \mu_t(dy_t).\end{aligned}$$

Using these notations, we show the following key lemma.

Lemma 5.4.3 (reversal formula). *For any $t \in \mathbb{N}_0$ and any $\varphi \in B(\mathcal{X}^{t+1})$, we have*

$$E\left[\varphi(Y_0, Y_1, \dots, Y_t)\right]_{\mu_t} = E\left[\varphi(Y_0, Y_1, \dots, Y_t) \prod_{s=0}^{t-1} g_s(Y_s, Y_{s+1})\right]_{\mu_0}.$$

Proof: Let $\varphi \in B(\mathcal{X}^{t+1})$ and set $\bar{y}_t = (y_0, y_1, \dots, y_t)$. Since the case $t = 0$ is trivial, we assume that $t > 0$. Then we have

$$\begin{aligned}
&E\left[\varphi(Y_0, Y_1, \dots, Y_t) \prod_{s=0}^{t-1} g_s(Y_s, Y_{s+1})\right]_{\mu_0} \\
&= \int_{\mathcal{X}} \int_{\mathcal{X}} \dots \int_{\mathcal{X}} \varphi(\bar{y}_t) \left(\prod_{s=0}^{t-1} g_s(y_s, y_{s+1})\, T_s(y_s, dy_{s+1})\right) \mu_0(dy_0) \\
&= \int_{E} \dots \int_{\mathcal{X}} \varphi(\bar{y}_t) \left(\prod_{s=1}^{t-1} g_s(y_s, y_{s+1})\, T_s(y_s, dy_{s+1})\right) \dots \\
&\qquad\qquad \dots \frac{d(\mu_1 \times T_0')_2}{d(\mu_0 \times T_0)_1}(y_0, y_1)\, (\mu_0 \times T_0)_1(d(y_0, y_1)) \\
&= \int_{E} \dots \int_{\mathcal{X}} \varphi(\bar{y}_t) \left(\prod_{s=1}^{t-1} g_s(y_s, y_{s+1})\, T_s(y_s, dy_{s+1})\right) (\mu_1 \times T_0')_2(d(y_0, y_1)) \\
&= \int_{\mathcal{X}} \dots \int_{\mathcal{X}} \int_{\mathcal{X}} \varphi(\bar{y}_t)\, T_0'(y_1, dy_0) \left(\prod_{s=1}^{t-1} g_s(y_s, y_{s+1})\, T_s(y_s, dy_{s+1})\right) \mu_1(dy_1) \\
&\ \vdots \\
&= \int_{\mathcal{X}} \int_{\mathcal{X}} \dots \int_{\mathcal{X}} \varphi(\bar{y}_t)\, T_0'(y_1, dy_0) \dots T_{t-1}'(y_t, dy_{t-1})\, \mu_t(dy_t).
\end{aligned}$$

□

Example 5.4.4. As in Examples 5.4.2 and 5.4.1, we consider the unnormalised Boltzmann-Gibbs measures

$$g_s(y_s) = \exp\left(-\beta_s V(y_s)\right),$$

for $0 \le s < t$ and a given $t \in \mathbb{N}_0$. Suppose there exist $(T_s)_{0 \le s < t}$, $(T_t')_{0 \le s < t}$, μ_0 and μ_t such that the functions g_s are the Radon-Nikodym derivatives in accordance with (5.4.7). Note that g_s depends only on the value y_s and not on y_{s+1}, for all s. Let $\varphi \in B(\mathcal{X})$. Then we define the extension $\bar{\varphi}(y_0, y_1, \dots, y_t) := \varphi(y_t)$ for all $(y_0, y_1, \dots, y_t) \in \mathcal{X}^{t+1}$, which is a bounded $\mathcal{B}(\mathcal{X}^{t+1})$-measurable function. Hence, the reversal formula gives

$$E\left[\varphi(Y_t) \exp\left(-\sum_{s=0}^{t-1} \beta_s V(Y_s)\right)\right]_{\mu_0} = E\left[\varphi(Y_t)\right]_{\mu_t} = \int_{\mathcal{X}} \varphi(y_t)\, \mu_t(dy_t).$$

In the following, we do not restrict ourself to Boltzann-Gibbs measures as in the example above. By putting $X_t := (Y_t, Y_{t+1})$, we get from the reversal formula

$$E\left[\varphi(X_t) \prod_{s=0}^{t-1} g_s(X_s)\right]_{(\mu_0 \times T_0)_1} = E\left[\varphi(X_t)\right]_{(\mu_t \times T_t)_1},$$

for any $\varphi \in B(E)$ and any $t \in \mathbb{N}_0$. Since

$$\int_E \int_{\mathcal{X}} \dots \int_{\mathcal{X}} T'_0(y_1, dy_0) \dots T'_{t-1}(y_t, dy_{t-1})\, \varphi(y_t, y_{t+1})\, (\mu_t \times T_t)_1(d(y_t, y_{t+1}))$$
$$= \int_E \varphi(x_t)\, (\mu_t \times T_t)_1(dx_t),$$

we are able to state the following corollary, which will be useful for designing an algorithm.

Corollary 5.4.5. *For any $t \in \mathbb{N}_0$ and any $\varphi \in B(E)$, we have*

$$E\left[\varphi(X_t) \prod_{s=0}^{t-1} g_s(X_s)\right]_{(\mu_0 \times T_0)_1} = \langle (\mu_t \times T_t)_1, \varphi \rangle.$$

Furthermore, the Feynman-Kac model associated with the pair $(g_t, \bar{T}_{t+1})$ is described by

$$\eta_t = (\mu_t \times T_t)_1.$$

5.4.3 Interacting Annealing Algorithm

In this section, we develop a Feynman-Kac-Metropolis model for Example 5.3.3. This model is used afterwards to design an algorithm that can be regarded as an interacting annealed importance sampling scheme.

Let us consider as above a sequence of Boltzmann-Gibbs measures

$$\nu_s(dx) := \frac{1}{\langle \lambda, \exp(-\beta_s V) \rangle} \exp\left(-\beta_s V(x)\right)\, \lambda(dx)$$

according to some schedule $0 = \beta_0 < \beta_1 < \beta_2 < \ldots < \beta_t < \beta_{t+1} = 1$. We assume that $\mu_0 \in \mathcal{P}(\mathcal{X})$ is absolutely continuous with respect to λ and define the sequence of distributions $(\mu_s)_{0 \le s \le t+1}$ by

$$\mu_s(dx) := \frac{1}{\langle \mu_0, \exp(-\beta_s V) \rangle} \exp\left(-\beta_s V(x)\right)\, \mu_0(dx). \tag{5.4.8}$$

We choose a family of Markov kernels $(T_s)_{0 \le s \le t}$ such that $T_s(y_s, \cdot)$ is absolutely continuous with respect to μ_{s+1}, for all $y_s \in \mathcal{X}$ and $0 \le s \le t$, and such that T_s leaves the measure μ_{s+1} invariant, i.e.

$$\mu_{s+1}(B) = \int_{\mathcal{X}} T_s(y_s, B)\, \mu_{s+1}(dy_s), \tag{5.4.9}$$

for all $B \in \mathcal{B}(\mathcal{X})$. Then, we set

$$T'_s(y_{s+1}, B) := \int_B \frac{dT_s(y_s, \cdot)}{d\mu_{s+1}}(y_{s+1})\, \mu_{s+1}(dy_s),$$

for $B \in \mathcal{B}(\mathcal{X})$ and $0 \le s \le t$. It is easy to see by equation (5.4.9) that all T'_s are well defined Markov kernels. The Radon-Nykodym derivatives, defined in (5.4.7), are given by the following lemma.

Lemma 5.4.6. *For any* $0 \le s \le t$*, we have*

$$g_s(y_s, y_{s+1}) = \frac{\langle \mu_0, \exp(-\beta_s V)\rangle}{\langle \mu_0, \exp(-\beta_{s+1} V)\rangle} \exp\left(-(\beta_{s+1} - \beta_s) V(y_s)\right).$$

Proof: Let $0 \le s \le t$ and $(B_s, B_{s+1}) \in \mathcal{E}$. Then we obtain

$$\begin{aligned}
&\int_{B_s} \int_{B_{s+1}} g_s(y_s, y_{s+1})\, T_s(y_s, dy_{s+1})\, \mu_s(dy_s) \\
&= \int_{B_s} \int_{B_{s+1}} \frac{1}{\langle \mu_0, \exp(-\beta_{s+1} V)\rangle} \exp\left(-\beta_{s+1} V(y_s)\right)\, T_s(y_s, dy_{s+1})\, \mu_0(dy_s) \\
&= \int_{B_s} \int_{B_{s+1}} T_s(y_s, dy_{s+1})\, \mu_{s+1}(dy_s) \\
&= \int_{B_{s+1}} \int_{B_s} \frac{dT_s(y_s, \cdot)}{d\mu_{s+1}}(y_{s+1})\, \mu_{s+1}(dy_s)\, \mu_{s+1}(dy_{s+1}) \\
&= \int_{B_{s+1}} \int_{B_s} T_s'(y_{s+1}, dy_s)\, \mu_{s+1}(dy_{s+1}).
\end{aligned}$$

□

Metropolis-Hastings updates (5.2.1) are suitable choices for Markov transitions T_s that leave μ_{s+1} invariant. What this means in practice is that we use one cycle of the Metropolis-Hastings algorithm (Algorithm 5.1) to sample from T_s. For instance, we have

$$T_s(y_s, dy_{s+1}) = p_s(y_s, y_{s+1})\, M_s(y_s, dy_{s+1}) + (1 - r_s(y_s))\ \delta_{y_s}(dy_{s+1}), \qquad (5.4.10)$$

where

$$\begin{aligned}
r_s(y_s) &= \int_{\mathcal{X}} p_s(y_s, y_{s+1})\, M_s(y_s, dy_{s+1}), \\
p_s(y_s, y_{s+1}) &= \exp\left(-\beta_{s+1} \left(V(y_{s+1}) - V(y_s)\right)^+\right)
\end{aligned}$$

and M_s leaves μ_0 invariant, for all $0 \le s \le t$.

Example 5.4.7. Suppose M_s is a Markov kernel that is absolutely continuous with respect to μ_0 and the Radon-Nykodym derivative $dM_s(y_s, \cdot)/d\mu_0$ is symmetric. Then we have for any $B \in \mathcal{B}(\mathcal{X})$

$$\begin{aligned}
\int_{\mathcal{X}} M_s(y_s, B)\, \mu_0(dy_s) &= \int_{\mathcal{X}} \int_B \frac{dM_s(y_s, \cdot)}{d\mu_0}(y_{s+1})\, \mu_0(dy_{s+1})\, \mu_0(dy_s) \\
&= \int_B \int_{\mathcal{X}} \frac{dM_s(y_{s+1}, \cdot)}{d\mu_0}(y_s)\, \mu_0(dy_s)\, \mu_0(dy_{s+1}) \\
&= \mu_0(B).
\end{aligned}$$

In the case that μ_0 possesses a density with respect to λ, we can also modify the Markov kernels (5.4.10) by setting

$$p_s(y_s, y_{s+1}) = \min\left\{ \frac{\frac{d\mu_0}{d\lambda}(y_{s+1})}{\frac{d\mu_0}{d\lambda}(y_s)} \exp\left(-\beta_{s+1}\left(V(y_{s+1}) - V(y_s)\right)\right), 1 \right\}.$$

Then it is sufficient that M_s possesses a symmetric density with respect to λ, e.g. M_s is Gaussian.

By Corollary 5.4.5, we have that the sequence of distributions $((\mu_s \times T_s)_1)_{0\le s\le t+1}$ on $E = (\mathcal{X} \times \mathcal{X})$ is a Feynman-Kac model associated with the pair $(g_s, \bar{T}_{s+1})$. Hence, the particle approximation of the flow

$$(\mu_{s+1} \times T_{s+1})_1 = \langle \Psi_s((\mu_s \times T_s)_1), \bar{T}_{s+1}\rangle,$$

where Ψ_s is the Boltzmann-Gibbs transformation (5.4.3), can be processed according to the Markov kernel

$$K_{s,(\mu_s\times T_s)_1} = S_{s,(\mu_s\times T_s)_1}\bar{T}_{s+1},$$

where $S_{s,(\mu_s\times T_s)_1}$ is defined in (5.4.5). The interacting annealing algorithm (Algorithm 5.3) results from these observations.

Algorithm 5.3 Interacting Annealing Algorithm

Requires: parameters $(\epsilon_s)_{0\le s\le t}$, number of particles n, number of runs t, initial distribution μ_0, weighting functions $(g_s)_{0\le s\le t}$ and transitions $(T_s)_{0\le s\le t}$

1. Initialisation
   - $s \leftarrow 0$
   - For $i = 1, \dots, n$, sample $y_0^{(i)}$ from μ_0
2. Mutation
   - For $i = 1, \dots, n$, sample $\tilde{y}_{s+1}^{(i)}$ from $T_s(y_s^{(i)}, \cdot)$
3. Selection
   - For $i = 1, \dots, n$, set $\pi^{(i)} \leftarrow g_s(y_s^{(i)}, \tilde{y}_{s+1}^{(i)})$
   - Set $\bar{\pi} \leftarrow \sum_{j=1}^{n} \pi^{(j)}$
   - For i from 1 to n:

     Sample κ from $U[0,1]$
     If $\kappa \le \epsilon_s \pi^{(i)}$ then
       $\star$ Set $y_{s+1}^{(i)} \leftarrow \tilde{y}_{s+1}^{(i)}$
     Else
       $\star$ Set $y_{s+1}^{(i)} \leftarrow \tilde{y}_{s+1}^{(j)}$ with probability $\frac{\pi^{(j)}}{\bar{\pi}}$
   - Until $s < t$: $s \leftarrow s + 1$ and go to step 2

Before we discuss the convergence of the algorithm, we comment on the approximation model described by the algorithm. We will use the notations $X_s^{(i)} := (Z_s^{(i)}, Z_{s+1}^{(i)})$ and $\check{X}_s^{(i)} := (\check{Z}_s^{(i)}, \check{Z}_{s+1}^{(i)})$ to avoid any confusion. Initially, $X_0^{(i)} = (Y_0^{(i)}, \widetilde{Y}_1^{(i)})_{1\le i\le n}$ are i.i.d. random variables with common law $(\mu_0 \times T_0)_1$ determining the random probability measure

$$(\mu_0 \times T_0)_1{}^n(\omega) := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_0^{(i)}(\omega)}.$$

Since the first mutation step is part of the initialisation, the order of the selection step and mutation step is exactly the same as described in (5.4.6). At each run $0 \leq s \leq t$, $(\mu_s \times T_s)_1$ is approximated as usual by

$$(\mu_s \times T_s)_1{}^n(\omega) := \frac{1}{n} \sum_{i=1}^{n} \delta_{X_s^{(i)}(\omega)}. \tag{5.4.11}$$

During the selection step, each particle $X_s^{(i)} = (Z_s^{(i)}, Z_{s+1}^{(i)})$ is accepted with probability $\epsilon_s g_s(X_s^{(i)})$ and we set $\check{X}_s^{(i)} = X_s^{(i)}$. Otherwise, we have $\check{X}_s^{(i)} = (Z_s^{(i)}, \check{Z}_{s+1}^{(i)})$, where $\check{Z}_{s+1}^{(i)}$ is randomly selected with distribution

$$\sum_{i=1}^{n} \frac{g_s(X_s^{(i)})}{\sum_{j=1}^{n} g_s(X_s^{(j)})} \delta_{Z_{s+1}^{(i)}}.$$

By the mutation step, each selected particle $\check{X}_s^{(i)} = (\check{Z}_s^{(i)}, \check{Z}_{s+1}^{(i)})$ evolves into $X_{s+1}^{(i)} = (\check{Z}_{s+1}^{(i)}, Z_{s+2}^{(i)})$, where $Z_{s+2}^{(i)}$ is distributed according to the Markov transition kernel $T_{s+1}(\check{Z}_{s+1}^{(i)}, \cdot)$. Note that $\check{Z}_s^{(i)}$ is not used for mutating. Therefore, we do not have to care about it during the selection step. In the last run, the selection/mutation cycle is not executed completely, instead we perform just a selection. If we added an additional mutation step, we would obtain $X_{t+1} = (Z_{t+1}, Z_{t+2})$. Since we are usually only interested in Z_{t+1} and since $Z_{t+1} = \check{Z}_{t+1}$, $\check{X}_t = (\check{Z}_t, \check{Z}_{t+1})$ already contains the information needed.

The weighting functions g_s calculated in Lemma 5.4.6 have a constant term

$$c_s := \frac{\langle \mu_0, \exp(-\beta_s V) \rangle}{\langle \mu_0, \exp(-\beta_{s+1} V) \rangle}$$

that is difficult to evaluate in many applications. However, that is not necessary. By setting

$$\epsilon'_s = \epsilon_s \, c_s,$$

we use

$$g'_s(y_s) := \exp\left(-(\beta_{s+1} - \beta_s) V(y_s)\right)$$

instead. Note that $g_s(y_s, y_{s+1})$ does not depend on y_{s+1} and that a constant term does not play any role for the expression $\pi^{(j)}/\bar{\pi}$.

Finally, we remark that we get an annealed importance sampling algorithm with resampling (Algorithm 5.4) as special case of the interacting annealing algorithm (Algorithm 5.3) when we put $\epsilon_s = 0$ for all s.

5.4.4 Convergence of the Interacting Annealing Algorithm

This section investigates the asymptotic behaviour of the particle approximation model determined by the interacting annealing algorithm (Algorithm 5.3). We know from Section 5.4.3 that the sequence of distributions

$$((\mu_s \times T_s)_1)_{0 \leq s \leq t+1}$$

Algorithm 5.4 Annealed Importance Sampling with Resampling

Requires: number of particles n, number of runs t, initial distribution μ_0, weighting functions $(g_s)_{0\leq s\leq t}$ and transitions $(T_s)_{0\leq s\leq t}$

1. - $s \leftarrow 0$
   - For $i = 1,\ldots,n$, sample $y_0^{(i)}$ from μ_0
2. - For $i = 1,\ldots,n$, set $\pi^{(i)} \leftarrow \exp(-(\beta_{s+1}-\beta_s)V(y_s))$
   - For $i = 1,\ldots,n$, set $\pi^{(i)} \leftarrow \frac{\pi^{(i)}}{\sum_{j=1}^{n}\pi^{(j)}}$
   - For $i = 1,\ldots,n$, sample $\tilde{y}_{s+1}^{(i)}$ from $T_s(y_s^{(i)},\cdot)$
   - For $i = 1,\ldots,n$, set $y_{s+1}^{(i)} \leftarrow \tilde{y}_{s+1}^{(j)}$ with probability $\pi^{(j)}$
   - Until $s < t$: $s \leftarrow s+1$ and go to step 2

is a Feynman-Kac model associated with the pair $(g_s, \bar{T}_{s+1})$. Furthermore, we have established that this Feynman-Kac model is approximated by

$$(\mu_s \times T_s)_1{}^n(\omega) = \frac{1}{n}\sum_{i=1}^{n}\delta_{X_s^{(i)}(\omega)}$$

as defined in (5.4.11). Hence, we obtain the following convergence theorem from [Mora04, Theorem 7.4.4].

Theorem 5.4.8. *For any $\bar{\varphi} \in B(E)$,*

$$E\left[\left|\langle(\mu_{t+1}\times T_{t+1})_1{}^n,\bar{\varphi}\rangle - \langle(\mu_{t+1}\times T_{t+1})_1,\bar{\varphi}\rangle\right|\right] \leq \frac{2\,\mathrm{osc}(\bar{\varphi})}{\sqrt{n}}\sum_{s=0}^{t+1} r_s\beta(M_s),$$

where

$$\begin{aligned} r_s \quad &:= \quad \begin{cases} \sup_{x,y\in E}\left(\frac{\prod_{r=s}^{t} g_r(x)}{\prod_{r=s}^{t} g_r(y)}\right) & \text{for} \quad 0\leq s\leq t \\ 1 & \text{for} \quad s = t+1 \end{cases}, \\ \langle M_s,\bar{f}\rangle(x_s) \quad &:= \quad \int_E \ldots \int_E \bar{f}(x_{t+1})\,\bar{T}_{t+1}(x_t,dx_{t+1})\ldots\bar{T}_{s+1}(x_s,dx_{s+1}), \end{aligned}$$

for $0 \leq s \leq t$, and $\langle M_{t+1},\bar{f}\rangle := \bar{f}$ for all $x_s \in E$ and $\bar{f} \in B(E)$. Moreover, $\mathrm{osc}(\bar{\varphi}) := \sup\{|\bar{\varphi}(x)-\bar{\varphi}(y)|\ ;\ x,y\in E\}$ *and $\beta(M_s)$ is the Dobrushin contraction coefficient of M_s, cf. Definition 2.3.2.*

When we consider any $\varphi \in B(\mathcal{X})$ and set $\bar{\varphi}(y_{t+1},y_{t+2}) := \varphi(y_{t+1})$ for all $(y_{t+1},y_{t+2}) \in E$, we observe

$$\int_{\mathcal{X}}\int_{\mathcal{X}}\varphi(y_{t+1})\,T_{t+1}(y_{t+1},dy_{t+2})\,\mu_{t+1}(dy_{t+1}) = \int_{\mathcal{X}}\varphi(y_{t+1})\,\mu_{t+1}(dy_{t+1})$$

and

$$\langle(\mu_{t+1}\times T_{t+1})_1{}^n(\omega),\bar{\varphi}\rangle = \frac{1}{n}\sum_{i=1}^{n}\varphi(Y_{t+1}^{(i)}(\omega)).$$

Additionally, Lemma 5.4.6 yields

$$\frac{\prod_{r=s}^{t} g_r(x, x')}{\prod_{r=s}^{t} g_r(y, y')} = \exp\left(-(1-\beta_s)(V(x) - V(y))\right),$$

for all $(x, x'), (y, y') \in E$ and $0 \leq s \leq t$. Finally, we remark that the Markov kernel T_{t+1} is not used in the interacting annealing algorithm (Algorithm 5.3). Thus for any $y_{t+1} \in \mathcal{X}$ and $\bar{f} \in B(E)$, we set $T_{t+1}(y_{t+1}, \cdot) := \delta_{y_{t+1}}$ and $f(y_{t+1}) := \bar{f}(y_{t+1}, y_{t+1})$. Then we obtain

$$\langle M_s, \bar{f}\rangle(x_s) = \int_{\mathcal{X}} \dots \int_{\mathcal{X}} f(y_{t+1})\, T_t(y_t, dy_{t+1}) \dots T_{s+1}(y_{s+1}, dy_{s+2}),$$

for $0 \leq s < t$. In summary, we have the following corollary:

Corollary 5.4.9. *For any $\varphi \in B(\mathcal{X})$, we have*

$$E\left[\left|\langle \mu_{t+1}^n, \varphi\rangle - \langle \mu_{t+1}, \varphi\rangle\right|\right] \leq \frac{2\,\mathrm{osc}(\varphi)\exp(\mathrm{osc}(V))}{\sqrt{n}} \sum_{s=0}^{t+1} r_s \beta(M_s),$$

where

$$\begin{aligned} r_s &= \exp\left(-\beta_s\,\mathrm{osc}(V)\right), \\ \langle M_s, f\rangle(y_{s+1}) &= \int_{\mathcal{X}} \dots \int_{\mathcal{X}} f(y_{t+1})\, T_t(y_t, dy_{t+1}) \dots T_{s+1}(y_{s+1}, dy_{s+2}), \end{aligned}$$

for $0 \leq s < t$, and $\langle M_t, f\rangle = \langle M_{t+1}, f\rangle = f$ for all $y_{s+1} \in \mathcal{X}$ and $f \in B(\mathcal{X})$.

This corollary gives us a rough estimate for the number of particles

$$n \geq \frac{4\,\mathrm{osc}(\varphi)^2 \exp(2\,\mathrm{osc}(V))}{\delta^2} \left(\sum_{s=0}^{t+1} r_s \beta(M_s)\right)^2 \tag{5.4.12}$$

needed to achieve a mean error less than a given $\delta > 0$. For evaluating the right hand side, we must calculate the Dobrushin contraction coefficient of a Markov kernel K on $\mathcal{X}$. The coefficient lies in the range 0 to 1, and the more the probability measure $K(x, \cdot)$ "depends" on $x \in \mathcal{X}$, the higher the coefficient is. We will illustrate this property in the following four examples where we always assume that $\mathcal{X} = [0, 1]$.

Example 5.4.10. If $K(x, \cdot) := \delta_x$ and $x_1, x_2 \in \mathcal{X}$ with $x_1 \neq x_2$, then we get $\sup_{B \in \mathcal{B}(\mathcal{X})} |\delta_{x_1}(B) - \delta_{x_2}(B)| = 1$. This yields $\beta(K) = 1$.

Example 5.4.11. If $K(x, \cdot) := \lambda$, then we have $\beta(K) = \sup_{B \in \mathcal{X}} |\lambda(B) - \lambda(B)| = 0$.

Example 5.4.12. Suppose $K := T_{s+1}T_{s+2} \dots T_t$, where $(T_k)_{s<k\leq t}$ are Markov kernels and $s < t$. Furthermore, we assume that there exists for all $s < k \leq t$ some $\varepsilon_k \in (0, 1)$ satisfying for all $x_1, x_2 \in \mathcal{X}$

$$T_k(x_1, \cdot) \geq \varepsilon_k\, T_k(x_2, \cdot). \tag{5.4.13}$$

Let $x_1, x_2 \in \mathcal{X}$ and $B \in \mathcal{B}(\mathcal{X})$. Then we get $|T_k(x_1, B) - T_k(x_2, B)| \leq 1 - \varepsilon_k$. Hence, it follows from inequality (2.3.1) that $\beta(K) \leq \prod_{k=s+1}^{t}(1 - \varepsilon_k)$.

Example 5.4.13. Let μ_s be the Boltzmann-Gibbs measure defined in (5.4.8) and let K_s be a Markov kernel like (5.2.1) that leaves μ_s invariant with a proposal distribution T_s. If T_s satisfies condition (5.4.13) with some $\varepsilon_s \in (0, 1)$, then we have

$$\beta(K_s) \leq 1 - \varepsilon_s \exp\left(-\operatorname{osc}(V)\right),$$

as stated in [MoDo03].

Note that the right hand side of (5.4.12) is minimised if we are able to choose Markov kernels T_s such that $\beta(M_s)$ is very small. However, if we compare the examples, we see that this corresponds to the fact that we do not trust our particles. In practice, it would be preferable to select the Markov kernels by means of the "quality" of the particles in the previous step. One approach is to select kernels that depend on a set of parameters, for example Gaussian kernels with the entries of the covariance matrix as parameters. The values of the parameters are then determined automatically by the particles, for example the variance is set proportional to the sampling variance of the particles. This is realised by a dynamic variance scheme, which we will discuss in Section 7.1.4.

6. Generalised Annealed Particle Filter

When we combine the generic particle filter (Algorithm 4.1) discussed in Chapter 4 and the interacting annealing algorithm (Algorithm 5.3) developed in Chapter 5, we obtain the generalised annealed particle filter (Algorithm 6.1). For it, we have to make the crucial assumption that the density g_t defined in Chapter 3 can be written as

$$g_t(x) = \frac{1}{\langle \lambda, \exp(-V_t) \rangle} \exp\left(-V_t(x)\right),$$

where $V_t \geq 0$ for all $t \in \mathbb{N}$. Furthermore, we suppose that the state space is $\mathcal{X} \subset \mathbb{R}^d$ and that $K_t(x, \cdot)$ is absolutely continuous with respect to λ, for all $x \in \mathcal{X}$. Let $0 = \beta_{t,M+1} < \beta_{t,M} < \beta_{t,M-1} < \ldots < \beta_{t,1} < \beta_{t,0} = 1$ be some schedules in accordance with Section 5.4.3, where $t \in \mathbb{N}$ and M denotes the number of annealing runs. Note that, for each t, the schedules start with index $m = M + 1$ instead of $m = 0$. We write $\mu_{t+1,M+1} := K_t(x_t, \cdot)$, where x_t denotes the signal. It follows from Lemma 5.4.6 that

$$g_{t,m}(x_{t,m}^{(i)}, x_{t,m-1}^{(i)}) = c_{t,m} \exp\left(-(\beta_{t,m-1} - \beta_{t,m}) V_t(y_t - h_t(x_{t,m}^{(i)}))\right),$$

where

$$c_{t,m} := \frac{\langle \mu_{t,M+1}, \exp(-\beta_{t,m} V_t(y_t - h_t)) \rangle}{\langle \mu_{t,M+1}, \exp(-\beta_{t,m-1} V_t(y_t - h_t)) \rangle},$$

y_t denotes the observation at time t and h_t is defined in Chapter 3 for all $t \in \mathbb{N}$ and $1 \leq m \leq M + 1$. Note that the constant $c_{t,m}$ does not have to be calculated as mentioned in Section 5.4.3.

We proved the convergence of the generic particle filter (Algorithm 4.1) and the convergence of the interacting annealing algorithm (Algorithm 5.3) in the previous chapters, see Theorem 4.3.5 and Corollary 5.4.9, respectively. However, we cannot immediately conclude that the generalised annealed particle filter (Algorithm 6.1) converges since the random variables $X_{t+1,M+1}^{(i)}$ at the beginning of each updating step are not i.i.d. as assumed in Chapter 5, but conditionally independent.

Algorithm 6.1 Generalised Annealed Particle Filter

Requires: number of particles n, number of annealing runs M, parameters $(\epsilon_{t,m})_{1\leq m\leq M+1,t\in\mathbb{N}}$, weighting functions $(g_{t,m})_{1\leq m\leq M+1,t\in\mathbb{N}}$, transitions $(T_{t,m})_{1\leq m\leq M+1,t\in\mathbb{N}}$, η_0 and $(K_t)_{t\in\mathbb{N}_0}$ as defined in Chapter 3

1. <u>Initialisation</u>
   - $t \leftarrow 0$
   - For $i = 1, \ldots, n$, sample $x_{0,0}^{(i)}$ from η_0
2. <u>Prediction</u>
   - For $i = 1, \ldots, n$, sample $x_{t+1,M+1}^{(i)}$ from $K_t(x_{t,0}^{(i)}, \cdot)$
3. <u>Update (Interacting Annealing)</u>
   - For m from M to 0:
     - ∗ For $i = 1, \ldots, n$, sample $\tilde{x}_{t+1,m}^{(i)}$ from $T_{t+1,m+1}(x_{t+1,m+1}^{(i)}, \cdot)$
     - ∗ For $i = 1, \ldots, n$, set $\pi^{(i)} \leftarrow g_{t+1,m+1}(x_{t+1,m+1}^{(i)}, \tilde{x}_{t+1,m}^{(i)})$
     - ∗ Set $\bar{\pi} \leftarrow \sum_{j=1}^{n} \pi^{(j)}$
     - ∗ For i from 1 to n:

       Sample κ from $U[0,1]$

       If $\kappa \leq \epsilon_{t+1,m+1}\pi^{(i)}$ then

       ⋆ Set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(i)}$

       Else

       ⋆ Set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(j)}$ with probability $\frac{\pi^{(j)}}{\bar{\pi}}$
   - $t \leftarrow t + 1$ and go to step 2

For studying the convergence, one can introduce the function $\iota(l) \rightsquigarrow (t, m)$ defined by

$$\begin{aligned} t &= \left\lceil \frac{l}{M+2} \right\rceil \\ m &= t(M+2) - l, \end{aligned}$$

where $\lceil a \rceil$ denotes the smallest integer greater than or equal to a. This yields that $X_l := X_{\iota(l)}$ is a $\mathcal{X}$-valued stochastic process with initial distribution η_0 and transitions $(T_{\iota(l)})_{l\in\mathbb{N}_0}$ when putting $T_{t,0} := K_t$. Next, one constructs a Feynman-Kac model such that the algorithm describes a corresponding particle approximation and proves the convergence as in Chapter 5. However, we must pay attention to the assumption that the transitions $T_{t,m}$ leave

$$\frac{1}{\langle \mu_{t,M+1}, \exp(-\beta_{t,m-1} V_t(y_t - h_t)) \rangle} \exp\left(-\beta_{t,m-1} V_t(y_t - h_t(x))\right) \mu_{t,M+1}(dx)$$

invariant, for all $t \in \mathbb{N}$ and $1 \le m \le M+1$. This means that $T_{t,m}$ depends on $\mu_{t,M+1}$ and thus on the signal x_t. Therefore, the process $(X_{\iota(l)})_{l\in\mathbb{N}_0}$ does not satisfy the condition (2.3.2). A solution could be to consider the Markov process $(X_{t,0})_{t\in\mathbb{N}_0}$ or $(X_{t,M+1})_{t\in\mathbb{N}}$ on the one side and to model the "Update" steps as Markov transitions

$$S_{t,\eta_t} := T_{t,M+1} S_{(t,M+1),\eta_{t,M+1}} T_{t,M} S_{(t,M),\eta_{t,M}} \ldots T_{t,1} S_{(t,1),\eta_{t,1}}$$

on the other, where $S_{(t,m),\eta_{t,m}}$ denote the selection kernels and $T_{t,m}$ the mutation kernels used for the algorithm. The Markov transitions of the process $(X_{t,M+1})_{t\in\mathbb{N}}$ are then $K_{t,\eta_t} = S_{t,\eta_t} K_t$. Another approach would be to choose transitions T'_t and T_t in equation (5.4.7) that are more suitable.

Anyway, it is worth to mention two special cases of the generalised annealed particle filter (Algorithm 6.1). If $\epsilon_{t,m} = 0$ for all $1 \le m \le M+1$ and $t \in \mathbb{N}$, we get a combination of the generic particle filter (Algorithm 4.1) and the annealed importance sampling with resampling (Algorithm 5.4), see Algorithm 6.2. This particle filter is comparable with the annealed particle filter (Algorithm 7.1) established by Jonathan Deutscher et al. in [DeRe05] and [DeBR00]. These algorithms differ mainly in the weighting functions $g_{t,m}$ and the transitions $T_{t,m}$.

The second special case occurs when we set the parameters

$$\epsilon_{t,m}(\eta_{t,m}) := \frac{\epsilon'_{t,m}}{\langle \eta_{t,m}, g_{t,m} \rangle},$$

where $0 < \epsilon'_{t,m} \le 1/g$ and

$$g := \sup_{\substack{1 \le m \le M+1 \\ t \in \mathbb{N}}} \left(\sup_{x,y\in E} \left(\frac{g_{t,m}(x)}{g_{t,m}(y)} \right) \right) < \infty, \tag{6.0.1}$$

for all $1 \le m \le M+1$ and $t \in \mathbb{N}$, as proposed in [MoDo03]. The selection kernels (5.4.5) get

$$S_{(t,m),\eta_{t,m}}(x_{t,m}, \cdot) = \epsilon'_{t,m} \frac{g_{t,m}(x_{t,m})}{\langle \eta_{t,m}, g_{t,m} \rangle} \delta_{x_{t,m}} + \left(1 - \epsilon'_{t,m} \frac{g_{t,m}(x_{t,m})}{\langle \eta_{t,m}, g_{t,m} \rangle} \right) \Psi_{t,m}(\eta_{t,m}). \tag{6.0.2}$$

Note that the necessary condition

$$\left\| \epsilon'_{t,m} \frac{g_{t,m}}{\langle \eta_{t,m}, g_{t,m} \rangle} \right\|_\infty \leq 1$$

is satisfied since $g_{t,m}/\langle \eta_{t,m}, g_{t,m} \rangle \leq g$. If we set the number of particles $n \geq g$, then we can choose $\epsilon'_{t,m} = 1/n$. For some random variables $X^{(i)}_{t,m}$ and the random probability measure $\eta^n_{t,m} = \sum_{j=1}^n \delta_{X^{(i)}_{t,m}}/n$, we thus have

$$\epsilon'_{t,m} \frac{g_{t,m}(X^{(i)}_{t,m})}{\langle \eta^n_{t,m}, g_{t,m} \rangle} = \frac{g_{t,m}(X^{(i)}_{t,m})}{\sum_{j=1}^n g_{t,m}(X^{(j)}_{t,m})}.$$

Pierre del Moral showed in [Mora04, Chapter 9.4] that for any $t \in \mathbb{N}$ and $\varphi \in B(\mathcal{X})$ the sequence of random variables

$$\sqrt{n}(\langle \eta^n_t, \varphi \rangle - \langle \eta_t, \varphi \rangle)$$

converges in law to a Gaussian random variable W when the selection kernel in (5.4.5) is used to approximate the flow (5.4.2). It turns out that when we use $\epsilon'_t = 1/n$, the variance of W is strictly smaller than in the case with $\epsilon_t = 0$. This seems to indicate that it is preferable to use Algorithm 6.3 instead of Algorithm 6.2.

Algorithm 6.2 Generalised Annealed Particle Filter with $\epsilon_{t,m} = 0$

Requires: number of particles n, number of annealing runs M, weighting functions $(g_{t,m})_{1 \le m \le M+1, t \in \mathbb{N}}$, transitions $(T_{t,m})_{1 \le m \le M+1, t \in \mathbb{N}}$, η_0 and $(K_t)_{t \in \mathbb{N}_0}$ as defined in Chapter 3

1. Initialisation
   - $t \leftarrow 0$
   - For $i = 1, \ldots, n$, sample $x_{0,0}^{(i)}$ from η_0
2. Prediction
   - For $i = 1, \ldots, n$, sample $x_{t+1,M+1}^{(i)}$ from $K_t(x_{t,0}^{(i)}, \cdot)$
3. Update (Interacting Annealing)
   - For m from M to 0:
     * For $i = 1, \ldots, n$, set

       $$\pi^{(i)} \leftarrow \exp\Big(-(\beta_{t+1,m} - \beta_{t+1,m+1}) V_{t+1}(y_{t+1} - h_{t+1}(x_{t+1,m+1}^{(i)}))\Big)$$

     * For $i = 1, \ldots, n$, set $\pi^{(i)} \leftarrow \frac{\pi^{(i)}}{\sum_{j=1}^{n} \pi^{(j)}}$
     * For $i = 1, \ldots, n$, sample $\tilde{x}_{t+1,m}^{(i)}$ from $T_{t+1,m+1}(x_{t+1,m+1}^{(i)}, \cdot)$
     * For $i = 1, \ldots, n$, set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(j)}$ with probability $\pi^{(j)}$
   - $t \leftarrow t + 1$ and go to step 2

Algorithm 6.3 Generalised Annealed Particle Filter with $\epsilon'_{t,m} = \frac{1}{n}$

Requires: number of particles n, number of annealing runs M, weighting functions $(g_{t,m})_{1\leq m\leq M+1, t\in\mathbb{N}}$, transitions $(T_{t,m})_{1\leq m\leq M+1, t\in\mathbb{N}}$, η_0 and $(K_t)_{t\in\mathbb{N}_0}$ as defined in Chapter 3

1. Initialisation
   - $t \leftarrow 0$
   - For $i = 1, \ldots, n$, sample $x_{0,0}^{(i)}$ from η_0
2. Prediction
   - For $i = 1, \ldots, n$, sample $x_{t+1,M+1}^{(i)}$ from $K_t(x_{t,0}^{(i)}, \cdot)$
3. Update (Interacting Annealing)
   - For m from M to 0:
     * For $i = 1, \ldots, n$, set
       $$\pi^{(i)} \leftarrow \exp\Big(-(\beta_{t+1,m} - \beta_{t+1,m+1})V_{t+1}(y_{t+1} - h_{t+1}(x_{t+1,m+1}^{(i)}))\Big)$$
     * For $i = 1, \ldots, n$, set $\pi^{(i)} \leftarrow \frac{\pi^{(i)}}{\sum_{j=1}^{n} \pi^{(j)}}$
     * For $i = 1, \ldots, n$, sample $\tilde{x}_{t+1,m}^{(i)}$ from $T_{t+1,m+1}(x_{t+1,m+1}^{(i)}, \cdot)$
     * For i from 1 to n:
       Sample κ from $U[0,1]$
       If $\kappa \leq \pi^{(i)}$ then
       ⋆ Set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(i)}$
       Else
       ⋆ Set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(j)}$ with probability $\pi^{(j)}$
   - $t \leftarrow t + 1$ and go to step 2

7. Applications

In this chapter, we give some applications for the evaluation. Before discussing the applications, we state the annealed particle filter (Algorithm 7.1), which was introduced by Jonathan Deutscher et al., for convenience. As mentioned in Chapter 6, there are various possibilities to specify the selection kernels (5.4.5) for resampling. The first occurs when the parameters ϵ_t of the selection kernels are equal to zero. These selection kernels are implemented for the annealed particle filter, denoted by APF. For the second possibility, the parameters $\epsilon_t(\eta_t) = 1/(n \langle \eta_t, g_t \rangle)$ are used assuming the number of particles is large enough, see Chapter 6. For implementing the second case for the APF, we only need to replace the lines

- For $i = 1, \ldots, n$, set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(j)}$ with probability $\pi_{t+1,m}^{(j)}$

in Algorithm 7.1 by

- For i from 1 to n:

  Sample κ from $U[0,1]$

  If $\kappa \leq \pi_{t+1,m}^{(i)}$ then

  $\star$ Set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(i)}$

  Else

  $\star$ Set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(j)}$ with probability $\pi_{t+1,m}^{(j)}$

for $0 \leq m \leq M$. We denote the annealed particle filter with the second version of the selection kernels by APF_ϵ. In the sections below, we will make use of the following abbreviations:

GPF: Generic Particle Filter;

APF: Annealed Particle Filter ($\epsilon_t = 0$);

Algorithm 7.1 Annealed Particle Filter

Requires: number of particles n, number of annealing runs M, weighting functions $(g_t)_{t\in\mathbb{N}}$, transitions $(T_{t,m})_{1\leq m\leq M, t\in\mathbb{N}}$, η_0 and $(K_t)_{t\in\mathbb{N}_0}$ as defined in Chapter 3

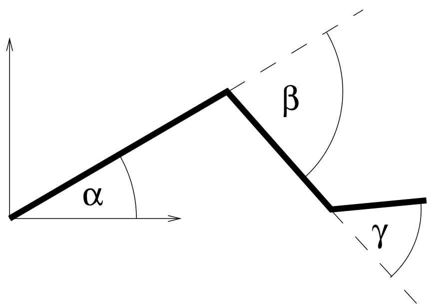
1. Initialisation
   - $t \leftarrow 0$
   - For $i = 1, \ldots, n$, sample $x_{0,0}^{(i)}$ from η_0
2. Prediction
   - For $i = 1, \ldots, n$, sample $\tilde{x}_{t+1,M}^{(i)}$ from $K_t(x_{t,0}^{(i)}, \cdot)$
3. Update (Annealing)
   - For m from M to 1:
     * For $i = 1, \ldots, n$, set $\pi_{t+1,m}^{(i)} \leftarrow g_{t+1}\left(y_{t+1}, \tilde{x}_{t+1,m}^{(i)}\right)^{\beta_{t+1,m}}$
     * For $i = 1, \ldots, n$, set $\pi_{t+1,m}^{(i)} \leftarrow \frac{\pi_{t+1,m}^{(i)}}{\sum_{j=1}^{n} \pi_{t+1,m}^{(j)}}$
     * For $i = 1, \ldots, n$, set $x_{t+1,m}^{(i)} \leftarrow \tilde{x}_{t+1,m}^{(j)}$ with probability $\pi_{t+1,m}^{(j)}$
     * For $i = 1, \ldots, n$, sample $\tilde{x}_{t+1,m-1}^{(i)}$ from $T_{t+1,m}(x_{t+1,m}^{(i)}, \cdot)$
   - For $i = 1, \ldots, n$, set $\pi_{t+1,0}^{(i)} \leftarrow g_{t+1}\left(y_{t+1}, \tilde{x}_{t+1,0}^{(i)}\right)$
   - For $i = 1, \ldots, n$, set $\pi_{t+1,0}^{(i)} \leftarrow \frac{\pi_{t+1,0}^{(i)}}{\sum_{j=1}^{n} \pi_{t+1,0}^{(j)}}$
4. Resampling
   - For $i = 1, \ldots, n$, set $x_{t+1,0}^{(i)} \leftarrow \tilde{x}_{t+1,0}^{(j)}$ with probability $\pi_{t+1,0}^{(j)}$
   - $t \leftarrow t + 1$ and go to step 2

APF_ϵ: Annealed Particle Filter ($\epsilon_t = 1/(n \langle \eta_t, g_t \rangle)$).

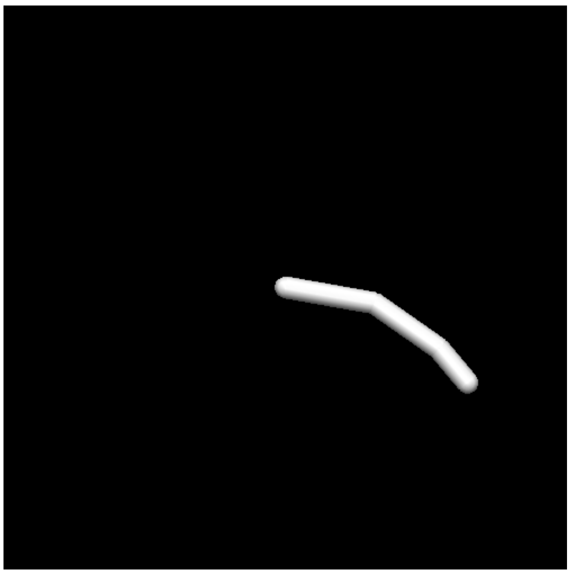
For evaluating these three algorithms, we consider two applications. The first application is in the field of visual tracking of articulated body motion, to where Jonathan Deutscher et al. ([DeBR00], [DeRe05]) apply the APF. In this connexion, the weighting function g_t does not represent a density as in Chapter 3. Instead, it should be regarded as a fitness function that measures the "quality" of a particle relative to an observation y_t. Whereas the application to the filtering problem is discussed in the second example.

7.1 Tracking Articulated Arm

In this section, we use the task of tracking an articulated arm for evaluating the algorithms. The arm consists of three limbs and three joints, namely shoulder, elbow and wrist. The shoulder is fixed at the origin of the coordinate system and the movement of the arm is restricted to the two-dimensional case. Hence, the position of the arm is completely described by the vector $x = (\alpha, \beta, \gamma)^T$, where $\alpha \in [-170, 170]$, $\beta \in [-125, 125]$ and $\gamma \in [-125, 125]$ denote the joint angles, as depicted in Figure 7.1(a). Note that the angles are measured in degrees. The arm is implemented in OpenGL by using quadrics, as shown in Figure 7.1(b).



(a) The postion of the arm is described by the vector $x = (\alpha, \beta, \gamma)^T$.



(b) Image of the OpenGL implementation of the arm.

Figure 7.1: Model of the articulated arm.

A sequence of images is generated by a stochastic process with state space $E := [-170, 170] \times [-125, 125] \times [-125, 125] \subset \mathbb{R}^3$. For initialisation, X_0 is uniformly distributed in E. That means that we do not know the position of the arm at the beginning. As transitions, we choose the Markov kernels

$$K_t(x_t, B) := c_t \int_B \exp\left(-\frac{1}{2}(x - x_t)^T \Sigma^{-1} (x - x_t)\right) dx, \qquad (7.1.1)$$

for $x_t \in E$ and $B \in \mathcal{E}$, where

$$\Sigma := \begin{pmatrix} 20 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 30 \end{pmatrix}$$

and

$$c_t := \int_E \exp\left(-\frac{1}{2}(x - x_t)^T \, \Sigma^{-1} \, (x - x_t)\right) \, dx.$$

This process models a difficult system for tracking since the velocity and the direction of the movement may change from frame to frame and the start position is unknown.

For calculating the weighting functions g_t, the image is converted to a binary image y_t by thresholding, as seen in Figure 7.2. The images y_t can be regarded as the observations of the signals x_t.

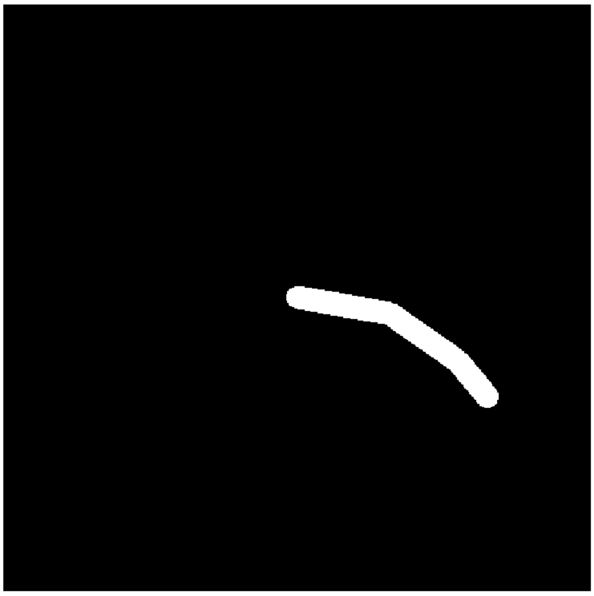


Figure 7.2: Image after thresholding: y_t.

Each particle $x_{t,m}^{(i)}$ takes values in the space E and determines a template for the articulated arm, as shown in Figure 7.3(a). The template consists of three rectangles with fixed size, where the positions of the rectangles are described by the joint angles. The conversion from an element $x \in E$ to the binary image is denoted by the function $h(x)$. When we have the template $h(x_{t,m}^{(i)})$ and the observation y_t, then an error map is calculated by the point operation $\neg y_t \wedge h(x_t^{(i)})$, as shown in Figure 7.3(b). When the particle $x_{t,m}^{(i)}$ is equal to the signal x_t, the error map is nearly black but not completely since the model used for the template does not exactly match the articulated arm.

Using the template and the error map, we introduce the following two variables:

N_p: Sum of the pixel values in the template;

N_e: Sum of the pixel values in the error map.

(a) Template $h(x_{t,m}^{(i)})$ (b) Error map $\neg y_t \wedge h(x_t^{(i)})$

Figure 7.3: Template and error map defined by a particle.

The weighting functions g_t are defined for $t \in \mathbb{N}$ by

$$g_t(y_t, x_{t,m}^{(i)}) := \exp\left(-4\,\frac{N_e}{N_p}\right). \tag{7.1.2}$$

Note that Jonathan et al. utilise similar weighting functions for tracking articulated body motion in [DeRe05] and [DeBR00]. However, they use edges as second image feature whereas we only use the silhouette. The edges are necessary when parts of the articulated object are in front of other parts. Since we restrict to the two-dimensional case in our example, this situation cannot occur. Hence, the edges would not give us more information about the position of the arm. Another aspect is the use of the factor 4 in the equation (7.1.2). It increases the difference between the minimum and the maximum of the weighting function, and it turned out in our simulations that the GPF performs better when using the factor for the weighting function. The factor is no longer necessary when more than one camera or more than one image feature is used since the difference between the minimum and the maximum is thereby greater, cf. [DeRe05] and [DeBR00]. The weighting function is plotted in Figure 7.4. In Figure 7.4(b), the graph of g_t over α is shown. As seen in Figure 7.4(a), the observation y_t, the elbow angle β and the wrist angle γ are fixed as the shoulder angle of the template α increases from -50 to 50. The graph of g_t over α and β is plotted in Figure 7.4(c) and Figure 7.4(d).

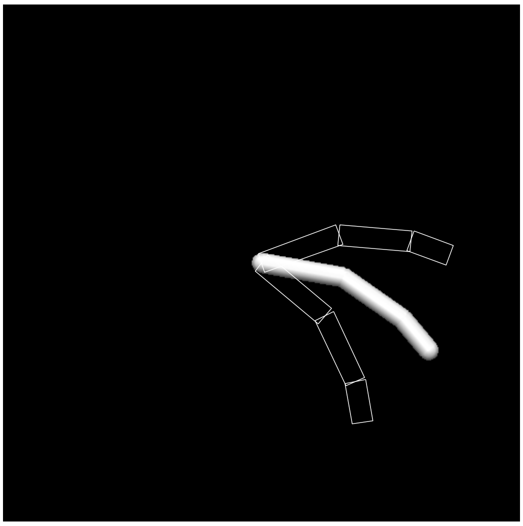
We observed that $\sqrt[4]{g_t}$ is in the range of 0.382 to 0.957 in our example. Hence, we get by equation (6.0.1) that

$$g = \sup_{t\in\mathbb{N}}\left(\sup_{\substack{x_1,x_2\in E\\ y_t}}\left(\frac{g_t(y_t,x_1)}{g_t(y_t,x_2)}\right)\right) = \left(\frac{0.957}{0.382}\right)^4 < 40.$$

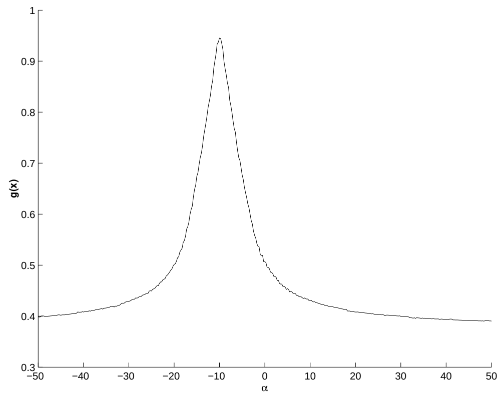
This means that the selection kernel (6.0.2) is valid if the number of particles is greater than or equal to 40.

For the simulation, we generated 201 images by the process mentioned above and applied the algorithms, implemented in MATLAB, to the sequence of images. Since the algorithms do not return an estimate for the first image, we got 200 estimates for the positions of the arm, where the estimates were computed between the “Updating” step and the “Resampling” step by

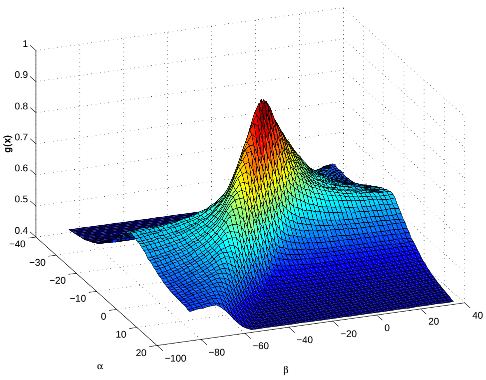
$$\hat{x}_t := \sum_{i=1}^{n} \pi_{t,0}^{(i)}\,\tilde{x}_{t,0}^{(i)},$$



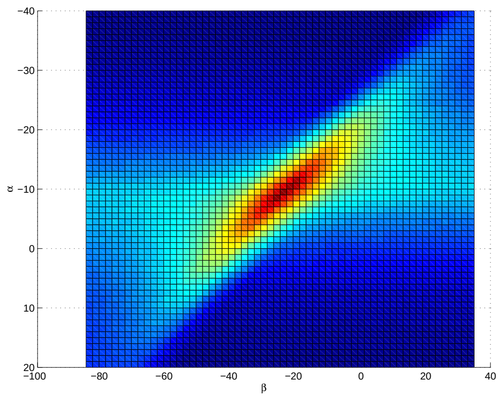
(a) Changing shoulder angle α of the template



(b) Graph of g_t over $\alpha \in [-50, 50]$



(c) Graph of g_t over $(\alpha, \beta) \in [-40, 20] \times [-85, 35]$



(d) Graph of g_t over $(\alpha, \beta) \in [-40, 20] \times [-85, 35]$

Figure 7.4: Weighting function.

for $1 \leq t \leq 200$. As measurement of the error, we used the weighting function without the factor 4:

$$g_t(y_t, \hat{x}_t) := \exp\left(-N_e/N_p\right), \tag{7.1.3}$$

where N_p and N_e are the sums of the pixel values in the template and the error map generated by the estimate $\hat{x}_t$, respectively. We computed the minimum of the error, the maximum of the error and the mean of the squared error for each sequence. This procedure describes one simulation run. Each simulation run was repeated 50 times, and the averages of the minimum, maximum and mean square error were calculated and are given in the tables below. The computations for 50 simulation runs took about 6 hours on a system equipped with an AMD Athlon XP 2800+ (2.09 GHZ) CPU and 1 GB RAM.

In the following Sections 7.1.1 - 7.1.4, we evaluate the performance of the APF and the APF_ϵ for a wide range of parameters and compare the results with the GPF. In Section 7.1.5, we investigate the case where the measurements are very noisy. Furthermore, the simulations are repeated with unknown dynamics of the arm in Sections 7.1.6 and 7.1.7. Finally in Section 7.1.8, we demonstrate the impact of the mixing condition on the APF.

7.1.1 Annealing Scheme

We evaluated the performance of the algorithms for various annealing schemes $0 < \beta_4 < \beta_3 < \beta_2 < \beta_1 < 1$, where we used the same scheme for the whole sequence $1 \leq t \leq 200$. Some examples are shown in Figure 7.5. The number of annealing runs M was set to 4, the initial distribution was the uniform distribution on E, and we chose the transitions $T_{t,m} = K_t$, as defined in (7.1.1), for $1 \leq m \leq 4$ and $1 \leq t \leq 200$.

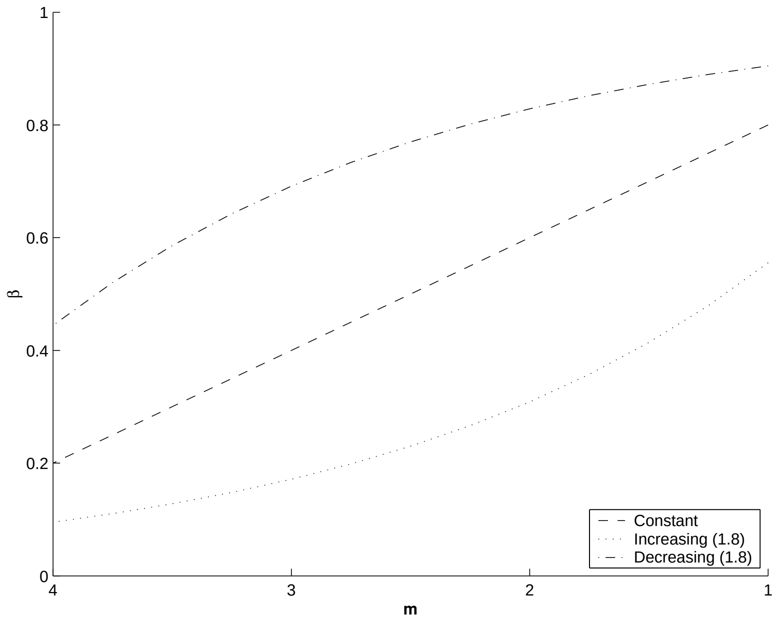


Figure 7.5: Annealing schemes with constant, increasing and decreasing increments.

In Table 7.1, the second column contains the used algorithm, namely GPF, APF and APF_ϵ, with the selected annealing scheme. There are three different schemes:

1. Constant increments by 0.2: $\beta_m = 0.2 \times (5 - m)$ (rows 2, 8);
2. Increasing increments with factor $c = 1.5$ and 1.8: $\beta_m = c^{-m}$ (rows 3, 9 and 4, 10, respectively);
3. Decreasing increments with factor $c = 1.5$ and 1.8: $\beta_m = 1 - c^{m-5}$ (rows 5, 11 and 6, 12, respectively).

Additionally, we evaluated the case where all β_m are equal to 1.0 in rows 7 and 13. To achieve the same computation time and to enable a fair comparison between the algorithms, the number of particles used for the GPF is divided by $M + 1$ for the other algorithms, as seen in column 3. The last three columns contain the minimum, the maximum and the mean of the squared error, as mentioned at the beginning of this section.

	$\beta_4 \leq \beta_3 \leq \beta_2 \leq \beta_1$	n	MIN	MAX	MSE
1.	GPF	250	0.0540	0.5825	0.0443
2.	APF (0.2 0.4 0.6 0.8)	50	0.0773	0.4351	0.0381
3.	APF (0.2 0.3 0.44 0.67)	50	0.0711	0.4778	0.0418
4.	APF (0.1 0.17 0.31 0.56)	50	0.0662	0.5189	0.0441
5.	APF (0.33 0.66 0.7 0.8)	50	0.0620	0.4214	0.0294
6.	APF (0.44 0.69 0.83 0.9)	50	0.0582	0.3849	0.0239
7.	APF (1.0 1.0 1.0 1.0)	50	0.0649	0.3471	0.0285
8.	APF_ϵ (0.2 0.4 0.6 0.8)	50	0.0588	0.4575	0.0280
9.	APF_ϵ (0.2 0.3 0.44 0.67)	50	0.0610	0.5000	0.0331
10.	APF_ϵ (0.1 0.17 0.31 0.56)	50	0.0664	0.5340	0.0400
11.	APF_ϵ (0.33 0.66 0.7 0.8)	50	0.0538	0.4010	0.0215
12.	APF_ϵ (0.44 0.69 0.83 0.9)	50	0.0504	0.3486	0.0204
13.	APF_ϵ (1.0 1.0 1.0 1.0)	50	0.0643	0.3553	0.0279

Table 7.1: 50 simulations with different values of $\beta_4 \leq \beta_3 \leq \beta_2 \leq \beta_1$.

The maximum of the error appears exceptional high at first glance. However, we said that the position of the articulated arm is unknown at the beginning. Thus the algorithms need about 4 - 6 frames until achieving good results, as demonstrated in Figures 7.6 - 7.8. Comparing the various annealing schemes for the APF, we find, cf. row 6, that the algorithm performs best using a scheme with decreasing increments, in particular with factor 1.8. Indeed, the mean square error is reduced by 46% in comparison to the GPF. The schemes with increasing increments are least efficient. The results for the APF_ϵ are similar. In row 12, we get the best performance and an error reduction of 54%. Moreover, the APF_ϵ outperforms the APF independent of the annealing scheme.

7.1.2 Number of Annealing Runs

We took the best annealing scheme $0.44 < 0.69 < 0.83 < 0.9$ and the parameter settings from Section 7.1.1 except for the number of annealing runs M. We set $M = 3$, $M = 4$ and $M = 5$ for evaluating. Note that the number of particles n were reduced to 62, 50 and 41, respectively, to achieve the same computation time.

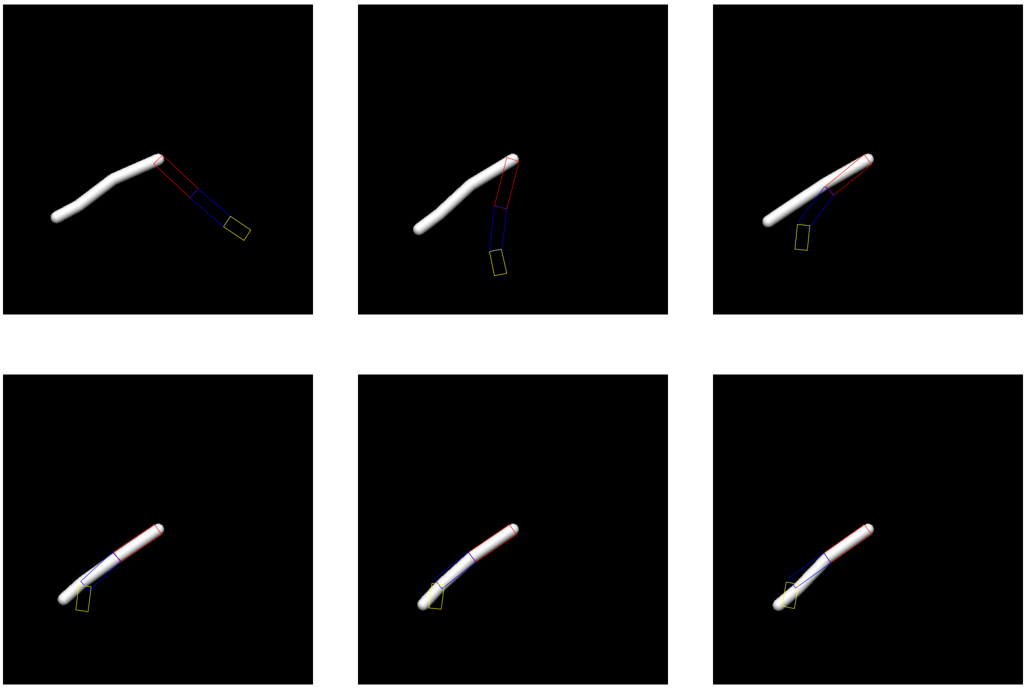


Figure 7.6: From top left to bottom right: Estimates (coloured) by the GPF for the articulated arm (white) at time $t = 1, \ldots, 6$.

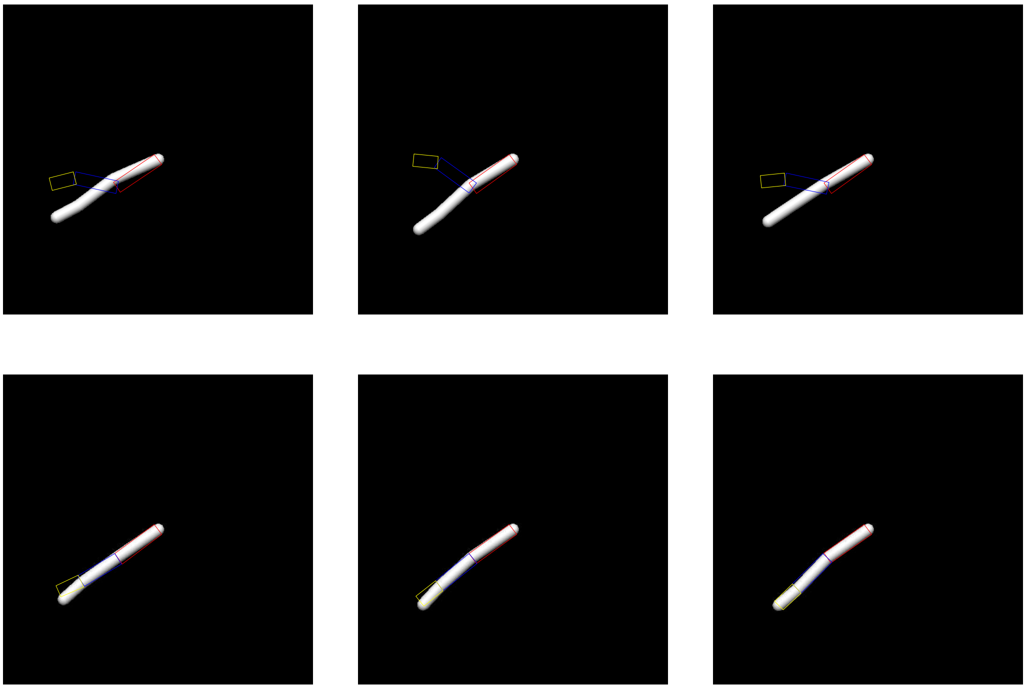


Figure 7.7: From top left to bottom right: Estimates (coloured) by the APF with annealing scheme $0.44 < 0.69 < 0.83 < 0.9$ for the articulated arm (white) at time $t = 1, \ldots, 6$.

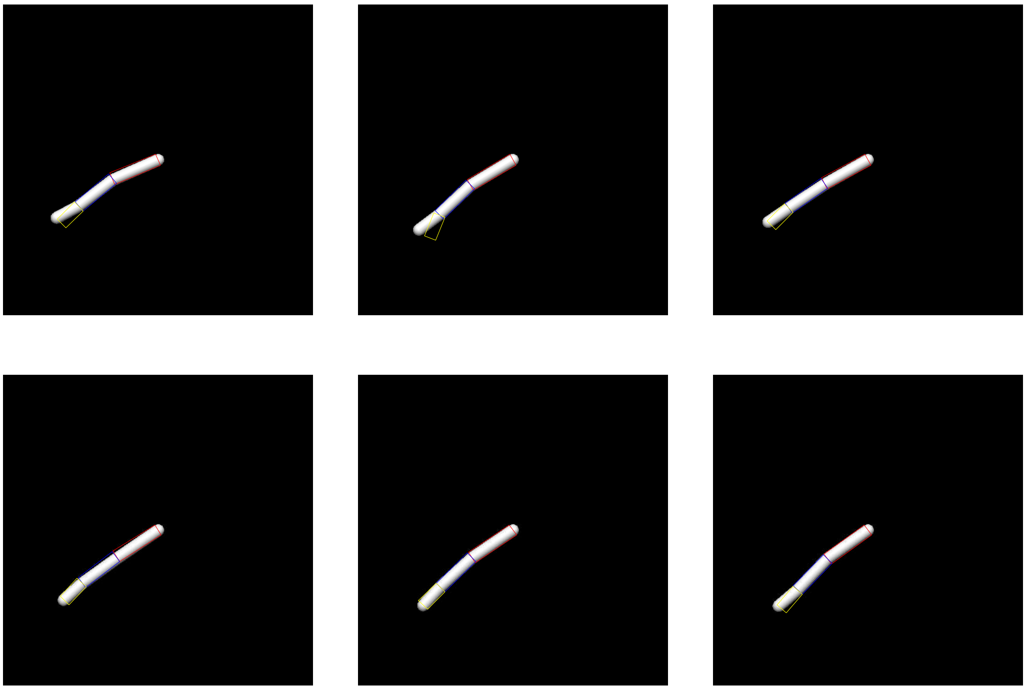


Figure 7.8: From top left to bottom right: Estimates (coloured) by the APF_ϵ with annealing scheme $0.44 < 0.69 < 0.83 < 0.9$ for the articulated arm (white) at time $t = 1, \ldots, 6$.

It can be seen from Table 7.2 that the least mean square error was obtained by $M = 4$, for both the APF and the APF_ϵ. Indeed, the number of annealing runs should be small. Jonathan Deutscher et al. ([DeBR00], [DeRe05]), for instance, used ten annealing runs for tracking human motion. On the one hand, a large number of annealing runs increases the annealing effect and reduces the error. On the other hand, the computation cost increases for each additional annealing run. This leads to a reduced number of particles and an increased error provided that the computation cost is fixed. Therefore, a compromise has to be found so that the advantage of the annealing effect exceeds the disadvantage of the reduced number of particles. Generally, we observed in our experiments that the particle filters perform poorly when the number of particles is below 30. When we compare the results of the APF and the APF_ϵ, we find that the APF_ϵ also outperforms the APF independent of the number of annealing runs.

7.1.3 Variance Scheme

The last parameters of the annealed particle filter for evaluating are the transitions $T_{t,m}$. We focus on Markov transitions of the form

$$T_{t,m}(x_{t,m}, B) := c_{t,m} \int_B \exp\left(-\frac{1}{2}(x - x_{t,m})^T \, \Sigma_m^{-1} \, (x - x_{t,m})\right) dx, \tag{7.1.4}$$

for $x_t \in E$ and $B \in \mathcal{E}$, where

$$\Sigma_m := \begin{pmatrix} \sigma_{m,1}^2 & 0 & 0 \\ 0 & \sigma_{m,2}^2 & 0 \\ 0 & 0 & \sigma_{m,3}^2 \end{pmatrix}$$

	$\beta_M \leq \beta_{M-1} \leq \ldots \leq \beta_1$	M	n	MIN	MAX	MSE
1.	GPF	-	250	0.0540	0.5825	0.0443
2.	APF (0.44 0.69 0.83)	3	62	0.0591	0.4320	0.0252
3.	APF (0.44 0.69 0.83 0.9)	4	50	0.0582	0.3849	0.0239
4.	APF (0.44 0.69 0.83 0.9 0.95)	5	41	0.0472	0.3623	0.0269
5.	APF_ϵ (0.44 0.69 0.83)	3	62	0.0511	0.4364	0.0204
6.	APF_ϵ (0.44 0.69 0.83 0.9)	4	50	0.0504	0.3486	0.0204
7.	APF_ϵ (0.44 0.69 0.83 0.9 0.95)	5	41	0.0541	0.3706	0.0222

Table 7.2: 50 simulations with different values of M.

	Variance ($\sigma_1^2\, \sigma_2^2\, \sigma_3^2$)	n	MIN	MAX	MSE
1.	GPF	250	0.0540	0.5825	0.0443
2.	APF (15 35 25)	50	0.0604	0.3784	0.0234
3.	APF (20 40 30)	50	0.0582	0.3849	0.0239
4.	APF (25 45 35)	50	0.0569	0.3792	0.0267
5.	APF (15 40 35)	50	0.0553	0.3778	0.0220
6.	APF (25 40 25)	50	0.0608	0.3929	0.0253
7.	APF_ϵ (15 35 25)	50	0.0537	0.3789	0.0222
8.	APF_ϵ (20 40 30)	50	0.0504	0.3486	0.0204
9.	APF_ϵ (25 45 35)	50	0.0609	0.3658	0.0224
10.	APF_ϵ (15 40 35)	50	0.0637	0.3611	0.0250
11.	APF_ϵ (25 40 25)	50	0.0660	0.3699	0.0247

Table 7.3: 50 simulations using a constant variance scheme with different values of Σ.

and

$$c_{t,m} := \int_E \exp\left(-\frac{1}{2}(x - x_{t,m})^T\, \Sigma_m^{-1}\, (x - x_{t,m})\right)\, dx,$$

for $1 \leq m \leq 4$ and $1 \leq t \leq 200$. Hence, the transitions are determined by the four covariance matrices Σ_4, Σ_3, Σ_2 and Σ_1, termed variance scheme. One could generalise the variance scheme by setting not all values that are not on the diagonal of the covariance matrix to zero. The introduction of correlation, however, makes it harder to find an optimal scheme. In the following, we write $(\sigma_{m,1}^2\, \sigma_{m,2}^2\, \sigma_{m,3}^2)$ instead of Σ_m. The number of annealing runs M was set to 4, and the other parameters were chosen as in Section 7.1.2.

As seen in Table 7.3, we investigated various constant variance schemes, that means $\Sigma := \Sigma_4 = \Sigma_3 = \Sigma_2 = \Sigma_1$. Comparing the mean square error of the APF, we find that the covariance matrix determined by (15 40 35), in row 5, performs best. When we look at the result of the APF_ϵ in row 10, we observe that the error is higher for the same variance scheme. This is not surprising since the advantage of the APF_ϵ is the variance reduction of the estimate, as discussed in Chapter 6. Therefore, the best variance scheme for the APF does not have to be the best for the APF_ϵ. Indeed, the best variance scheme for the APF_ϵ outperforms the best variance scheme for the APF, cf. row 5 and 8. In general, we can say that a new best variance scheme has to be found after any modification or improvement of the APF. It is interesting to remark that the transitions K_t of the stochastic process X_t (7.1.1) are determined by (20 40 30) since the least mean square errors were achieved by selecting similar variance schemes. The fact that we got better results for the APF with the scheme (15 40 35), that means a lower variance for the shoulder angle and a higher variance for the wrist angle, can be explained by the hierarchical structure of the articulated arm. If the shoulder angle is well estimated, in contrast to the elbow angle and wrist angle, the weighting function returns a higher value than if the estimates for the elbow angle and wrist angle are correct but the shoulder angle is estimated poorly. Hence, it is easier to estimate the shoulder angle than the wrist angle, as seen in Figures 7.6 - 7.8.

In the following, we do not restrict to constant variance schemes but to deterministic variance schemes, that means the variance schemes are given from the beginning. There are three different schemes in Table 7.4:

1. Constant decrements (rows 1 - 6, 12 -17);
2. Increasing decrements (rows 7 - 8, 18 - 19);
3. Decreasing decrements (rows 9 - 11, 20 - 22).

Column 2 contains the used algorithm and the values for $(\sigma_{4,1}^2\,\sigma_{4,2}^2\,\sigma_{4,3}^2)$ determining Σ_4. The decreasing scheme is described in column 3, whereby the notation is explained through the following examples. The variance scheme $(\sigma_{m,1}^2\,\sigma_{m,2}^2\,\sigma_{m,3}^2)$ is specified in

row 2 by $(\{27-(4-m)\times 4\}\,\{49-(4-m)\times 3\}\,\{41-(4-m)\times 2\})$,

row 7 by $(\{34\times\beta_{5-m}\}\,\{90\times\beta_{5-m}\}\,\{79\times\beta_{5-m}\})$,

row 8 by $(\{23-\sum_{k=m}^{4}1.5^{4-k}\}\,\{48-\sum_{k=m}^{4}1.5^{4-k}\}\,\{43-\sum_{k=m}^{4}1.5^{4-k}\})$.

Comparing the results of the APF, we find that the best constant variance scheme, row 1, was not significantly improved. The mean square errors for schemes with constant decrements, rows 2 - 6, are equal to the error of the constant variance scheme in the best cases. The schemes with increasing decrements are even worse. Only the schemes with decreasing decrements, rows 9 - 11, reduced the error. Before looking at the performance of the APF_ϵ in terms of these deterministic variance schemes, we should remark that the schemes were optimised for the APF. Thus, as discussed above, the APF_ϵ may perform better than the results in Table 7.4 indicate. Nevertheless, the APF_ϵ with the deterministic scheme in row 17 performs better than the best APF in row 9.

Furthermore, we observed in our experiments that the variance schemes for the APF did well when $(\sigma_{1,1}^2\,\sigma_{1,2}^2\,\sigma_{1,3}^2)$ was approximately (15 40 35). According to this, we selected the schemes in Table 7.4. Indeed, as seen in Table 7.5, the mean square error for the increasing variance scheme given in row 3 is less than the error for the corresponding decreasing variance scheme with $(\sigma_{1,1}^2\,\sigma_{1,2}^2\,\sigma_{1,3}^2) = (11\ 34\ 27)$ in row 2. This is surprising since we expected that the increasing variance scheme would give poor results.

	Variance ($\sigma^2_{4,1}\,\sigma^2_{4,2}\,\sigma^2_{4,3}$)	Decreasing Scheme	n	MIN	MAX	MSE
1.	APF (15 40 35)	$(-0\ -0\ -0)$	50	0.0553	0.3778	0.0220
2.	APF (27 49 41)	$(-4\ -3\ -2)$	50	0.0622	0.3778	0.0262
3.	APF (27 58 59)	$(-4\ -6\ -8)$	50	0.0470	0.3799	0.0221
4.	APF (27 80 59)	$(-4\ -10\ -8)$	50	0.0585	0.3653	0.0219
5.	APF (27 52 47)	$(-4\ -4\ -4)$	50	0.0596	0.3717	0.0224
6.	APF (24 52 50)	$(-3\ -4\ -5)$	50	0.0491	0.4037	0.0288
7.	APF (34 90 79)	$\times\beta_1\,\beta_2\,\beta_3\,\beta_4$	50	0.0637	0.3725	0.0320
8.	APF (22 47 42)	$-0\,1.5\,1.5^2\,1.5^3$	50	0.0632	0.3724	0.0287
9.	APF (22 47 42)	$-0\,1.5^3\,1.5^2\,1.5$	50	0.0567	0.3658	0.0207
10.	APF (36 97 85)	$\times0.8\,0.8^2\,0.8^3\,0.8^4$	50	0.0535	0.3874	0.0214
11.	APF (22 60 53)	$\times0.9\,0.9^2\,0.9^3\,0.9^4$	50	0.0540	0.3947	0.0216
12.	APF_ϵ (15 40 35)	$(-0\ -0\ -0)$	50	0.0637	0.3611	0.0250
13.	APF_ϵ (27 49 41)	$(-4\ -3\ -2)$	50	0.0536	0.3716	0.0206
14.	APF_ϵ (27 58 59)	$(-4\ -6\ -8)$	50	0.0476	0.3693	0.0197
15.	APF_ϵ (27 80 59)	$(-4\ -10\ -8)$	50	0.0507	0.3862	0.0200
16.	APF_ϵ (27 52 47)	$(-4\ -4\ -4)$	50	0.0560	0.3851	0.0291
17.	APF_ϵ (24 52 50)	$(-3\ -4\ -5)$	50	0.0461	0.3575	0.0192
18.	APF_ϵ (34 90 79)	$\times\beta_1\,\beta_2\,\beta_3\,\beta_4$	50	0.0545	0.3697	0.0258
19.	APF_ϵ (22 47 42)	$-0\,1.5^3\,1.5^2\,1.5$	50	0.0551	0.3819	0.0269
20.	APF_ϵ (22 47 42)	$-0\,1.5\,1.5^2\,1.5^3$	50	0.0566	0.3786	0.0273
21.	APF_ϵ (36 97 85)	$\times0.8\,0.8^2\,0.8^3\,0.8^4$	50	0.0536	0.4100	0.0250
22.	APF_ϵ (22 60 53)	$\times0.9\,0.9^2\,0.9^3\,0.9^4$	50	0.0507	0.3709	0.0247

Table 7.4: 50 simulations using a deterministic variance scheme with different values of Σ_m.

	Variance ($\sigma^2_{4,1}\,\sigma^2_{4,2}\,\sigma^2_{4,3}$)	Decreasing Scheme	n	MIN	MAX	MSE
1.	APF (27 58 59)	$(-4\ -6\ -8)$	50	0.0470	0.3799	0.0221
2.	APF (23 52 51)	$(-4\ -6\ -8)$	50	0.0704	0.3862	0.0269
3.	APF (3 22 11)	$(+4\ +6\ +8)$	50	0.0542	0.4035	0.0235

Table 7.5: 50 simulations using increasing and decreasing variance schemes, where $(\sigma^2_{1,1}\,\sigma^2_{1,2}\,\sigma^2_{1,3}) = (11\ 34\ 27)$ in row 2, and $(\sigma^2_{1,1}\,\sigma^2_{1,2}\,\sigma^2_{1,3}) = (15\ 40\ 35)$ in row 1 and 3.

7.1.4 Dynamic Variance Scheme

The deterministic variance schemes have the disadvantage that it is difficult to find the best one for an application. When the state space is high dimensional, as it is in tracking human motion, it is not feasible to determine for each joint angle an efficient variance σ^2. Moreover, the schemes are predefined for the complete sequence and do not react to changes of the motion, for example, when the movement of the articulated arm speeds up or slows down. Another drawback of these schemes is not taking the "quality" of the particles into account. The particles may well estimate the position of the arm already after the second annealing run, so a high variance could degrade the estimate. On the contrary, the particles may be far away from the real position after the second annealing run, then the variance predefined by the scheme could be too low for obtaining a good estimate. A solution to this problem is using dynamic variance schemes that depend on the particles and thus vary over time t.

For this purpose we set the covariance matrix $\Sigma_{t,m}$ proportional to the sampling variance after resampling, as suggested in [DeRe05]. That is, for a constant $c > 0$,

$$\Sigma_{t,m} := \frac{c}{n-1} \sum_{i=1}^{n} (x_{t,m}^{(i)} - \mu_{t,m})\,(x_{t,m}^{(i)} - \mu_{t,m})^T,$$

where

$$\mu_{t,m} := \frac{1}{n} \sum_{i=1}^{n} x_{t,m}^{(i)},$$

for $1 \leq m \leq 4$ and $1 \leq t \leq 200$. The other parameters were chosen as in Section 7.1.3. The algorithms with a dynamic variance scheme are denoted by APF^s and APF_ϵ^s.

	Constant c	n	MIN	MAX	MSE
1.	GPF	250	0.0540	0.5825	0.0443
2.	APF $-0\ 1.5^3\ 1.5^2\ 1.5$	50	0.0567	0.3658	0.0207
3.	APF^s 0.5	50	0.0479	0.5192	0.0311
4.	APF^s 0.25	50	0.0432	0.4961	0.0165
5.	APF^s 0.1	50	0.0497	0.4659	0.0163
6.	APF_ϵ^s 0.5	50	0.0512	0.4996	0.0325
7.	APF_ϵ^s 0.25	50	0.0430	0.4705	0.0145
8.	APF_ϵ^s 0.1	50	0.0450	0.4242	0.0143

Table 7.6: 50 simulations using a dynamic variance scheme with different values of c.

For comparing the dynamic schemes with the deterministic schemes and the GPF, the results of the GPF and of the APF with the best deterministic scheme are given in row 1 and row 2, respectively, of Table 7.6. The dynamic schemes are not only easier to handle since they have just one parameter c but also give a better performance, as seen in Table 7.6. Moreover, the APF_ϵ^s outperforms the APF^s provided that an appropriate parameter c is chosen, cf. rows 4, 5, 7 and 8. In comparison to the GPF, the mean square error was reduced by more than 67%.

7.1.5 Noisy Measurements

In the sections above, the quality of the images was perfect for tracking since the white model of the articulated arm was clearly silhouetted against the black background. In real world applications, however, we have to deal with noisy measurements caused by clutter, film grain, bad lighting conditions, CCD camera noise, etc. Therefore, we added strong noise to the weighting functions (7.1.2) by

$$g_t(y_t, x_{t,m}^{(i)}) := \exp\left(-4\,\frac{\vartheta\left(N_e + W_{t,m}^{(i)}\right)}{N_p}\right), \tag{7.1.5}$$

where

$$\vartheta(N) := \begin{cases} N_p & \text{if} \quad N > N_p, \\ N & \text{if} \quad 0 \leq N \leq N_p, \\ 0 & \text{if} \quad N < 0, \end{cases}$$

and $W_{t,m}^{(i)}$ are independent zero-mean Gaussian random variables with variance 8000. For comparison, a template (Figure 7.3(a)) consists of about 4000 pixels, that is $N_p \approx 4000$. The error was measured by the function (7.1.3) as above.

	$\beta_4 \leq \beta_3 \leq \beta_2 \leq \beta_1$	n	MIN	MAX	MSE
1.	GPF	250	0.0578	0.5846	0.0444
2.	APF (0.2 0.4 0.6 0.8)	50	0.0775	0.4602	0.0390
3.	APF (0.2 0.3 0.44 0.67)	50	0.0742	0.4905	0.0442
4.	APF (0.1 0.17 0.31 0.56)	50	0.0726	0.5207	0.0420
5.	APF (0.33 0.66 0.7 0.8)	50	0.0779	0.4250	0.0359
6.	APF (0.44 0.69 0.83 0.9)	50	0.0555	0.3613	0.0211
7.	APF (1.0 1.0 1.0 1.0)	50	0.0537	0.3251	0.0179
8.	APF_ϵ (0.2 0.4 0.6 0.8)	50	0.0676	0.4211	0.0410
9.	APF_ϵ (0.2 0.3 0.44 0.67)	50	0.0672	0.4595	0.0311
10.	APF_ϵ (0.1 0.17 0.31 0.56)	50	0.0802	0.5066	0.0485
11.	APF_ϵ (0.33 0.66 0.7 0.8)	50	0.0705	0.3915	0.0310
12.	APF_ϵ (0.44 0.69 0.83 0.9)	50	0.0663	0.3909	0.0319
13.	APF_ϵ (1.0 1.0 1.0 1.0)	50	0.0712	0.3720	0.0283

Table 7.7: 50 simulations with different values of $\beta_4 \leq \beta_3 \leq \beta_2 \leq \beta_1$.

As seen in Tables 7.7 - 7.11, we evaluated the algorithms for nearly the same parameter settings as in Sections 7.1.1 - 7.1.4. The rows and columns of the tables are explained in the mentioned sections above. Comparing the results for the APF in Table 7.7, we see that the "annealing" scheme with all $\beta_m = 1.0$ performs best. This shows that it may be suitable to relax the restriction that the annealing scheme is strictly increasing. Indeed, there is no mathematical necessity for the assumption that the difference of two succeeding β_{m+1} and β_m is strictly positive in contrast to the generalised annealed particle filter since the annealed particle filter uses β_m instead of the difference $\beta_m - \beta_{m+1}$ as exponent for the weighting function. Hence, the schemes for the annealed particle filter should satisfy

$$0 < \beta_{t,M} \leq \beta_{t,M-1} \leq \ldots \leq \beta_{t,1} \leq 1, \tag{7.1.6}$$

	$\beta_M \leq \beta_{M-1} \leq \ldots \leq \beta_1$	M	n	MIN	MAX	MSE
1.	GPF	-	250	0.0578	0.5846	0.0444
2.	APF (0.44 0.69 0.83)	3	62	0.0512	0.4313	0.0206
3.	APF (0.44 0.69 0.83 0.9)	4	50	0.0555	0.3613	0.0211
4.	APF (0.44 0.69 0.83 0.9 0.95)	5	41	0.0537	0.3558	0.0220
5.	APF_ϵ (0.44 0.69 0.83)	3	62	0.0548	0.4322	0.0242
6.	APF_ϵ (0.44 0.69 0.83 0.9)	4	50	0.0663	0.3909	0.0319
7.	APF_ϵ (0.44 0.69 0.83 0.9 0.95)	5	41	0.0715	0.3779	0.0379

Table 7.8: 50 simulations with different values of M.

	Variance ($\sigma_1^2\,\sigma_2^2\,\sigma_3^2$)	n	MIN	MAX	MSE
1.	GPF	250	0.0578	0.5846	0.0444
2.	APF (15 35 25)	50	0.0597	0.3997	0.0241
3.	APF (20 40 30)	50	0.0555	0.3613	0.0211
4.	APF (25 45 35)	50	0.0584	0.3962	0.0283
5.	APF (15 40 35)	50	0.0467	0.3664	0.0176
6.	APF (25 40 25)	50	0.0547	0.3337	0.0208
7.	APF_ϵ (15 35 25)	50	0.0724	0.4040	0.0240
8.	APF_ϵ (20 40 30)	50	0.0663	0.3909	0.0319
9.	APF_ϵ (25 45 35)	50	0.0554	0.3614	0.0216
10.	APF_ϵ (15 40 35)	50	0.0664	0.3677	0.0300
11.	APF_ϵ (25 40 25)	50	0.0547	0.3337	0.0208

Table 7.9: 50 simulations using a constant variance scheme with different values of Σ.

for all $t \in \mathbb{N}$.

This time, the APF_ϵ does not outperform the APF for all schemes. It seems that noisy measurements affect the APF_ϵ more than the APF, particularly when we compare the results with those from Table 7.1. However, we have to consider that the best variance scheme may have changed by the additional noise. In fact, as seen in Table 7.9, the APF_ϵ performs poorly with the constant variance scheme (20 40 30), which was used for evaluating the various annealing schemes.

In Table 7.8, the results for different number of annealing runs are given. Comparing the errors for both algorithms, we see that the mean square error is less for a smaller number of annealing runs while the maximum error is larger. The reason is that the position of the articulated arm is more easily found with more annealing runs at the beginning of the sequence but the algorithms work better with only few runs afterwards since the arm is a very simple object for tracking.

The best constant variance scheme for the APF was (15 40 35) also in the case of noisy measurements, cf. Table 7.9. Moreover, it was the best deterministic variance scheme, as seen in Table 7.10. Looking at the results of the APF_ϵ, we have to consider that the variance schemes were optimised for the APF. Hence, we may have achieved better results with other deterministic schemes for the APF_ϵ. However, when we

	Variance ($\sigma^2_{4,1}\,\sigma^2_{4,2}\,\sigma^2_{4,3}$)	Decreasing Scheme	n	MIN	MAX	MSE
1.	APF (15 40 35)	$(-0\ -0\ -0)$	50	0.0467	0.3664	0.0176
2.	APF (27 49 41)	$(-4\ -3\ -2)$	50	0.0499	0.3528	0.0182
3.	APF (27 58 59)	$(-4\ -6\ -8)$	50	0.0606	0.3599	0.0272
4.	APF (27 80 59)	$(-4\ -10\ -8)$	50	0.0511	0.3641	0.0182
5.	APF (27 52 47)	$(-4\ -4\ -4)$	50	0.0528	0.3610	0.0201
6.	APF (24 52 50)	$(-3\ -4\ -5)$	50	0.0549	0.3604	0.0240
7.	APF (22 47 42)	$-0\ 1.5\ 1.5^2\ 1.5^3$	50	0.0554	0.4099	0.0224
8.	APF (22 47 42)	$-0\ 1.5^3\ 1.5^2\ 1.5$	50	0.0522	0.3759	0.0230
9.	APF (22 60 53)	$\times 0.9\ 0.9^2\ 0.9^3\ 0.9^4$	50	0.0532	0.3829	0.0246
10.	APF_ϵ (15 40 35)	$(-0\ -0\ -0)$	50	0.0664	0.3677	0.0300
11.	APF_ϵ (27 49 41)	$(-4\ -3\ -2)$	50	0.0569	0.3933	0.0231
12.	APF_ϵ (27 58 59)	$(-4\ -6\ -8)$	50	0.0575	0.3537	0.0261
13.	APF_ϵ (27 80 59)	$(-4\ -10\ -8)$	50	0.0594	0.3706	0.0260
14.	APF_ϵ (27 52 47)	$(-4\ -4\ -4)$	50	0.0528	0.3593	0.0246
15.	APF_ϵ (24 52 50)	$(-3\ -4\ -5)$	50	0.0637	0.3760	0.0260
16.	APF_ϵ (22 47 42)	$-0\ 1.5^3\ 1.5^2\ 1.5$	50	0.0513	0.3614	0.0222
17.	APF_ϵ (22 47 42)	$-0\ 1.5\ 1.5^2\ 1.5^3$	50	0.0517	0.3651	0.0200
18.	APF_ϵ (22 60 53)	$\times 0.9\ 0.9^2\ 0.9^3\ 0.9^4$	50	0.0568	0.3859	0.0217

Table 7.10: 50 simulations using a deterministic variance scheme with different values of Σ_m.

compare the rows 2 and 4 of Table 7.11, we find that the best dynamic variance scheme outperforms the best deterministic variance scheme. Furthermore, it seems that the noisy measurements affect the APF_ϵ more than the APF indeed. While the least mean square error for the APF is similar to the one in Table 7.6, we could not achieve the same results for the APF_ϵ. Overall, the APF performs slightly better than the APF_ϵ when the noise is strong. Compared to the GPF, the mean square error was reduced by more than 63%. We finish this section with the following remark.

	Constant c	n	MIN	MAX	MSE
1.	GPF	250	0.0578	0.5846	0.0444
2.	APF (15 40 35)	50	0.0467	0.3664	0.0176
3.	APF^s 0.5	50	0.0485	0.5242	0.0270
4.	APF^s 0.25	50	0.0428	0.4693	0.0160
5.	APF^s 0.1	50	0.0447	0.4494	0.0170
6.	APF^s_ϵ 0.5	50	0.0483	0.5119	0.0294
7.	APF^s_ϵ 0.25	50	0.0459	0.4705	0.0195
8.	APF^s_ϵ 0.1	50	0.0474	0.4272	0.0172

Table 7.11: 50 simulations using a dynamic variance scheme with different values of c.

Remark 7.1.1. When the measurements are not noisy and the values of the states are exactly at the maximum of the weighting function, other algorithms such as optimisation algorithms may perform better than particle filters. This is due to the fact that the particle filters return the weighted average of the particles as estimate while optimisation algorithms usually return the best one. But these algorithms are not able to deal with noise since their results are easily misled by the noisy measurements. Particle filters, on the other side, compensate for this by using the average and are therefore more robust to noise. This can be seen when comparing the first rows of Table 7.6 and Table 7.11. The basic idea behind the robustness of the annealed particle filter is the following fact: Suppose there exist n random variables $(X_i)_{1\leq i\leq n}$ such that $\sum_{i=1}^{n} X_i/n$ converges almost surely to a random variable Z. If $(W_i)_{1\leq i\leq n}$ are i.i.d. random variables with zero mean, we get by the law of large numbers that $\sum_{i=1}^{n}(X_i + W_i)/n$ converges almost surely to Z.

7.1.6 Unknown Dynamics

In the previous sections, we assumed that the dynamics are exactly known. However, this is not the case in many applications. In this and the following sections, the transitions used for the "Prediction" steps in the algorithms differ from those of the process that generates the sequence of images. In return, the dynamics are relatively simple compared to those used above. The articulated arm starts at position $a := (-30, -80, -40)^T$ and moves to position $b := (50, 30, 20)^T$ with constant speed, as illustrated in Figure 7.9. Before the arm returns to the start position, it remains in this position for two frames. Moreover, we added some noise to each position vector $(\alpha_t, \beta_t, \gamma_t)^T$.

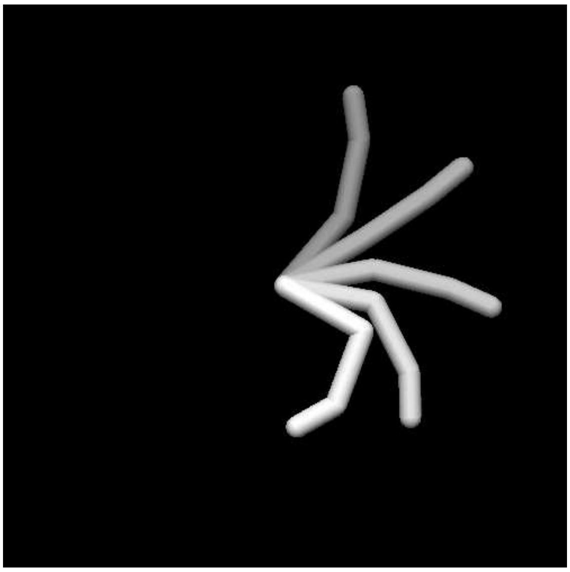


Figure 7.9: Motion sequence of the articulated arm.

Hence, the image sequence of length $T := 200$ is generated by the process

$$
\begin{aligned}
X_0 &:= a, && \\
X_t &:= a + (t-1)\frac{(b-a)}{98} + V_t && \text{for } 1 \leq t \leq 99, \\
X_t &:= b + V_t && \text{for } 100 \leq t \leq 101, \\
X_t &:= b - (t-102)\frac{(b-a)}{98} + V_t && \text{for } 102 \leq t \leq 200,
\end{aligned}
$$

where $(V_t)_{1\leq t\leq 200}$ are i.i.d. normal random variables with zero mean and covariance matrix

$$\frac{1}{98}\begin{pmatrix}(b_1-a_1) & 0 & 0\\ 0 & (b_2-a_2) & 0\\ 0 & 0 & (b_3-a_3)\end{pmatrix}.$$

	$\beta_4 \leq \beta_3 \leq \beta_2 \leq \beta_1$	n	MIN	MAX	MSE
1.	GPF	225	0.0517	0.1642	0.0115
2.	APF (0.2 0.4 0.6 0.8)	45	0.0431	0.1762	0.0097
3.	APF (0.2 0.3 0.44 0.67)	45	0.0432	0.1998	0.0119
4.	APF (0.1 0.17 0.31 0.56)	45	0.0477	0.2263	0.0153
5.	APF (0.33 0.66 0.7 0.8)	45	0.0416	0.1616	0.0083
6.	APF (0.44 0.69 0.83 0.9)	45	0.0411	0.1573	0.0077
7.	APF (1.0 1.0 1.0 1.0)	45	0.0409	0.1389	0.0064
8.	APF_ϵ (0.2 0.4 0.6 0.8)	45	0.0432	0.1831	0.0100
9.	APF_ϵ (0.2 0.3 0.44 0.67)	45	0.0443	0.2057	0.0119
10.	APF_ϵ (0.1 0.17 0.31 0.56)	45	0.0467	0.2310	0.0155
11.	APF_ϵ (0.33 0.66 0.7 0.8)	45	0.0420	0.1635	0.0085
12.	APF_ϵ (0.44 0.69 0.83 0.9)	45	0.0419	0.1565	0.0077
13.	APF_ϵ (1.0 1.0 1.0 1.0)	45	0.0404	0.1424	0.0066

Table 7.12: 40 simulations with different values of $\beta_4 \leq \beta_3 \leq \beta_2 \leq \beta_1$.

Since we assume that the dynamics are not known, we did not use the initial distribution and the transitions of the process above for the algorithms. These were instead initialised by the uniform distribution on $[-20,-40]\times[-60,-100]\times[-20,-60] \subset E$. That means that even though the start position of the arm is unknown the potential area is restricted by prior knowledge, which simplifies matters. For the "Prediction" step, we chose the transitions K_t, as defined in (7.1.1). The transitions $T_{t,m}$ were determined by $(\sigma^2_{m,1}\ \sigma^2_{m,2}\ \sigma^2_{m,3})$ in accordance with (7.1.4). As in the previous sections, we evaluated the algorithms for various annealing and variance schemes, where the meaning of the table rows and columns is the same.

In Tables 7.12 and 7.13, the constant variance scheme (5 5 5) was used for the simulations. Since the motion of the articulated arm is simple, local maxima rarely occur. Thus it is not astonishing that the scheme with all $\beta_m = 1.0$ performs best,

	$\beta_M \leq \beta_{M-1} \leq \ldots \leq \beta_1$	M	n	MIN	MAX	MSE
1.	GPF	-	225	0.0517	0.1642	0.0115
2.	APF (1.0 1.0)	2	75	0.0391	0.1232	0.0056
3.	APF (1.0 1.0 1.0)	3	62	0.0392	0.1295	0.0058
4.	APF (1.0 1.0 1.0 1.0)	4	45	0.0409	0.1389	0.0064
5.	APF_ϵ (1.0 1.0)	2	75	0.0383	0.1267	0.0056
6.	APF_ϵ (1.0 1.0 1.0)	3	62	0.0387	0.1327	0.0059
7.	APF_ϵ (1.0 1.0 1.0 1.0)	4	45	0.0404	0.1424	0.0066

Table 7.13: 40 simulations with different values of M.

	Variance ($\sigma_1^2\,\sigma_2^2\,\sigma_3^2$)	n	MIN	MAX	MSE
1.	GPF	225	0.0517	0.1642	0.0115
2.	APF (0.5 0.5 0.5)	45	0.0411	0.1513	0.0075
3.	APF (2 2 2)	45	0.0416	0.1467	0.0071
4.	APF (5 5 5)	45	0.0411	0.1573	0.0077
5.	APF (0.5 2 5)	45	0.0398	0.1561	0.0075
6.	APF (5 2 0.5)	45	0.0406	0.1571	0.0077
7.	APF_ϵ (0.5 0.5 0.5)	45	0.0420	0.1526	0.0076
8.	APF_ϵ (2 2 2)	45	0.0399	0.1489	0.0072
9.	APF_ϵ (5 5 5)	45	0.0419	0.1565	0.0077
10.	APF_ϵ (0.5 2 5)	45	0.0407	0.1528	0.0073
11.	APF_ϵ (5 2 0.5)	45	0.0408	0.1621	0.0079

Table 7.14: 40 simulations using a constant variance scheme with different values of Σ.

see rows 7 and 13 of Table 7.12. This demonstrates that only the repeating effect and not the annealing effect influences the results when the weighting function does not have any local maxima. Furthermore, we observed that the simpler the system is the lower the number of annealing runs is to choose, cf. Table 7.13. Finally, we remark that the mean square errors of the APF and the APF_ϵ do not differ significantly.

Using the annealing scheme $0.44 < 0.69 < 0.83 < 0.9$, we achieved the best result for the APF with the constant variance scheme (2 2 2), as seen in Table 7.14. The mean square error could not be reduced significantly neither by the deterministic variance schemes, with the best result in row 4 of Table 7.15, nor by the best dynamic scheme with $c = 0.025$, in row 5 of Table 7.16. However, the dynamic scheme is also recommendable for simple motion since we achieved nearly the same error as by the best deterministic scheme, cf. rows 2 and 5 of Table 7.16. For the APF_ϵ, we obtained the best result with a dynamic variance scheme in conjunction with the parameter $c = 0.05$. We see from Table 7.16 that the mean square error was reduced by more than 39% relative to the GPF. It is interesting to note that the error could not be reduced below 0.0070 by the various variance schemes using the annealing scheme $0.44 < 0.69 < 0.83 < 0.9$, while the algorithm performed obviously better with a lower number of annealing runs, as seen in Table 7.13. This leads to the conclusion that all the parameters should be chosen carefully for optimal performance.

7.1.7 Unknown Dynamics and Noisy Measurements

We evaluated the algorithms using the same dynamics and settings as in Section 7.1.6 but adding noise to the measurements by (7.1.5). When we compare Tables 7.12 - 7.16 with Tables 7.17 - 7.21, we see that the mean square errors rarely differ from those without noise by more than 0.0004. The only exceptions are in row 4 of Tables 7.15 and 7.20, and in row 10 of Tables 7.12 and 7.17. These are also the best and worst results from the previous section. This indicates that the noise has a low impact on the algorithms when the dynamics are simple. From the results of Section 7.1.5, we already expected that this is the case for the GPF and the APF, but in contrast to the difficult system, the results are not worse even for the APF_ϵ.

	Variance ($\sigma^2_{4,1}\,\sigma^2_{4,2}\,\sigma^2_{4,3}$)	Decreasing Scheme	n	MIN	MAX	MSE
1.	APF (2 2 2)	$(-0\ -0\ -0)$	45	0.0416	0.1467	0.0071
2.	APF (5 5 5)	$(-1\ -1\ -1)$	45	0.0404	0.1524	0.0075
3.	APF (8 8 8)	$(-2\ -2\ -2)$	45	0.0416	0.1541	0.0078
4.	APF (5 5 5)	$(-1.5\ -1.5\ -1.5)$	45	0.0400	0.1433	0.0070
5.	APF (5 6.5 8)	$(-1\ -1.5\ -2)$	45	0.0402	0.1546	0.0076
6.	APF (8 6.5 5)	$(-2\ -1.5\ -1)$	45	0.0403	0.1541	0.0076
7.	APF (9 9 9)	$-0\;1.5\;1.5^2\;1.5^3$	45	0.0407	0.1635	0.0079
8.	APF (9 9 9)	$-0\;1.5^3\;1.5^2\;1.5$	45	0.0415	0.1592	0.0075
9.	APF (3 3 3)	$\times 0.9\;0.9^2\;0.9^3\;0.9^4$	45	0.0408	0.1540	0.0073
10.	APF (2 2 2)	$(-0\ -0\ -0)$	45	0.0399	0.1489	0.0072
11.	APF (5 5 5)	$(-1\ -1\ -1)$	45	0.0417	0.1506	0.0073
12.	APF (8 8 8)	$(-2\ -2\ -2)$	45	0.0407	0.1560	0.0077
13.	APF (5 5 5)	$(-1.5\ -1.5\ -1.5)$	45	0.0396	0.1537	0.0074
14.	APF (5 6.5 8)	$(-1\ -1.5\ -2)$	45	0.0413	0.1511	0.0073
15.	APF (8 6.5 5)	$(-2\ -1.5\ -1)$	45	0.0410	0.1561	0.0077
16.	APF (9 9 9)	$-0\;1.5\;1.5^2\;1.5^3$	45	0.0422	0.1584	0.0078
17.	APF (9 9 9)	$-0\;1.5^3\;1.5^2\;1.5$	45	0.0405	0.1569	0.0076
18.	APF (3 3 3)	$\times 0.9\;0.9^2\;0.9^3\;0.9^4$	45	0.0404	0.1502	0.0075

Table 7.15: 40 simulations using a deterministic variance scheme with different values of Σ_m.

	Constant c	n	MIN	MAX	MSE
1.	GPF	225	0.0517	0.1642	0.0115
2.	APF $(-1.5\ -1.5\ -1.5)$	45	0.0400	0.1433	0.0070
3.	APF^s 0.1	45	0.0413	0.1760	0.0078
4.	APF^s 0.05	45	0.0399	0.1513	0.0072
5.	APF^s 0.025	45	0.0404	0.1513	0.0071
6.	APF^s 0.01	45	0.0396	0.1584	0.0076
7.	APF^s 0.1	45	0.0410	0.1582	0.0075
8.	APF^s 0.05	45	0.0406	0.1502	0.0070
9.	APF^s 0.025	45	0.0404	0.1510	0.0074
10.	APF^s 0.01	45	0.0397	0.1521	0.0074

Table 7.16: 40 simulations using a dynamic variance scheme with different values of c.

Moreover, we can observe that the optimal value of the constant c for the dynamic variance scheme increases when the measurements are noisy, cf. Tables 7.6, 7.11, 7.16 and 7.21. More precisely, this applies only for the APF since we obtained the least error for the APF_ϵ with the same value of the parameter c. If the optimal value of c were independent of the noise, it would be a great advantage of the APF_ϵ since this would make it easier to find an optimal configuration for the algorithm. For a conclusion, however, we need to evaluate the APF_ϵ for some more values of c.

	$\beta_4 \leq \beta_3 \leq \beta_2 \leq \beta_1$	n	MIN	MAX	MSE
1.	GPF	225	0.0503	0.1673	0.0114
2.	APF (0.2 0.4 0.6 0.8)	45	0.0434	0.1806	0.0097
3.	APF (0.2 0.3 0.44 0.67)	45	0.0438	0.1995	0.0116
4.	APF (0.1 0.17 0.31 0.56)	45	0.0472	0.2304	0.0152
5.	APF (0.33 0.66 0.7 0.8)	45	0.0416	0.1718	0.0087
6.	APF (0.44 0.69 0.83 0.9)	45	0.0405	0.1563	0.0077
7.	APF (1.0 1.0 1.0 1.0)	45	0.0408	0.1400	0.0065
8.	APF_ϵ (0.2 0.4 0.6 0.8)	45	0.0442	0.1828	0.0096
9.	APF_ϵ (0.2 0.3 0.44 0.67)	45	0.0471	0.2071	0.0120
10.	APF_ϵ (0.1 0.17 0.31 0.56)	45	0.0465	0.2262	0.0146
11.	APF_ϵ (0.33 0.66 0.7 0.8)	45	0.0430	0.1756	0.0088
12.	APF_ϵ (0.44 0.69 0.83 0.9)	45	0.0410	0.1534	0.0076
13.	APF_ϵ (1.0 1.0 1.0 1.0)	45	0.0393	0.1350	0.0065

Table 7.17: 40 simulations with different values of $\beta_4 \leq \beta_3 \leq \beta_2 \leq \beta_1$.

7.1.8 Mixing Condition

In Section 4.3, we discussed that the mixing condition, cf. Definition 4.3.2, is sufficient for the uniform convergence in time, where the idea is that any error is forgotten after some time. We illustrate now what can happen if the mixing condition is not met by the APF. For this purpose, we consider the task of tracking a stiff arm, i.e. $x = (\alpha, 0, 0)^T$, for a finite sequence of images. In contrast to the sections above, α is not restricted to the interval $[-170, 170]$. Since the angles of the elbow and the

	$\beta_M \leq \beta_{M-1} \leq \ldots \leq \beta_1$	M	n	MIN	MAX	MSE
1.	GPF	-	225	0.0503	0.1673	0.0114
2.	APF (1.0 1.0)	2	75	0.0393	0.1227	0.0056
3.	APF (1.0 1.0 1.0)	3	62	0.0393	0.1313	0.0060
4.	APF (1.0 1.0 1.0 1.0)	4	45	0.0408	0.1400	0.0065
5.	APF_ϵ (1.0 1.0)	2	75	0.0390	0.1253	0.0057
6.	APF_ϵ (1.0 1.0 1.0)	3	62	0.0380	0.1299	0.0060
7.	APF_ϵ (1.0 1.0 1.0 1.0)	4	45	0.0393	0.1350	0.0065

Table 7.18: 40 simulations with different values of M.

	Variance ($\sigma_1^2\,\sigma_2^2\,\sigma_3^2$)	n	MIN	MAX	MSE
1.	GPF	225	0.0503	0.1673	0.0114
2.	APF (0.5 0.5 0.5)	45	0.0411	0.1550	0.0075
3.	APF (2 2 2)	45	0.0416	0.1472	0.0072
4.	APF (5 5 5)	45	0.0405	0.1563	0.0077
5.	APF (0.5 2 5)	45	0.0402	0.1526	0.0073
6.	APF (5 2 0.5)	45	0.0407	0.1548	0.0078
7.	APF_ϵ (0.5 0.5 0.5)	45	0.0406	0.1514	0.0074
8.	APF_ϵ (2 2 2)	45	0.0400	0.1476	0.0073
9.	APF_ϵ (5 5 5)	45	0.0410	0.1534	0.0076
10.	APF_ϵ (0.5 2 5)	45	0.0416	0.1497	0.0073
11.	APF_ϵ (5 2 0.5)	45	0.0425	0.1541	0.0077

Table 7.19: 40 simulations using a constant variance scheme with different values of Σ.

	Variance ($\sigma_{4,1}^2\,\sigma_{4,2}^2\,\sigma_{4,3}^2$)	Decreasing Scheme	n	MIN	MAX	MSE
1.	APF (2 2 2)	$(-0\ -0\ -0)$	45	0.0416	0.1472	0.0072
2.	APF (5 5 5)	$(-1\ -1\ -1)$	45	0.0411	0.1531	0.0075
3.	APF (8 8 8)	$(-2\ -2\ -2)$	45	0.0410	0.1584	0.0076
4.	APF (5 5 5)	$(-1.5\ -1.5\ -1.5)$	45	0.0415	0.1506	0.0076
5.	APF (5 6.5 8)	$(-1\ -1.5\ -2)$	45	0.0414	0.1523	0.0075
6.	APF (8 6.5 5)	$(-2\ -1.5\ -1)$	45	0.0415	0.1578	0.0078
7.	APF (9 9 9)	$-0\,1.5\,1.5^2\,1.5^3$	45	0.0418	0.1624	0.0080
8.	APF (9 9 9)	$-0\,1.5^3\,1.5^2\,1.5$	45	0.0419	0.1550	0.0079
9.	APF (3 3 3)	$\times 0.9\,0.9^2\,0.9^3\,0.9^4$	45	0.0417	0.1509	0.0074
10.	APF (2 2 2)	$(-0\ -0\ -0)$	45	0.0400	0.1476	0.0073
11.	APF (5 5 5)	$(-1\ -1\ -1)$	45	0.0415	0.1525	0.0075
12.	APF (8 8 8)	$(-2\ -2\ -2)$	45	0.0417	0.1523	0.0075
13.	APF (5 5 5)	$(-1.5\ -1.5\ -1.5)$	45	0.0414	0.1524	0.0074
14.	APF (5 6.5 8)	$(-1\ -1.5\ -2)$	45	0.0406	0.1493	0.0073
15.	APF (8 6.5 5)	$(-2\ -1.5\ -1)$	45	0.0415	0.1577	0.0079
16.	APF (9 9 9)	$-0\,1.5\,1.5^2\,1.5^3$	45	0.0399	0.1548	0.0077
17.	APF (9 9 9)	$-0\,1.5^3\,1.5^2\,1.5$	45	0.0407	0.1577	0.0076
18.	APF (3 3 3)	$\times 0.9\,0.9^2\,0.9^3\,0.9^4$	45	0.0412	0.1546	0.0075

Table 7.20: 40 simulations using a deterministic variance scheme with different values of Σ_m.

	Constant c	n	MIN	MAX	MSE
1.	GPF	225	0.0503	0.1673	0.0114
2.	APF (2 2 2)	45	0.0416	0.1472	0.0072
3.	APF^s 0.1	45	0.0415	0.1630	0.0077
4.	APF^s 0.05	45	0.0409	0.1527	0.0072
5.	APF^s 0.025	45	0.0411	0.1515	0.0074
6.	APF^s 0.01	45	0.0423	0.1561	0.0078
7.	APF^s 0.1	45	0.0420	0.1652	0.0075
8.	APF^s 0.05	45	0.0412	0.1505	0.0071
9.	APF^s 0.025	45	0.0401	0.1570	0.0075
10.	APF^s 0.01	45	0.0401	0.1560	0.0076

Table 7.21: 40 simulations using a dynamic variance scheme with different values of c.

wrist are fixed, we write $x = \alpha$. We suppose that the motion of the articulated arm can be modelled exactly and is defined by the process

$$\begin{aligned} X_0 &:= 0, \\ X_t &:= X_{t-1} + V_t \qquad \text{for } 1 \leq t \leq 400, \end{aligned}$$

where V_t are i.i.d. uniform random variables on $[-10, 10]$. Let us examine the situations where $V_t(\omega) \in [9.75, 10]$ for $1 \leq t \leq 400$. Even though the probability that this occurs is very small, more precisely 0.0125^{400}, it is strictly greater than zero. We set the parameters of the APF as follows: 100 particles, one annealing run with $\beta_1 = 0.8$, initial distribution δ_0 and as transitions $T_{t,1}(x, \cdot)$ the uniform distributions on $[x - 2, x + 2]$.

First, we used the uniform distribution on $[x - 10, x + 10]$ for the transition kernels $K_t(x, \cdot)$ in accordance with the process $(X_t)_{0 \leq t \leq 400}$. These kernels do not fulfil the mixing condition, which is obvious from Example 4.3.3. As we see from Figure 7.10, the APF is not capable of tracking the articulated arm in this case. The tracking process was aborted when the error exceeded 0.55. The algorithm not only lost track of the arm after some time but was also not able to recover afterwards since the prediction of a particle $x^{(i)}_{t,0}$ is restricted to the space $[x^{(i)}_{t,0} - 10, x^{(i)}_{t,0} + 10]$. For a second simulation, the transition kernels $K_t(x, \cdot)$ were Gaussian with mean x and variance 100. Because of the circular motion of the arm, the state space is bounded, and thus the kernels are mixing. The probability that a particle $x^{(i)}_{t,0}$ is inside the interval $[x^{(i)}_{t,0} - 10, x^{(i)}_{t,0} + 10]$ after the "Prediction" step is then about 0.68, i.e.

$$P\left(x^{(i)}_{t,0} - 10 \leq \tilde{X}^{(i)}_{t+1,1} \leq x^{(i)}_{t,0} + 10\right) \approx 0.68.$$

We repeated the simulations 25 times and the maximal error of the estimate we obtained was 0.2766, see Figure 7.11.

This shows that the APF may fail when the mixing condition is not met, even though the particles are correctly predicted according to the dynamics. We remark that it is not necessary that each kernel of the algorithm is mixing. Instead, it is sufficient that the composite kernels $K_t T_{t,M} T_{t,M-1} \ldots T_{t,1}$ are mixing,

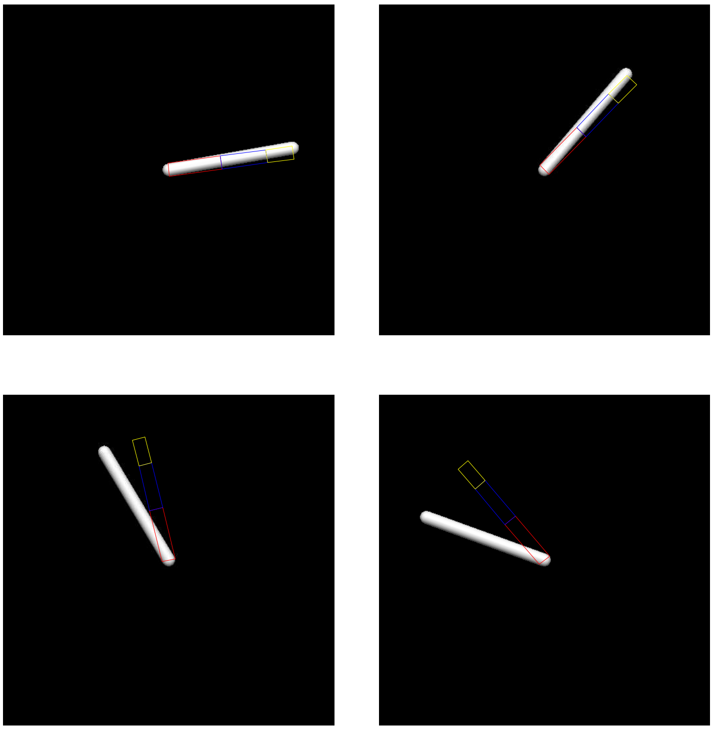


Figure 7.10: When the mixing condition is not met, the APF loses track of the articulated arm after some time and is not able to recover. From top left to bottom right: $t = 1,\ 5,\ 158,\ 165$.

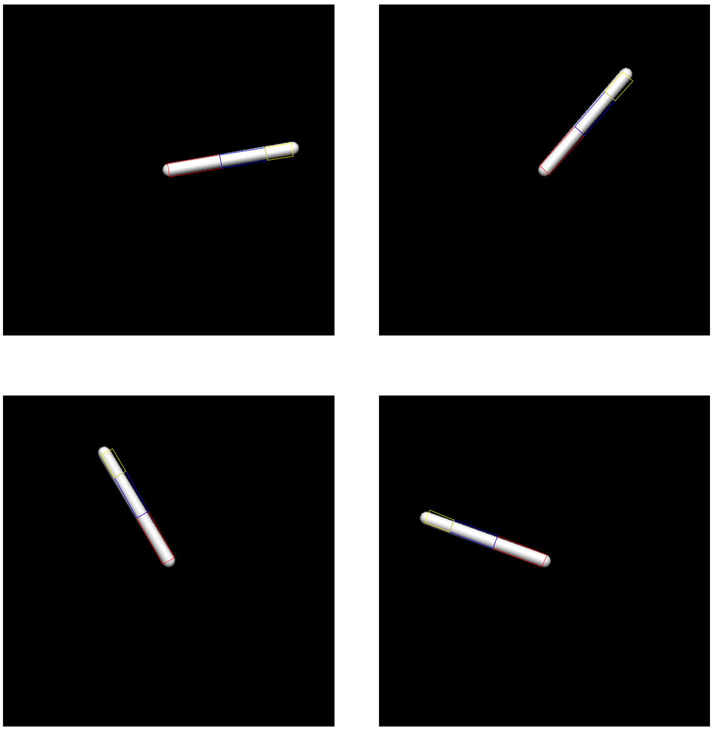


Figure 7.11: When the mixing condition is met, the APF is able to track the articulated arm. From top left to bottom right: $t = 1,\ 5,\ 158,\ 165$.

cf. [Mora04, Chapters 3.5.2 and 4]. In the example above, only the transitions K_t and not the transitions $T_{t,1}$ actually satisfied the mixing condition.

7.2 Filtering Problem

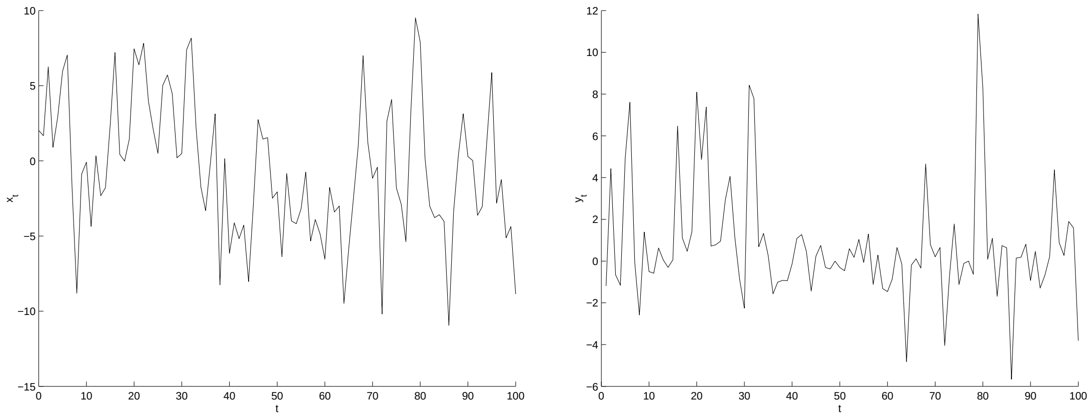
We give an example showing that the APF is inapplicable for the filtering problem as stated in Chapter 3. The reason for this is that the APF does not approximate a distribution, particularly the posterior distribution (3.1.1), by a weighted particle set but attempts to move the particles nearer to the global maximum of a fitness function. In order to achieve convergence to the posterior distribution, we discussed some modifications of the APF in Chapter 6, termed as generalised annealed particle filter.

For this purpose, we apply the algorithms to a slightly modified one-dimensional nonlinear example, where the extended Kalman filter does not work well, as shown in [GoSS93], [KiGe96] and [DoFG01, Chapter 9]. The signal and observation process are defined for $t \in \mathbb{N}$ by

$$X_t = \frac{X_{t-1}}{4} + 5\,\frac{X_{t-1}}{1+X_{t-1}^2} + 2\,\cos(1.2\,t) + V_t, \tag{7.2.1}$$

$$Y_t = \frac{X_t^2}{20} + \frac{X_t^3}{100} + W_t, \tag{7.2.2}$$

where V_t and W_t are independent zero-mean Gaussian random variables with variances 10 and 1, respectively, and X_0 having a standard normal distribution. The task is to estimate the unobserved signal x_t from the observations y_t assuming that the model above is known. Paths of the signal and the observation process are plotted in Figure 7.12(a) and Figure 7.12(a), respectively.



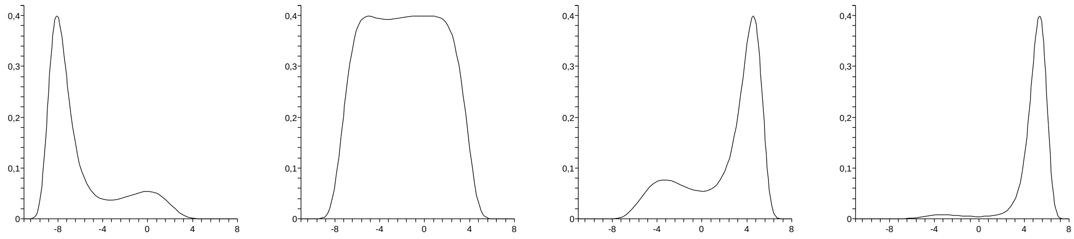
(a) Signal process x_t (b) Observation process y_t

Figure 7.12: Realisations of equations (7.2.1) and (7.2.2) for $t = 0 \ldots 100$.

It follows from Chapter 3 and Chapter 4 that the weighting functions g_t are given by

$$g_t(x_t, y_t) = \frac{1}{\sqrt{2\pi}}\,\exp\left(-\frac{(y_t - \frac{x_t^2}{20} - \frac{x_t^3}{100})^2}{2}\right), \tag{7.2.3}$$

for $t \in \mathbb{N}$. The functions have local and global maxima for different values of the observations y_t, as shown in Figure 7.13. But in contrast to the application in Section 7.1, the functions g_t represent densities determined by the observation process. Thus, the signal x_t is usually not located at the global maximum resulting in a poor performance of the APF.



(a) $y_t = -2$ (b) $y_t = 0$ (c) $y_t = 2$ (d) $y_t = 3$

Figure 7.13: The function $x_t \mapsto g_t(x_t, y_t)$, see (7.2.3), has a local maximum for different values of y_t.

Likewise to the example with the articulated arm, we evaluated the algorithms with various settings for the annealing scheme and the variance scheme. In our experiments, we generated a sequence of 200 values according to the equations (7.2.1) and (7.2.2). The estimates of the signal x_t were computed between the "Updating" step and the "Resampling" step by

$$\hat{x}_t := \sum_{i=1}^{n} \pi_{t,0}^{(i)} \, \tilde{x}_{t,0}^{(i)},$$

for $1 \leq t \leq 200$. We used the squared error

$$\sum_{t=1}^{200} (x_t - \hat{x}_t)^2$$

as measurement of the performance, where we calculated the minimum, the maximum and the average of the squared error for each sequence. We repeated each simulation 100 times, and the averages of the results are given in the tables below. The computations and the implementations of the algorithms were done in MATLAB.

The number of annealing runs was set to 4, and the Markov kernels

$$T_{t,m}(x_{t,m}, B) := \frac{1}{\sqrt{2\,\pi\,\sigma^2}} \int_B \exp\left(-\frac{(x - x_{t,m})^2}{2\,\sigma^2}\right) \, dx$$

were used as transitions $T_{t,m}$. Table 7.22 contains the results for various annealing schemes $\beta_4 < \beta_3 < \beta_2 < \beta_1$, where we set $\sigma^2 = 20$. For evaluating the deterministic and dynamic variance schemes for the APF, the best annealing scheme $0.2 < 0.3 < 0.44 < 0.67$ in Table 7.22 was selected. When comparing the errors of the APF both with different annealing schemes and with different variance schemes, as seen in Tables 7.22 - 7.24, it is obvious that the GPF performs better than the APF. This result agrees with our theoretical observations.

	$\beta_4 < \beta_3 < \beta_2 < \beta_1$	n	MIN	MAX	AVG
1.	GPF	300	0.0004	52.7474	6.7867
2.	APF (0.2 0.4 0.6 0.8)	60	0.0005	56.9980	7.8547
3.	APF (0.2 0.3 0.44 0.67)	60	0.0004	55.9007	7.8465
4.	APF (0.1 0.17 0.31 0.56)	60	0.0003	56.2017	7.8629
5.	APF (0.33 0.66 0.7 0.8)	60	0.0003	56.7339	7.8861
6.	APF (0.44 0.69 0.83 0.9)	60	0.0005	55.9324	7.8617

Table 7.22: 100 simulations with different values of $\beta_4 < \beta_3 < \beta_2 < \beta_1$.

	Variance σ^2	n	MIN	MAX	AVG
1.	GPF	300	0.0002	53.1010	6.8672
2.	APF 10	60	0.0008	63.6161	8.6935
3.	APF 15	60	0.0006	56.1458	8.1201
4.	APF 18	60	0.0005	55.9383	7.9897
5.	APF 20	60	0.0004	55.2982	7.9422
6.	APF 22	60	0.0005	56.3298	8.0193
7.	APF 26	60	0.0004	58.4089	7.9957
8.	APF 30	60	0.0003	59.4976	8.0778
9.	APF 40	60	0.0004	60.4589	8.1517

Table 7.23: 100 simulations with different values of σ^2.

	$\sigma_4^2 \geq \sigma_3^2 \geq \sigma_2^2 \geq \sigma_1^2$	n	MIN	MAX	AVG
1.	GPF	300	0.0003	50.9290	6.7794
2.	APF (20 20 20 20)	60	0.0006	56.2248	7.8748
3.	APF (26 24 22 20)	60	0.0004	56.5432	7.8331
4.	APF (32 28 24 20)	60	0.0004	55.6429	7.8601
5.	APF (38 32 26 20)	60	0.0005	55.1748	7.8894
6.	APF (34 26 22 20)	60	0.0003	55.4868	7.8481
7.	APF (26 25 23 20)	60	0.0006	56.4470	7.8630
8.	APF 120 $*$ (0.67 0.44 0.3 0.2)	60	0.0004	56.0294	7.9001
9.	APF 45 $*$ (0.8 0.64 0.51 0.41)	60	0.0006	55.5026	7.8727
10.	APF 34 $*$ (0.9 0.81 0.73 0.66)	60	0.0004	56.5076	7.8506
	c	n	MIN	MAX	AVG
11.	APF^s 4	60	0.0008	130.2932	11.4351
12.	APF^s 8	60	0.0008	129.6012	11.2247
13.	APF^s 16	60	0.0007	128.7720	11.1441

Table 7.24: 100 simulations with deterministic schemes ($\sigma_4^2 \geq \sigma_3^2 \geq \sigma_2^2 \geq \sigma_1^2$) and dynamic schemes ($c$).

8. Conclusion

We have developed a framework within which we discussed the mathematical properties of the annealed particle filter. For this purpose, we used the filtering problem stated in Chapter 3 as a fundamental model for the application of the algorithm. In accordance with the suggestion of Godsill and Clapp [DoFG01, Chapter 7], and based on the same ideas as the heuristic annealed particle filter, we have derived the generalised annealed particle filter in Chapter 6. It varies only slightly from the annealed particle filter but imposes a strong restriction on the transitions during the annealing step. This indicates that the annealed particle filter introduced by Jonathan Deutscher et al. does not converge to the posterior distribution and thus is not suitable for the filtering problem in contrast to other particle filters, like the generic particle filter. In addition, we validated this conclusion by our simulations in Section 7.2. The second difference concerns the annealing scheme. In contrast to the generalised annealed particle filter and the algorithms in Chapter 5, there is no mathematical reasoning for the requirement that the annealing schemes for the APF increase strictly. Therefore, we relaxed the assumption, cf. condition (7.1.6). Indeed, we discovered that the additional annealing schemes perform better in some situations.

Furthermore, we found that the mixing condition is sufficient for uniform convergence in time and should be fulfilled by the annealed particle filter even though it is not clear to where the algorithm converges. We demonstrated in Section 7.1.8 what might happen if the mixing condition is not met. Another important result from the mathematical framework is the generalisation of the selection kernel. According to this, we replaced the selection kernel of the APF by another selection kernel with better mathematical properties. This novel modification of the annealed particle filter was denoted by APF_ϵ. We showed that the APF_ϵ outperforms the APF as long as the noise of the measurements is not too strong. It would be interesting to compare the performance of the APF and the APF_ϵ in more sophisticated applications like human body motion tracking, for example.

Even though the fundamental model for the mathematical framework bases on the filtering problem, it is not restricted to this context. We can derive the convergence of the annealed particle filter directly from this framework and state even uniform convergence in time as long as the mixing condition is satisfied. However, it remains to solve the problem to which distribution the algorithm converges. More precisely, the question is: How does the limiting distribution correspond to the optimal estimate? For the filtering problem, we have seen that the limiting distribution does not agree with the posterior distribution. In the context of genetic algorithms, one would have to investigate if the algorithm converges to the global maximum of the fitness function as M goes to infinity whereby this function changes with a new observation y_t. One approach would be to describe the algorithm as an annealed Feynman Kac model, which was studied in [MoMi03] and [MoMi99]. However, an essential assumption for the convergence is then that the "inverse" temperature goes to infinity, similarly to simulated annealing [Haje88], and not to one, as it is the case for the annealed particle filter. Therefore, we cannot apply the results for simulated annealing, such as optimal temperature schemes [CoFi99], to the annealed particle filter.

According to our experiments, we observed that the annealing schemes with decreasing increments performed better than the schemes with constant or decreasing increments. Though we suspect that this is valid in most cases, the results do not provide evidence for a general conclusion since the optimal annealing scheme is likely to depend on the shape of the weighting function and thus on the application. Hence, more simulations would be necessary not only to corroborate the observation but also to evaluate different schemes with decreasing increments. Furthermore, we achieved better results with the additional annealing schemes in some cases, as mentioned above. Interestingly, this occurred when the measurements were noisy, as seen in Section 7.1.5. Moreover, the simulations support the following rule of thumb: the simpler the system the lower the optimal number of annealing runs. Finally, we found that the dynamic variance schemes, which were suggested by Jonathan Deutscher et al. in [DeRe05], outperform the deterministic variance schemes.

In addition to the already mentioned suggestions for further work, a better modelling might improve the performance of the annealed particle filter. One approach could be to allow correlation between the dimensions as suggested in Section 7.1.3. Another possibility for improvement is to use other branching procedures, like the ones proposed in [CrML99], instead of the resampling procedure of the generic particle filter as we remarked in Chapter 4. One could also relax the assumption that the number of particles n is fixed such that n may depend on the "quality" of the particles in the previous step. Indeed, this condition is not essential for the convergence of particle filters as shown in [CrML99].

Literature

[AaKo89] Emile Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines: A Stochastical Approach to Combinatorial Optimization and Neural Computing*. Wiley. 1989.

[AlSo72] D. L. Alspach and H. W. Sorenson. Nonlinear Bayesian Estimation using Gaussian Sum Approximations. *IEEE Transactions on Automatic Control* 17(4), 1972, p. 439–448.

[AMGC02] Sanjeev Arulampalam, Simon Maskell, Neil Gordon and Tim Clapp. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing* 50(2), 2002, p. 174–188.

[Baue90] Heinz Bauer. *Maß- und Integrationstheorie*. de Gruyter. 1990.

[Baue91] Heinz Bauer. *Wahrscheinlichkeitstheorie*. de Gruyter. 4 ed., 1991.

[Bill95] Patrick Billingsley. *Probability and Measure*. Wiley Series in Probability and Mathematical Statistics. Wiley. 3 ed., 1995.

[BlIs00] Andrew Blake and Michael Isard. *Active Contours*. Springer. 2000.

[ChLi04] Pavel Chigansky and Robert Liptser. Stability of Nonlinear Filters in Nonmixing Case. *The Annals of Applied Probability* 14(4), 2004, p. 2038–2056.

[CoFi99] Harry Cohn and Mark Fielding. Simulated Annealing: Searching for an Optimal Temperature Schedule. *SIAM Journal on Optimization* 9(3), 1999, p. 779–802.

[CrDo02] Dan Crisan and Arnaud Doucet. A Survey of Convergence Results on Particle Filtering Methods for Practitioners. *IEEE Transaction on Signal Processing* 50(3), 2002, p. 736–746.

[CrGr99] Dan Crisan and Malte Grunwald. Large Deviation Comparison of Branching Algorithms versus Resampling Algorithms: Application to Discrete Time Stochastic Filtering. Technical Report, Statistical Laboratory, Cambridge University, U.K., 1999.

[CrML99] Dan Crisan, Pierre Del Moral and Terry Lyons. Discrete Filtering Using Branching and Interacting Particle Systems. *Markov Processes and Related Fields* 5(3), 1999, p. 293–319.

[DeBR00] Jonathan Deutscher, Andrew Blake and Ian Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *Proc. Conf. Computer Vision and Pattern Recognition*, Vol. 2, 2000, p. 1144–1149.

[DeRe05] Jonathan Deutscher and Ian Reid. Articulated Body Motion Capture by Stochastic Search. *International Journal of Computer Vision* 61(2), 2005, p. 185–205.

[DoFG01] Arnaud Doucet, Nando de Freitas and Neil Gordon (Eds.). *Sequential Monte Carlo Methods in Practice.* Statistics for Engineering and Information Science. Springer. 2001.

[DoMo02] Arnaud Doucet and Pierre Del Moral. Sequential Monte Carlo Samplers. Technical Report 444, Cambridge University, 2002.

[GeGe84] Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 6, 1984, p. 721–741.

[Gelb01] Arthur Gelb (Ed.). *Applied Optimal Estimation.* MIT Press. 2001.

[Gida95] Basilis Gidas. *Topics in Contemporary Probability and Its Applications*, Chapter 7 Metropolis-type Monte Carlo Simulation Algorithms and Simulated Annealing, p. 159–232. Probability and Stochastics Series. CRC. 1995.

[GlOu04] François Le Gland and Nadia Oudjane. Stability and uniform approximation of nonlinear filters using the Hilbert metric and application to particle filters. *The Annals of Applied Probability* 14(1), 2004, p. 144–187.

[GoSS93] Neil Gordon, David Salmond and Adrian Smith. Novel Approach to Non-linear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings-F* 140(2), 1993, p. 107–113.

[HaHa67] J. M. Hammersley and D.C. Handscomb. *Monte Carlo Methods.* Methuen. 1967.

[Haje88] Bruce Hajek. Cooling Schedules for Optimal Annealing. *Mathematics of Operations Research* 13(2), 1988, p. 311–329.

[Hast70] W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika* 57(1), 1970, p. 97–109.

[HSFl00] Michael J. Black Hedvig Sidenbladh and David J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *European Conference on Computer Vision*, Vol. 2, 2000, p. 702–718.

[IsBl96] Michael Isard and Andrew Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *Proc. European Conference on Computer Vision*, Vol. 1, 1996, p. 343–356.

[IsBl98] Michael Isard and Andrew Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision* 29(1), 98, p. 5–28.

[Jazw70] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press. 1970.

[JuUh97] Simon J. Julier and Jeffrey K. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. In *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[KaKR95] Keiji Kanazawa, Daphne Koller and Stuart Russell. Stochastic Simulation Algorithms for Dynamic Probabilistic Networks. In *Proceedings of the Eleventh Annual Conference on Uncertainty in AI (UAI '95)*, 1995, p. 346–351.

[Kalm60] Emil Kalman, Rudolph. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering* 82(Series D), 1960, p. 35–45.

[KiGe96] Genshiro Kitagawa and Will Gersch. *Smoothness Priors Analysis of Time Series*, Vol. 116 of *Lecture Notes in Statistics*. Springer. 1996.

[KiJV83] S. Kirkpatrick, C. D. Gelatt Jr. and M.P. Vecchi. Optimization by Simulated Annealing. *Science* 220(4598), 1983, p. 671–680.

[MacC00] John MacCormick. *Probabilistic models and stochastic algorithms for visual tracking*. Dissertation, University of Oxford, 2000.

[MeTw93] Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Springer. 1993.

[MoDo03] Pierre Del Moral and Arnaud Doucet. On a Class of Genealogical and Interacting Metropolis Models. In *Séminaire de Probabilités XXXVII*, No. 1832 of Lecture Notes in Mathematics. Springer, 2003.

[MoDP04] Pierre Del Moral, Arnaud Doucet and Gareth W. Peters. Asymptotic and Increasing Propagation of Chaos Expansions for Genealogical Particle Models. *Publications du Laboratoire de Statistique et Probabilités*, 2004. Preprint.

[MoGu01] Pierre Del Moral and Alice Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincaré (B), Probabilités et statistiques* 37(2), 2001, p. 155–194.

[MoMi99] Pierre Del Moral and L. Miclo. On the Convergence and Applications of Generalized Simulated Annealing. *SIAM Journal on Control and Optimization* 37(4), 1999, p. 1222–1250.

[MoMi00] Pierre Del Moral and L. Miclo. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non linear filtering. In *Séminaire de Probabilités XXXIV*, No. 1729 of Lecture Notes in Mathematics, p. 1–145. Springer, 2000.

[MoMi03] Pierre Del Moral and L. Miclo. Annealed Feynman-Kac Models. *Communications in Mathematical Physics* vol. 235, 2003, p. 191–214.

[Mora98] Pierre Del Moral. Measure-valued Processes and Interacting Particle Systems. Application to Nonlinear Filtering Problems. *Annals of Applied Probability* 8(2), 1998, p. 438–495.

[Mora04] Pierre Del Moral. *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications.* Probability and its Applications. Springer. 2004.

[MRRT$^+$53] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller and Edward Teller. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* 21(6), 1953, p. 1087–1092.

[Neal98] Radford M. Neal. Annealed Inportance Sampling. Technical Report 9805, Department of Statistics, University of Toronto, 1998.

[PoWH88] A. Pole, M. West and P. J. Harrison. *Bayesian Analysis of Time Series and Dynamic*, Chapter Non-normal and non-linear dynamic Bayesian modelling. Dekker. 1988.

[RoCa02] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods.* Springer. 2002.

[RoWi01] L. C. G. Rogers and David Williams. *Diffusions, Markov Processes and Martingales*, Vol. 1. Cambridge University Press. 2 ed., 2001.

[SeVa05] Sunder Sethuraman and S. R. S. Varadhan. A martingale proof of Dobrushin's theorem for non-homogeneous Markov chains, 2005.

[Shir84] Albert N. Shiryaev. *Probability.* Springer. 1984.

Index

W