

Interacting and Annealing
Particle Filters: Mathematics
and a Recipe for Applications

Jürgen Gall Jürgen Potthoff
Christoph Schnörr Bodo
Rosenhahn Hans-Peter Seidel

MPI-I-2006-4-009 September 2006

Authors' Addresses

Jürgen Gall
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Jürgen Potthoff
Lehrstuhl für Mathematik V
Universität Mannheim
68131 Mannheim
Germany

Christoph Schnörr
Computer Vision, Graphics, and Pattern Recognition Group
Universität Mannheim
68131 Mannheim
Germany

Bodo Rosenhahn
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Acknowledgements

This research is partially funded by the Max-Planck Center for Visual Computing and Communication. We thank Uwe Kersting for providing the walking sequence.

Abstract

Interacting and annealing are two powerful strategies that are applied in different areas of stochastic modelling and data analysis. Interacting particle systems approximate a distribution of interest by a finite number of particles where the particles interact between the time steps. In computer vision, they are commonly known as particle filters. Simulated annealing, on the other hand, is a global optimization method derived from statistical mechanics. A recent heuristic approach to fuse these two techniques for motion capturing has become known as annealed particle filter. In order to analyze these techniques, we rigorously derive in this paper two algorithms with annealing properties based on the mathematical theory of interacting particle systems. Convergence results and sufficient parameter restrictions enable us to point out limitations of the annealed particle filter. Moreover, we evaluate the impact of the parameters on the performance in various experiments, including the tracking of articulated bodies from noisy measurements. Our results provide a general guidance on suitable parameter choices for different applications.

Keywords

Interacting Particle Systems, Particle Filter, Simulated Annealing, Annealed Particle Filter, Motion Capture

Contents

1	Introduction	2
1.1	Motivation	2
1.2	Related Work	4
1.3	Outline and Contribution	4
2	Particle Filter	6
2.1	Notations	6
2.2	Definition	6
2.3	Convergence	9
3	Interaction and Annealing	12
3.1	Feynman-Kac Model	12
3.2	Convergence	16
3.3	Interacting Simulated Annealing Algorithm	18
3.4	Interacting Annealed Sampling Algorithm	20
4	Evaluation	23
4.1	Motion Capturing	24
4.2	Mixing Condition	32
4.3	Filtering Problem	32
5	Conclusions	35

1 Introduction

1.1 Motivation

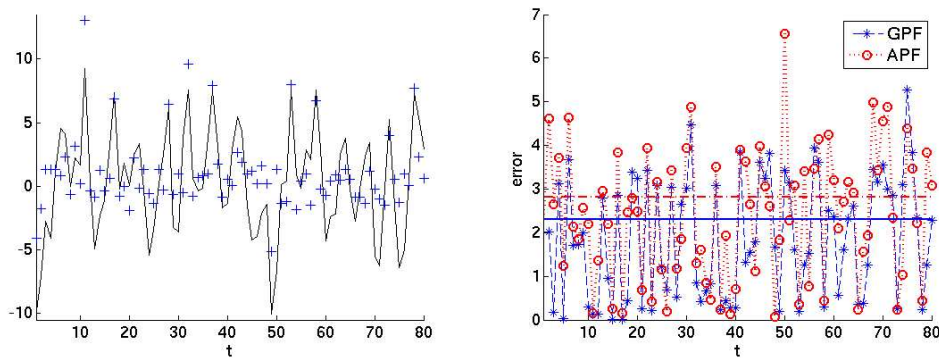


Figure 1.1: Filtering problem. **Left:** Observations (*crosses*) of an unknown signal (*solid line*). **Right:** Estimation error of the signal by a generic particle filter (*GPF*) and an annealed particle filter (*APF*). The root mean square errors are given by the horizontal lines. The *GPF* (*solid line*) outperforms the *APF* (*dash-dot line*).

Many real-world applications require the estimation of an unknown state of a system from given observations at each time step. An example from signal processing is shown on the left in Fig. 1.1 where the solid line represents the true signal and the crosses represent the measurements. The classical filtering problem consists in estimating the unknown signal from the observed measurements under some assumptions on the signal and on the observations. In 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. In human motion capturing for example, one estimates the state parameters such as joint angles and position of the human body in a

given image sequence. The estimated state is displayed by a mesh model of the body as depicted in Fig. 1.2.



Figure 1.2: Motion capturing. **Left:** Estimate by a *GPF* with 2750 particles. **Right:** Estimate by an *APF* with 250 particles. The *APF* outperforms the *GPF*.

During the last years, particle filters have become very popular for solving these problems. Reasons for their popularity include that they are easy to implement and they do not assume that the signal and observations can be well approximated by linear or Gaussian models like other filters [18, 16, 17, 1]. For an overview and numerous applications we refer the interested reader to [8].

While the mathematical fundamentals including convergence results have been developed further by Pierre del Moral in [24, 28, 25], a number of improved particle filters [8] have been proposed. A heuristically justified modification, the *annealed particle filter (APF)*, was introduced for articulated body motion tracking by Jonathan Deutscher et al. [6]. They demonstrated superior performance in experiments but, in view of the mathematical theory, did not gain further insight into bounds of the quality of estimates or restrictions necessary for the stability of the algorithm. As a result, it is

not clear if the convergence results as stated in the survey [3] are valid for the *APF*. Such results, however, would be helpful for further improvements, simplifying the parameter choice in applications, and for comparisons with alternative approaches.

Further motivation to relate the design of particle filters to the available mathematical theory is provided by two representative experimental results: While the *APF* outperforms a basic particle filter (defined in Section 2.2) in the domain of motion tracking as illustrated in Fig. 1.2, the *APF* falls short of the particle filter for a filtering problem as shown in Fig. 1.1.

1.2 Related Work

Particle filters [8] are recursive Bayesian filters that are based on Monte Carlo simulations [13]. They provide a convenient approach to approximate the distribution of interest. This technique is known as bootstrap filtering [12], condensation [15], Monte Carlo filters [21], survival of the fittest [19], and interacting particle approximations [25], depending on the area of research.

Convergence results have been established by Pierre Del Moral using discrete Feynman-Kac models [25]. These Feynman-Kac modelling techniques are powerful tools that can be applied in various domains of research. In the present paper, we restrict ourselves to two models with annealing properties that are related to the annealed particle filter [7]. The first model uses a principle similar to simulated annealing [20] which is a Markov process based method for optimization. The second model is inspired by annealed importance sampling [30]. It is an importance sampling method [13] that uses a sequence of densities for interpolation between a proposal density and the density of a complex target distribution.

1.3 Outline and Contribution

We begin with the fundamentals of particle filters and discuss convergence results under various assumptions as well as their impact on applications, particularly on motion capturing. Section 3 reveals the coherence between Feynman-Kac models and the annealed particle filter and explains the results shown in Figs. 1.1 and 1.2.

Specifically, the flows of the Feynman-Kac distributions and a particle approximation of these flows by the interacting annealing algorithm are given in Section 3.1. We state convergence properties of the algorithm in Section 3.2. While Section 3.3 presents an interacting version of simulated annealing that

converges to the regions of global minima, an interacting version of annealed importance sampling is derived in Section 3.4. We validate the conclusions from the theory and assess their impact on applications in Section 4 using a tracking and a filtering example, respectively. The results provide information for suitable parameter settings for motion capturing and show the robustness in the presence of noise. We conclude with a discussion and indicate further work in Section 5.

2 Particle Filter

In this section, we introduce a basic particle filter for solving a filtering problem as described in [8, Ch. 2] and [4]. Furthermore, we discuss its mathematical properties and explain the poor performance in the human motion capturing experiment, see Fig. 1.2.

2.1 Notations

Let (E, τ) be a topological space, and let $\mathcal{B}(E)$ denote its Borel σ -algebra. $B(E)$, $C_b(E)$ and $\mathcal{P}(E)$ denote the set of bounded measurable functions, bounded continuous functions and probability measures, respectively. δ_x is the Dirac measure concentrated in $x \in E$, and $\|\cdot\|_\infty$ is the supremum norm. Let $f \in B(E)$, $\mu \in \mathcal{P}(E)$, and let K be a Markov kernel on E . We write $\langle \mu, f \rangle = \int_E f(x) \mu(dx)$, $\langle K, f \rangle(x) = \int_E f(y) K(x, dy)$ for $x \in E$ and $\langle \mu, K \rangle(B) = \int_E K(x, B) \mu(dx)$ for $B \in \mathcal{B}(E)$. The *Dobrushin contraction coefficient* [10] is defined by

$$\beta(K) := \sup_{x_1, x_2 \in E} \sup_{B \in \mathcal{B}(E)} |K(x_1, B) - K(x_2, B)|.$$

Note that $\beta(K) \in [0, 1]$, and $\beta(K_1 K_2) \leq \beta(K_1) \beta(K_2)$. A family of transition kernels $(K_t)_{t \in \mathbb{N}_0}$ is said to satisfy the *Feller property* [32] if $\langle K_t, f \rangle \in C_b(E)$ for all t and $f \in C_b(E)$.

2.2 Definition

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

$$Y_t = h_t(X_t) + W_t \quad \text{for } t > 0, \quad Y_0 = 0,$$

where for each $t \in \mathbb{N}$, $h_t : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is a continuous function, $(W_t, t \in \mathbb{N})$ are independent m -dimensional random vectors and their distributions possess densities $g_t \in C_b(\mathbb{R}^m)$, $t \in \mathbb{N}$. The filtering problem consists in computing the conditional distribution

$$\eta_t(B) := P(X_t \in B \mid Y_t, \dots, Y_0), \quad (2.2.1)$$

for all $B \in \mathcal{B}(\mathbb{R}^d)$ or, alternatively, $\langle \eta_t, \varphi \rangle = E[\varphi(X_t) \mid Y_t, \dots, Y_0]$ for all $\varphi \in B(\mathbb{R}^d)$.

Algorithm 1 Generic Particle Filter

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

1. Initialization

- Sample $x_0^{(i)}$ from $\eta_0 \forall i$

2. Prediction

- Sample $\bar{x}_{t+1}^{(i)}$ from $K_t(x_t^{(i)}, \cdot) \forall i$

3. Updating

- Set $\pi_{t+1}^{(i)} \leftarrow g_{t+1}(y_{t+1} - h_{t+1}(\bar{x}_{t+1}^{(i)})) \forall i$
- Set $\pi_{t+1}^{(i)} \leftarrow \frac{\pi_{t+1}^{(i)}}{\sum_{j=1}^n \pi_{t+1}^{(j)}} \forall i$

4. Resampling

- Set $x_{t+1}^{(i)} \leftarrow \bar{x}_{t+1}^{(j)}$ with probability $\pi_{t+1}^{(j)} \forall i$ and go to step 2
-

The *generic particle filter (GPF)* is a commonly used particle filter for the solution of the filtering problem, which provides a basis for further developments and modifications for other applications. The algorithm consists of the four steps “Initialization”, “Prediction”, “Updating” and “Resampling”. During the initialization, we sample n times from the initial distribution η_0 . By saying that we sample $x^{(i)}$ from a distribution μ , we mean that we simulate n independent random samples, also named particles, according to μ . Hence, the n random variables $(X_0^{(i)})$ 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 y_t . The “Resampling” step can be regarded as a special case of a “Selection” step. The particles are selected in accordance with the weight function g_t . This step gives birth to some particles at the expense of light particles which die. For the particle filter also other “Resampling” steps than the one described in Algorithm 1 have been employed, e.g., branching procedures [4, 5, 8]. A detailed discussion can be found in [25, Ch. 11.8]. The particle system is also called *interacting particle system* [24] since the particles are (obviously) not independent after resampling.

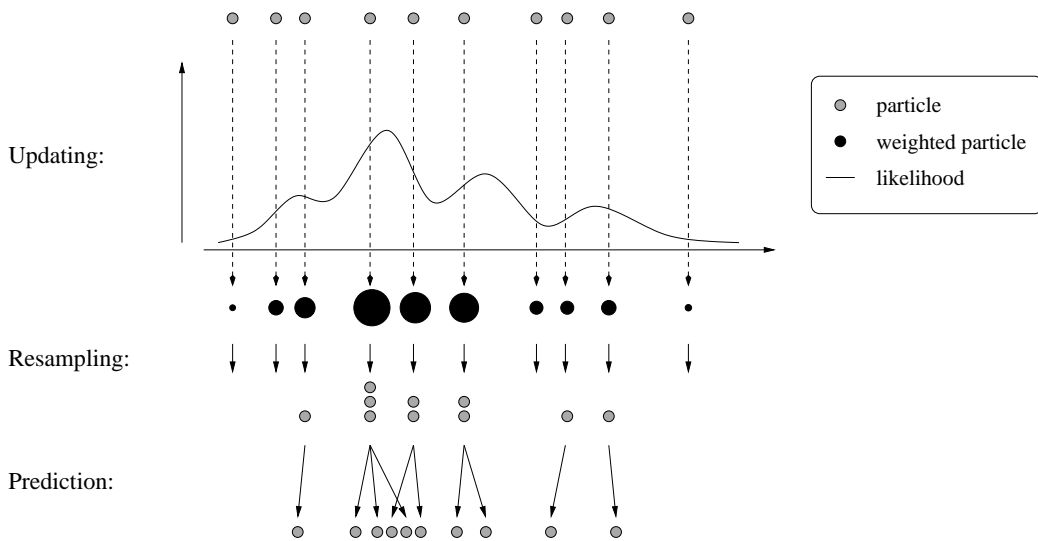


Figure 2.1: Operation of the generic particle filter.

For the case of a one-dimensional signal process, the operation of the algorithm is illustrated in Fig. 2.1, 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 large weight generate more offsprings than particles with lower weight during the “Resampling” step. In order to discuss the mathematical properties of the algorithm, we use the following notions (cf. also [22]).

Definition 2.2.1. 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 n th member of the sequence is

a set of n random variables $S^{(n)} = \{(X^{(1)}, \Pi^{(1)}), \dots, (X^{(n)}, \Pi^{(n)})\}$, where $\sum_{i=1}^n \Pi_i^{(n)} = 1$.

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

$$\sum_{i=1}^n \Pi^{(i)} \delta_{X^{(i)}} \quad \text{for } n \in \mathbb{N}.$$

The idea now is to approximate the conditional distribution η_t (2.2.1) by the distribution of 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:

$$\hat{\eta}_t^n := \frac{1}{n} \sum_{i=1}^n \delta_{\bar{X}_t^{(i)}}; \quad \bar{\eta}_t^n := \sum_{i=1}^n \Pi_t^{(i)} \delta_{\bar{X}_t^{(i)}}; \quad \eta_t^n := \frac{1}{n} \sum_{i=1}^n \delta_{X_t^{(i)}}.$$

With this notation, the algorithm is illustrated by the three separate steps

$$\eta_t^n \xrightarrow{\text{Prediction}} \hat{\eta}_{t+1}^n \xrightarrow{\text{Updating}} \bar{\eta}_{t+1}^n \xrightarrow{\text{Resampling}} \eta_{t+1}^n. \quad (2.2.2)$$

2.3 Convergence

The proof of the following convergence result can be found in [3].

Theorem 2.3.1. *For all $t \in \mathbb{N}_0$, there exists c_t independent of n such that*

$$E [(\langle \eta_t^n, \varphi \rangle - \langle \eta_t, \varphi \rangle)^2] \leq c_t \frac{\|\varphi\|_\infty^2}{n} \quad \forall \varphi \in B(\mathbb{R}^d). \quad (2.3.1)$$

Inequality (2.3.1) 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 state below a recent convergence result (Theorem 2.3.5) which is uniform in time 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 [11] and [27].

Definition 2.3.2. A kernel on E is called *mixing* if there exists a constant $0 < \varepsilon \leq 1$ and a measure μ on E such that

$$\varepsilon\mu(B) \leq K(x, B) \leq \frac{1}{\varepsilon}\mu(B) \quad \forall x \in E, B \in \mathcal{B}(E). \quad (2.3.2)$$

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

Example 2.3.3. Let $E = \{a, b\}$ and $K(x, B) := \delta_x(B)$. Assume that K is mixing. From inequality (2.3.2) we get the following contradiction

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

Example 2.3.4. Let $E = \mathbb{R}$ and

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

Suppose there exists an $\varepsilon > 0$ and a measure μ such that the inequality (2.3.2) is satisfied. Note that for all $x \in \mathbb{R}$ and all intervals $I = [a, b]$, $a < b$, we have $K(x, I) > 0$. Our assumption entails that $\mu(I) > 0$. But then $\varepsilon\mu(I) < K(x, I)$ cannot hold for all $x \in \mathbb{R}$, since $K(x, I) \rightarrow 0$ as $|x| \rightarrow +\infty$.

Le Gland and Oudjane [11] showed the uniform convergence of the generic particle filter (Theorem 2.3.5) by using the mixing condition for the family of random kernels $(R_t)_{t \in \mathbb{N}_0}$ defined by

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

Theorem 2.3.5. *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*

$$E [(\langle \eta_t^n, \varphi \rangle - \langle \eta_t, \varphi \rangle)^2] \leq c(\varepsilon) \frac{\|\varphi\|_\infty^2}{n} \quad \forall t \in \mathbb{N}_0, \varphi \in B(\mathbb{R}^d).$$

This means that as long as the mixing condition (2.3.2) is satisfied there exists an upper bound of the error that is independent of the time parameter. Hence, the number of particles, that ensures a given precision in an application, does not increase over time. An example that demonstrates the

impact of the condition is given in Section 4.2. The mixing condition can furthermore 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 [2].

It is important to note that the results above are only valid when the signal and the observation process are known and satisfy the assumptions stated at the beginning. Since this is rarely the case for applications, good approximations are needed. In applications like motion capture, it is very difficult to model the noise of the observation process in an appropriate way whereas a weight function g_t , which measures the “quality” of a particle based on some image features, can be easily designed such that the maximum is attained for the true value of the signal. In this case particle filters perform poorly as we will show in Section 4 and as illustrated in Fig. 1.2.

3 Interaction and Annealing

Before we go into detail, we sketch the idea of annealing. As seen in the top left image of Fig. 3.1, 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 weight function. One approach is to repeat the procedure, that means to let the particles undergo diffusion, to attribute weights to the particles, and to resample several times before the next time step. However as seen on the left hand side of Fig. 3.1, the particles might get stuck near a local maximum. To avoid this misbehavior, the particles are previously weighted by smoothed versions of the 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 depicted on the right hand side of Fig. 3.1. In the following sections, we discuss Feynman-Kac models with annealing properties and reveal relations to the annealed particle filter [7] that also relies on this annealing effect.

3.1 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 $t \in \mathbb{N}_0$,

$$P_{\eta_0}(d(x_0, x_1, \dots, x_t)) = K_{t-1}(x_{t-1}, dx_t) \dots K_0(x_0, dx_1) \eta_0(dx_0),$$

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

$$E \left[\prod_{s=0}^t g_s(X_s) \right]_{\eta_0} > 0 \quad \forall t \in \mathbb{N}_0.$$

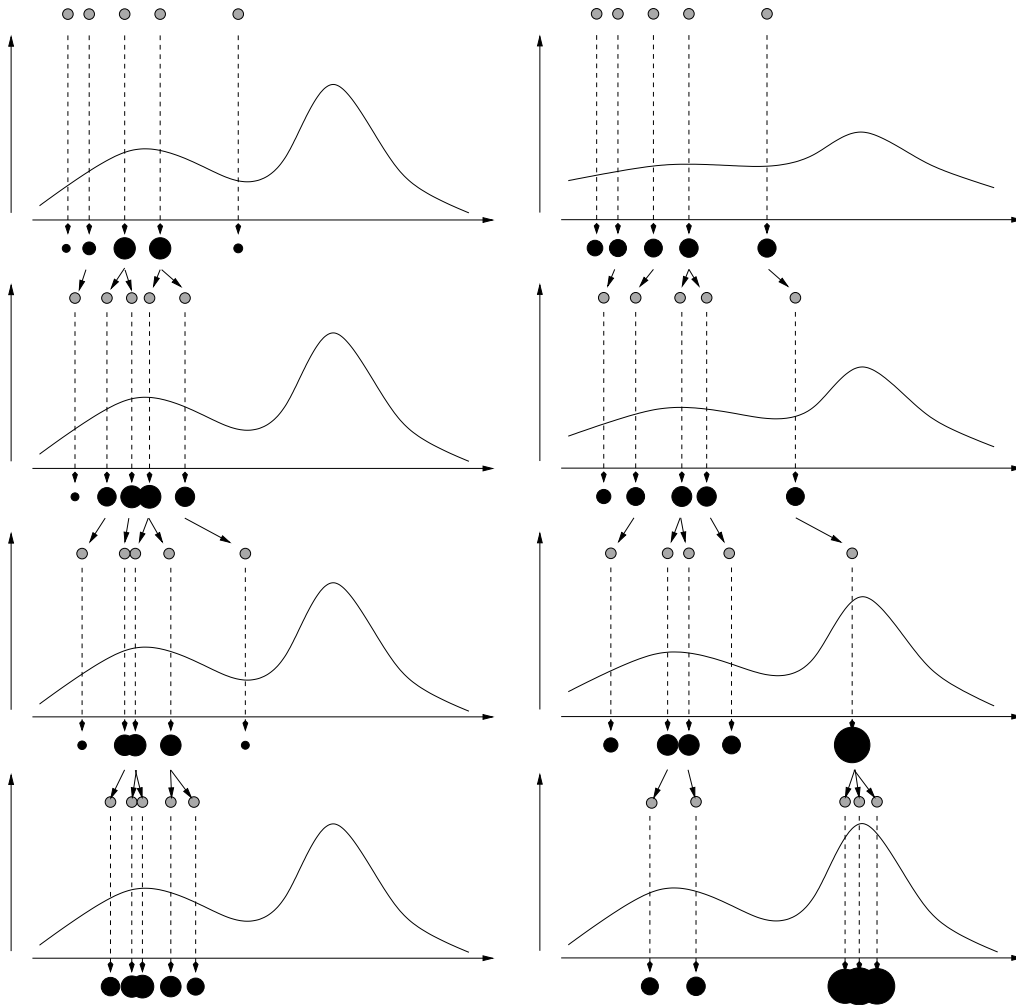


Figure 3.1: **Left:** Without an annealing effect, the particles get stuck in the local maximum. **Right:** The annealing effect ensures that the particles escape from the local maximum.

Definition 3.1.1. The sequence of distributions $(\eta_t)_{t \in \mathbb{N}_0}$ on E defined for any $\varphi \in B(E)$ as

$$\langle \eta_t, \varphi \rangle := \frac{\langle \gamma_t, \varphi \rangle}{\langle \gamma_t, 1 \rangle}, \quad \langle \gamma_t, \varphi \rangle := E \left[\varphi(X_t) \prod_{s=0}^{t-1} g_s(X_s) \right]_{\eta_0} \quad \forall t \in \mathbb{N}_0, \quad (3.1.1)$$

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

Example 3.1.2. The functions $(g_t)_{t \in \mathbb{N}_0}$ are often chosen as unnormalized Boltzmann-Gibbs measures $g_t(x) = \exp(-\beta_t V_t(x))$. Equation (3.1.1) then becomes

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

It is straightforward to check that the Feynman-Kac model as defined above satisfies the recursion relation

$$\eta_{t+1} = \langle \Psi_t(\eta_t), K_t \rangle, \quad (3.1.2)$$

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

$$\Psi_t(\eta_t)(dx_t) := \frac{1}{\langle \eta_t, g_t \rangle} g_t(x_t) \eta_t(dx_t). \quad (3.1.3)$$

The particle approximation of the flow (3.1.2) depends on a chosen family of Markov transition kernels $(K_{t, \eta_t})_{t \in \mathbb{N}_0}$ satisfying the compatibility condition

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

The family $(K_{t, \eta_t})_{t \in \mathbb{N}_0}$ of kernels is not uniquely determined by these conditions. For example, we can choose, as in [25, Ch. 2.5.3],

$$K_{t, \eta_t} = S_{t, \eta_t} K_t, \quad (3.1.4)$$

where

$$S_{t, \eta_t}(x_t, dy_t) = \epsilon_t g_t(x_t) \delta_{x_t}(dy_t) + (1 - \epsilon_t g_t(x_t)) \Psi_t(\eta_t)(dy_t), \quad (3.1.5)$$

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

Example 3.1.3. We continue Example 3.1.2. The selection kernel becomes

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

where

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

Algorithm 2 Interacting Annealing Algorithm

Requires: parameters $(\epsilon_t)_{t \in \mathbb{N}_0}$, number of particles n , initial distribution η_0 , weighting functions $(g_t)_{t \in \mathbb{N}_0}$ and transitions $(K_t)_{t \in \mathbb{N}_0}$

1. Initialization

- Sample $x_0^{(i)}$ from $\eta_0 \forall i$

2. Selection

- Set $\pi^{(i)} \leftarrow g_t(x_t^{(i)}) \forall i$

- For i from 1 to n :

Sample κ from $U[0, 1]$

If $\kappa \leq \epsilon_t \pi^{(i)}$ then

★ Set $\check{x}_t^{(i)} \leftarrow x_t^{(i)}$

Else

★ Set $\check{x}_t^{(i)} \leftarrow x_t^{(j)}$ with probability $\frac{\pi^{(j)}}{\sum_{k=1}^n \pi^{(k)}}$

3. Mutation

- Sample $x_{t+1}^{(i)}$ from $K_t(\check{x}_t^{(i)}, \cdot) \forall i$ and go to step 2
-

The *interacting annealing algorithm (IAA)* describes the approximation by a particle set using equation (3.1.4). The particle system is initialized by n i.i.d. random variables $X_0^{(i)}$ with common law η_0 determining the random probability measure $\eta_0^n := \sum_{i=1}^n \delta_{X_0^{(i)}}/n$. 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

$$\eta_t^n \xrightarrow{\text{Selection}} \check{\eta}_t^n \xrightarrow{\text{Mutation}} \eta_{t+1}^n, \quad (3.1.6)$$

where

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

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 letting each selected particle $\check{X}_t^{(i)}$ evolve according to the Markov transition kernel $K_t(\check{X}_t^{(i)}, \cdot)$.

3.2 Convergence

In this section the asymptotic behavior of the particle approximation model determined by the *IAA* is studied. Del Moral established the following convergence theorem [25, Theorem 7.4.4].

Theorem 3.2.1. *For any $\varphi \in B(E)$,*

$$E \left[\left| \langle \eta_{t+1}^n, \varphi \rangle - \langle \eta_{t+1}, \varphi \rangle \right| \right]_{\eta_0} \leq \frac{2 \operatorname{osc}(\varphi)}{\sqrt{n}} \left(1 + \sum_{s=0}^t r_s \beta(M_s) \right),$$

where

$$r_s := \sup_{x,y \in E} \left(\frac{\prod_{r=s}^t g_r(x)}{\prod_{r=s}^t g_r(y)} \right),$$

$$M_s := K_s K_{s+1} \dots K_t,$$

for $0 \leq s \leq t$. Moreover, $\operatorname{osc}(\varphi) := \sup\{|\varphi(x) - \varphi(y)| ; x, y \in E\}$.

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

$$n \geq \frac{4 \operatorname{osc}(\varphi)^2}{\delta^2} \left(1 + \sum_{s=0}^t r_s \beta(M_s) \right)^2 \quad (3.2.1)$$

needed to achieve a mean error less than a given $\delta > 0$. In order to evaluate the right hand side, we must calculate the Dobrushin contraction coefficient of the Markov kernel M . The coefficient lies in the range 0 to 1, and the more the probability measure $M(x, \cdot)$ “depends” on $x \in E$ the higher the coefficient will be. We will illustrate this property in the following three examples where we always assume that $E = [0, 1]$.

Example 3.2.2. If $M(x, \cdot) := \delta_x$ and $x_1, x_2 \in E$ with $x_1 \neq x_2$, then we get $\sup_{B \in \mathcal{B}(E)} |\delta_{x_1}(B) - \delta_{x_2}(B)| = 1$. This yields $\beta(M) = 1$.

Example 3.2.3. If $M(x, \cdot)$ is independent of x , e.g., if it is equal to the Lebesgue measure λ , we have $\beta(M) = 0$.

Example 3.2.4. Suppose $M := K_s K_{s+1} \dots K_t$, where $(K_k)_{s \leq k \leq t}$ are Markov kernels and $s \leq t$. Furthermore, we assume that there exists for all $s \leq k \leq t$ some $\varepsilon_k \in (0, 1)$ satisfying for all $x_1, x_2 \in E$

$$K_k(x_1, \cdot) \geq \varepsilon_k K_k(x_2, \cdot), \quad (3.2.2)$$

i.e., the mixing condition (2.3.2). Let $x_1, x_2 \in E$ and $B \in \mathcal{B}(E)$. Then we get $|K_k(x_1, B) - K_k(x_2, B)| \leq 1 - \varepsilon_k$ and thus $\beta(M) \leq \prod_{k=s}^t (1 - \varepsilon_k)$.

Note that the right hand side of (3.2.1) is minimized if we are able to choose Markov kernels K_s such that $\beta(M_s)$ is small. However, if we compare the examples, we see that this can be interpreted as if we do not “trust the 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, e.g., the variance is set proportional to the sampling variance of the particles. This can be realized by a dynamic variance scheme as we will explain in Section 3.3.

It is worth to mention two special cases of the selection kernel (3.1.5) that defines the resampling procedure in the interacting annealing algorithm. If $\epsilon_t = 0$ for all t , we get the resampling step of the generic particle filter. The

second special case occurs when we set the parameters $\epsilon_t(\eta_t) := \epsilon'_t / \langle \eta_t, g_t \rangle$, where $0 < \epsilon'_t \leq 1/g$ and

$$g := \sup_{t \in \mathbb{N}_0} \left(\sup_{x, y \in E} \left(\frac{g_t(x)}{g_t(y)} \right) \right) < \infty, \quad (3.2.3)$$

as proposed in [26]. The selection kernel becomes

$$S_{t, \eta_t}(x_t, \cdot) = \epsilon'_t \frac{g_t(x_t)}{\langle \eta_t, g_t \rangle} \delta_{x_t} + \left(1 - \epsilon'_t \frac{g_t(x_t)}{\langle \eta_t, g_t \rangle} \right) \Psi_t(\eta_t). \quad (3.2.4)$$

Note that the necessary condition $\|(\epsilon'_t g_t) / \langle \eta_t, g_t \rangle\|_\infty \leq 1$ is satisfied since $g_t / \langle \eta_t, g_t \rangle \leq g$. If we set the number of particles $n \geq g$, then we can choose $\epsilon'_t = 1/n$. For some random variables $X_t^{(i)}$ and the random probability measure $\eta_t^n = \sum_{j=1}^n \delta_{X_t^{(j)}} / n$, we thus have

$$\epsilon'_t \frac{g_t(X_t^{(i)})}{\langle \eta_t^n, g_t \rangle} = \frac{g_t(X_t^{(i)})}{\sum_{j=1}^n g_t(X_t^{(j)})}.$$

This means that the expression $\epsilon_t \pi^{(i)}$ in Algorithm 2 is replaced by $\pi^{(i)} / \sum_{k=1}^n \pi^{(k)}$.

Pierre del Moral showed in [25, Ch. 9.4] that for any $t \in \mathbb{N}_0$ and $\varphi \in B(E)$ the sequence of random variables

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

converges in law to a Gaussian random variable W when the selection kernel in (3.1.5) is used to approximate the flow (3.1.2). 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 the selection kernel (3.2.4).

3.3 Interacting Simulated Annealing Algorithm

In the preceding section, we discussed how a Feynman-Kac model associated with a pair (g_t, K_t) can be approximated by the IAA without giving details on g_t and on K_t . However, we already introduced unnormalized Boltzmann-Gibbs measures $\exp(-\beta_t V)$ in Examples 3.1.2 and 3.1.3. In statistical mechanics, $V \geq 0$ is interpreted as energy, and $\beta_t \geq 0$ is the inverse temperature. In the following we outline an interacting algorithm that can be regarded as an *interacting simulated annealing algorithm (ISA)*. This means that the distribution of global minima of V is approximated when β_t tends to infinity.

We suppose that K is a Markov kernel satisfying the mixing condition (3.2.2) for an $\varepsilon \in (0, 1)$ and $\text{osc}(V) < \infty$. A time mesh is defined by

$$t(n) := n(1 + \lfloor c(\varepsilon) \rfloor) \quad c(\varepsilon) := (1 - \ln(\varepsilon/2))/\varepsilon^2 \quad \text{for } n \in \mathbb{N}_0. \quad (3.3.1)$$

Let $0 \leq \beta_0 \leq \beta_1 \dots$ be an annealing scheme such that $\beta_t = \beta_{t(n+1)}$ is constant in the interval $(t(n), t(n+1)]$. Furthermore, we denote by $\check{\eta}_t$ the Feynman-Kac distribution after the selection step, i.e., $\check{\eta}_t = \Psi_t(\eta_t)$. According to [25, Proposition 6.3.2], cf. also [29], we have

Theorem 3.3.1. *Let $b \in (0, 1)$ and $\beta_{t(n+1)} = (n+1)^b$. Then for each $\delta > 0$*

$$\lim_{n \rightarrow \infty} \check{\eta}_{t(n)}(V \geq V_* + \delta) = 0,$$

where $V_* = \sup\{v \geq 0; V \geq v \text{ a.e.}\}$.

The rate of convergence is $d/n^{(1-b)}$ where d depends on b and $c(\varepsilon)$ as given in [25, Theorem 6.3.1]. This theorem establishes that the flow of the Feynman-Kac distribution $\check{\eta}_t$ becomes concentrated in the region of global minima as $t \rightarrow +\infty$. The flow can be approximated by the interacting annealing algorithm with $g_t = \exp(-\beta_t V)$ and $K_t = K$.

The mixing condition is not only essential for the convergence result but also influences the time mesh by the parameter ε . In view of equation (3.3.1), kernels with ε close to 1 are preferable, e.g., Gaussian kernels on a bounded set with a very high variance as discussed in Section 2.3. However, we cannot sample from the measure $\check{\eta}_t$ directly, instead we approximate it by n particles. Now the following problem arises. On one hand the mass of the measure concentrates on a small region of E , and on the other hand the particles are spread over E if ε is large. As a result we get a degenerated system where the weights of most of the particles are zero and thus the global minima are estimated inaccurately, particularly for small n . If we choose a kernel with small ε in contrast, the convergence rate is very slow. Since neither of them is suitable for applications, we suggest a dynamic variance scheme instead of a fixed kernel K as already mentioned in Section 3.2.

Let K_t be a family of Gaussian kernels on a bounded set E with covariance matrices Σ_t proportional to the sample covariance after the ‘‘Resampling’’ step. That is, for a constant $c > 0$,

$$\Sigma_t := \frac{c}{n-1} \sum_{i=1}^n (x_t^{(i)} - \mu_t)_\rho (x_t^{(i)} - \mu_t)_\rho^T, \quad \mu_t := \frac{1}{n} \sum_{i=1}^n x_t^{(i)}, \quad (3.3.2)$$

where $((x)_\rho)_k = \max(x_k, \rho)$ for a $\rho > 0$. The value ρ ensures that the variance does not become zero. The elements off the diagonal are usually set to zero, in order to reduce computation time.

We remark that the *APF* is a particle filter where the “Updating” and “Resampling” steps are replaced by the interacting simulated annealing algorithm with $\epsilon_t = 0$. The algorithm is illustrated similarly as in (2.2.2) by

$$\eta_t^n \xrightarrow{\text{Prediction}} \hat{\eta}_{t+1}^n \xrightarrow{\text{ISA}} \eta_{t+1}^n. \quad (3.3.3)$$

The *ISA* is initialized by the predicted particles $\hat{X}_{t+1}^{(i)}$ and performs M times the selection and mutation steps. Afterwards the particles $X_{t+1}^{(i)}$ are obtained by an additional selection. This shows that the annealed particle filter uses a simulated annealing principle to locate the global minimum of a function V at each time step. Hence, it is suitable for applications like motion capturing as illustrated in Fig. 1.2 and demonstrated in Section 4. However, it also reveals that the conditional distribution (2.2.1) is no longer approximated in the way of the generic particle filter, and therefore the arguments in Section 2 cannot be applied without modifications. In the next section, we present a model that approximates a given distribution by the interacting annealing algorithm.

3.4 Interacting Annealed Sampling Algorithm

Our method is based on annealed importance sampling [30] that allows additional interaction of the particles during the steps. Let us consider a finite sequence of Boltzmann-Gibbs measures

$$\mu_t(dx) := \frac{1}{\langle \mu_0, \exp(-\beta_t V) \rangle} \exp(-\beta_t V(x)) \mu_0(dx) \quad (3.4.1)$$

according to some schedule $0 = \beta_0 < \beta_1 < \dots < \beta_{T-1} < \beta_T = 1$, where $\mu_0 \in \mathcal{P}(E)$. In contrast to simulated annealing and *ISA* that converge to the global minima of V , annealed importance sampling approximates the distribution μ_T .

We use a Feynman-Kac model associated with a pair (g_t, K_t) as introduced in Section 3.1 to describe the mathematical framework. For any $0 \leq t < T$, we define

$$g_t(x_t) := \frac{\langle \mu_0, \exp(-\beta_t V) \rangle}{\langle \mu_0, \exp(-\beta_{t+1} V) \rangle} \exp(-(\beta_{t+1} - \beta_t)V(x_t)). \quad (3.4.2)$$

The Markov kernels $(K_t)_{0 \leq t < T}$ are chosen such that K_t leaves the measure μ_{t+1} invariant, i.e.,

$$\mu_{t+1}(B) = \int_E K_t(x_t, B) \mu_{t+1}(dx_t), \quad (3.4.3)$$

for all $B \in \mathcal{B}(E)$. Metropolis-Hastings updates [23, 14, 31], for instance, are suitable choices for Markov transitions that leave a measure invariant. The following lemma reveals that $(\mu_t)_{0 \leq t \leq T}$ are the Feynman-Kac distributions associated with the pair (g_t, K_t) .

Lemma 3.4.1. *For any $0 \leq t \leq T$, we have*

$$E \left[\varphi(X_t) \prod_{s=0}^{t-1} g_s(X_s) \right]_{\mu_0} = \langle \mu_t, \varphi \rangle \quad \forall \varphi \in B(E).$$

Proof: Let $\varphi \in B(E)$. From (3.4.2) and (3.4.3) we obtain

$$\begin{aligned} & E \left[\varphi(X_t) \prod_{s=0}^{t-1} g_s(X_s) \right]_{\mu_0} \\ &= \int_E \int_E \dots \int_E \varphi(x_t) \left(\prod_{s=0}^{t-1} K_s(x_s, dx_{s+1}) g_s(x_s) \right) \mu_0(dy_0) \\ &= \int_E \dots \int_E \varphi(x_t) \left(\prod_{s=1}^{t-1} K_s(x_s, dx_{s+1}) g_s(x_s) \right) \int_E K_0(x_0, dx_1) \mu_1(dx_0) \\ &= \int_E \dots \int_E \varphi(x_t) \left(\prod_{s=1}^{t-1} K_s(x_s, dx_{s+1}) g_s(x_s) \right) \mu_1(dx_1) \\ &\quad \vdots \\ &= \int_E \varphi(x_t) \mu_t(dx_t). \end{aligned}$$

□

Note that the constant term of g_t in Equation (3.4.2) is unimportant for the algorithm since it is compensated by ε_t of the selection kernel (3.1.5) and by the normalization factor of the Boltzmann-Gibbs transformation (3.1.3). The resulting interacting algorithm can be regarded as an *interacting annealing sampling algorithm (IAS)* that converges to μ_T according to Theorem 3.2.1.

In the context of filtering μ_0 is the predicted conditional distribution, $\exp(-V)$ is the likelihood function, and μ_T is the posterior distribution approximated by the weighted particle set. Hence, it would be desirable to combine *IAS* with the generic particle filter as suggested in [8, Ch. 7]. However, we must pay attention to the crucial assumption that the transitions K_t leave the measures μ_{t+1} invariant. This means that the transitions depend on μ_0 and thus on the unknown signal. On account of this limitation of the

IAS, we believe that the *ISA* is more relevant for applications, particularly for motion capturing. We will therefore restrict the evaluation in Section 4 to the *ISA*.

Another important consequence of this result is that the annealed particle filter does not approximate the conditional distribution (2.2.1), since it diffuses the particles by kernels that do not satisfy (3.4.3). Hence, the *APF* is not suitable for filtering problems as shown in Fig. 1.1.

4 Evaluation

In Section 3.3, we observed that the *APF* uses *ISA* for each time step and thus performs well for motion capturing. For an exhaustive experimental evaluation, we track an articulated arm with less DOF than in the example given in Fig. 1.2. The aim of this section is not to find “the best” parameters since these depend on the specific application. Rather, we reveal the general impact of the parameters on the performance using an experimental setting that is typical for human motion capturing. The evaluation results provide a general guidance and a good starting point for finding the optimal setting for a particular application.

Furthermore, we compare the two selection kernels discussed in Section 3.2. The *ISA* with $\epsilon_t = 0$ (3.1.5) is denoted by ISA_0 and with $\epsilon'_t = 1/n$ (3.2.4) by $ISA_{1/n}$. In Section 4.2, we demonstrate the influence of the mixing condition that is essential for the convergence of the *ISA* (Theorem 3.3.1). Finally, the filtering example illustrated in Fig. 1.1 is discussed in detail.

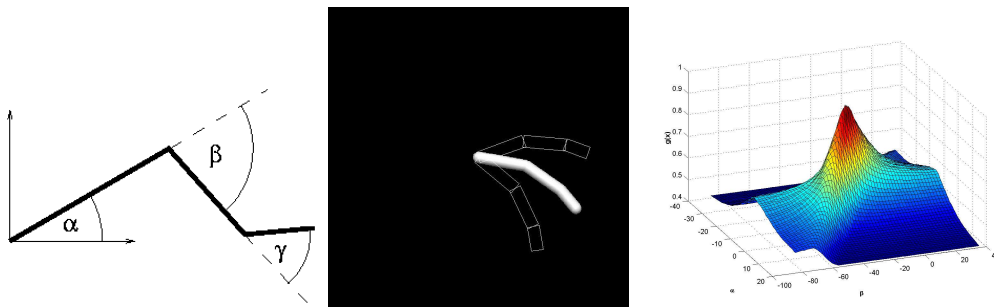


Figure 4.1: **Left:** Pose of the arm is described by the vector $x = (\alpha, \beta, \gamma)^T$. **Center:** Varying α of the template. **Right:** Graph of g_t over (α, β) .

4.1 Motion Capturing

Experimental Set-up and Implementation Details The arm consists of three limbs and three joints. The position of the arm is described by $x^T = (\alpha, \beta, \gamma) \in E$, where $E := [-170, 170] \times [-125, 125] \times [-125, 125]$ as depicted in Fig. 4.1. For evaluating, a sequence of 201 synthetic images is generated. X_0 is uniformly distributed in E yielding an unknown arm position at the beginning. The angles α_{t+1} , β_{t+1} , and γ_{t+1} are sampled from Gaussian distributions on E with mean α_t , β_t , and γ_t and variance $\sigma_\alpha = 20$, $\sigma_\beta = 40$, and $\sigma_\gamma = 30$, respectively. This sequence (Seq_1) is difficult for tracking since the velocity and the direction of the movement may change from frame to frame. In a second sequence (Seq_2), the arm moves from position $(-30, -80, -40)^T$ to $(50, 30, 20)^T$ and back with constant speed as illustrated in Fig. 4.2. Moreover, we added some Gaussian noise to each position vector.

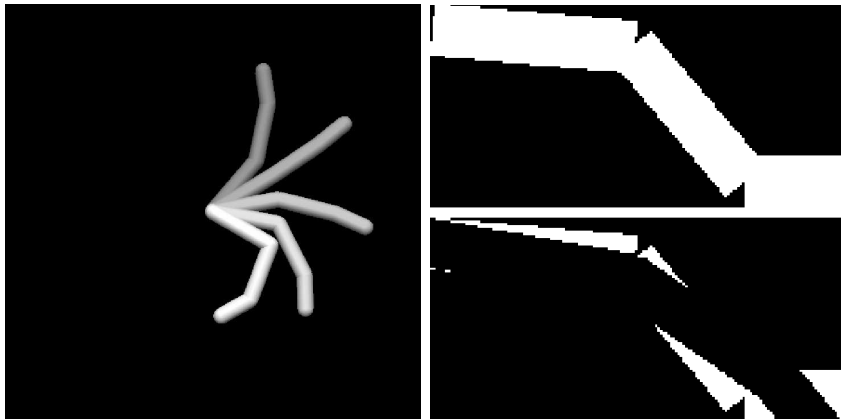


Figure 4.2: **Left:** Motion sequence Seq_2 . **Right:** Template's silhouette (*top*). Error map (*bottom*).

For calculating the weighting functions g_t , the image is converted to a binary image by thresholding. This image is compared with the silhouette of each arm template that is determined by a particle $x_t^{(i)}$ as shown in Fig. 4.2. An error map is obtained by a pixelwise AND operation between the inverted binary image and the template's silhouette. The weighting functions are then calculated by $g_t := \exp(-N_e/N_p)$, where N_p denotes the number of pixels of the template's silhouette and N_e the sum of the pixel values in the error map. The graph of the weighting function is plotted in Fig. 4.1. We observed in our experiments that $g \approx 40$ (3.2.3). This means that the selection kernel (3.2.4) is valid if the number of particles is greater than 40.

In the following, we evaluate the performance of the ISA_0 and $ISA_{1/n}$ in combination with different annealing schemes, variance schemes, number of annealing runs, and number of particles. The simulations for Seq_1 and Seq_2 were repeated 50 and 40 times, respectively. The error of an estimate $\sum_i \pi^{(i)} x_t^{(i)}$ was measured by $1 - g_t$. The averages of the mean square errors (MSE) for each sequence indicate the performance.

Since in real world applications the measurements are noisy caused by clutter, film grain, bad lighting conditions, CCD camera noise, etc., we also added strong noise to the weighting functions by $\exp(-\vartheta(N_e + W_t^{(i)})/N_p)$, where $\vartheta(N) = \max(0, \min(N, N_p))$ and $W_t^{(i)}$ are independent zero-mean Gaussian random variables with variance 40000. For comparison, $N_p \approx 4000$.

GPF vs. ISA We assumed that the dynamics for Seq_1 were known. Hence, the algorithms were initialized by the uniform distribution on E and the prediction step (3.3.3) was performed according to the Gaussian transitions used for the arm simulation. By contrast, we did not use the dynamical model for tracking Seq_2 . The initial distribution was instead the uniform distribution on $[-20, -40] \times [-60, -100] \times [-20, -60] \subset E$ and the transitions kernels were the same as for Seq_1 . In order to provide a fair comparison between GPF with n_T particles and ISA with various annealing schemes, the number of particles is given by $n = \lfloor n_T/T \rfloor$ where T denotes the number of annealing runs. The GPF with $n_T = 250$ produced a MSE of **0.04386** for Seq_1 and **0.04481** for the noisy sequence. Seq_2 was tracked with 225 particles and MSE of **0.01099** and **0.01157**, respectively.

Annealing Schemes We evaluated the performance of various annealing schemes $0 \leq \beta_0 \leq \dots \leq \beta_{T-1}$ with fixed length $T = 5$. While the particles were diffused between the annealing steps for Seq_1 by Gaussian kernels with $\sigma_\alpha = 20$, $\sigma_\beta = 40$, and $\sigma_\gamma = 30$, we set $\sigma_\alpha = \sigma_\beta = \sigma_\gamma = 5$ for Seq_2 . In Fig. 4.3, the MSE for the annealing schemes with decreasing increments

$$\begin{aligned} \beta_t &= \alpha (1 - c^{-(t+1)}) && (\textit{geometric}), \\ \beta_t &= \alpha \ln(t+c)/\ln(T+c-1) && (\textit{logarithmic}), \\ \beta_t &= \alpha ((t+1)/T)^c && (\textit{polynomial}) \end{aligned}$$

are given. The schemes are normalized such that $\beta_{T-1} = \alpha = 4$. When c tends to infinity or to 0 in the case of a polynomial scheme, $\beta_t \rightarrow \alpha$ for all $0 \leq t < T$.

The diagrams show that the geometric annealing schemes are unstable in the sense that the curves of the MSE with respect to c contain many local optima, particularly for Seq_1 . It makes the optimization of the scheme for

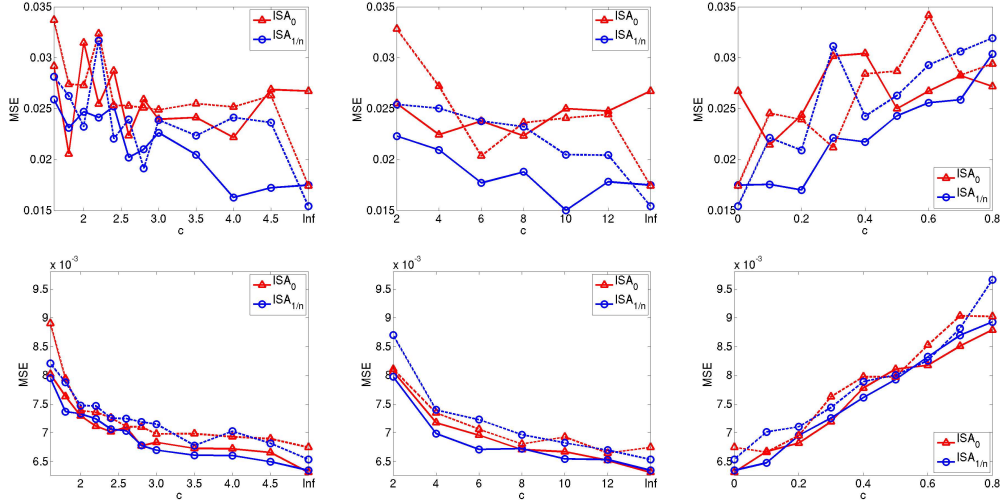


Figure 4.3: Performance for different annealing schemes with $T = 5$. Average of the MSE for the sequences Seq_1 (top) and Seq_2 (bottom) with noisy measurements (dashed) and without noise (solid). **Left:** $\beta_t = \alpha(1 - c^{-(t+1)})$. **Center:** $\beta_t = \alpha \ln(t+c)/\ln(T+c-1)$. **Right:** $\beta_t = \alpha((t+1)/T)^c$. **Top:** The curves for the geometric annealing schemes are unstable and the best result is obtained by $ISA_{1/n}$ with a logarithmic scheme. **Bottom:** The error decreases when $\beta_t \rightarrow \alpha$. The impact of the selection kernel and noise is small.

β_t	ISA_0	$ISA_{1/n}$	ISA_0	$ISA_{1/n}$
	Seq_1		Seq_1 with noise	
$\alpha(t+1)/T$	0.03634	0.03029	0.03220	0.02809
$\alpha 1.2^{t+1-T}$	0.02819	0.02302	0.03185	0.02609
$\alpha 1.8^{t+1-T}$	0.04214	0.05128	0.03891	0.04452
	Seq_2		Seq_2 with noise	
$\alpha(t+1)/T$	0.01006	0.00948	0.00988	0.01026
$\alpha 1.2^{(t+1-T)}$	0.00818	0.00805	0.00827	0.00858
$\alpha 1.8^{(t+1-T)}$	0.01514	0.01501	0.01557	0.01543

Table 4.1: MSE error for annealing schemes with constant and increasing increments ($T = 5$). The schemes are outperformed by the annealing schemes given in Fig. 4.3.

a particular application quite difficult. The logarithmic schemes performed best where the lowest MSE for Seq_1 , namely 0.01501, was achieved by an $ISA_{1/n}$ with $c = 10$. In comparison, the errors for Seq_2 are significant lower and the scheme with $\beta_t = \alpha$ performs best since the motion is simple and local maxima rarely occur. Furthermore, the difference between the two selection kernels is small. The impact of noise on the results is also minor when the dynamics are simple in contrast to the more difficult sequence. The observation that the error for Seq_1 with noise significantly declines as c goes to infinity indicates that the other parameters are not well chosen for this noisy sequence. Providing some results for schemes with constant or increasing increments in Table 4.1 reveals that these schemes are outperformed by the schemes given in Fig. 4.3. We use henceforth a polynomial annealing scheme with $c = 0.1$ since both ISA_0 and $ISA_{1/n}$ performed well for the scheme.

$(\sigma_\alpha^2 \sigma_\beta^2 \sigma_\gamma^2)$	ISA_0	$ISA_{1/n}$	ISA_0	$ISA_{1/n}$
	Seq_1		Seq_1 with noise	
(15 35 25)	0.02527	0.01985	0.02787	0.02573
(20 40 30)	0.02145	0.01756	0.02453	0.02213
(25 45 35)	0.02341	0.02011	0.02506	0.02357
(15 40 35)	0.02238	0.01891	0.02035	0.02510
(25 40 25)	0.02240	0.01905	0.02622	0.02345
	Seq_2		Seq_2 with noise	
(0.5 0.5 0.5)	0.00637	0.00631	0.00643	0.00664
(2 2 2)	0.00612	0.00627	0.00639	0.00652
(5 5 5)	0.00668	0.00648	0.00666	0.00702
(0.5 2 5)	0.00611	0.00626	0.00643	0.00629
(5 2 0.5)	0.00661	0.00674	0.00674	0.00695

Table 4.2: MSE error for constant variance schemes. The decreasing schemes perform better (Tables 4.3 and 4.4).

Variance Schemes During the mutation step of the ISA , the particles were diffused according to a Gaussian distribution where the variance for each annealing step is defined by a variance scheme. The errors for constant schemes are given in Table 4.2, for deterministic schemes in Tables 4.3 and 4.4, and for dynamic schemes (3.3.2) in Fig. 4.4. The first column of Tables 4.3 and 4.4 contains the reference variance that is reduced for each annealing step by the decreasing scheme given in the second column. We give three examples where $\iota \in \{\alpha, \beta, \gamma\}$: $(-d_\alpha - d_\beta - d_\gamma)$ means that $\sigma_{\iota,t}^2 = \sigma_{\iota,t-1}^2 - d_\iota$.

$(\sigma_\alpha^2 \sigma_\beta^2 \sigma_\gamma^2)$	Decreasing scheme	ISA_0	$ISA_{1/n}$	ISA_0	$ISA_{1/n}$
		Seq_1		Seq_1 with noise	
(32 49 36)	(-4 - 3 - 2)	0.01997	0.01920	0.02437	0.02335
(32 58 54)	(-4 - 6 - 8)	0.02243	0.02485	0.02480	0.02093
(32 70 54)	(-4 - 10 - 8)	0.02048	0.02066	0.02332	0.02411
(32 52 42)	(-4 - 4 - 4)	0.02193	0.01919	0.02489	0.01795
(29 52 45)	(-3 - 4 - 5)	0.01989	0.01666	0.02029	0.02074
(23 47 35)	$\times \frac{\beta_3}{\alpha} \frac{\beta_2}{\alpha} \frac{\beta_1}{\alpha} \frac{\beta_0}{\alpha}$	0.02230	0.01950	0.02654	0.02203
(27 47 37)	$-0.1.5.1.5^2.1.5^3$	0.02187	0.02324	0.01807	0.02328
(27 47 37)	$-0.1.5^3.1.5^2.1.5$	0.02048	0.02219	0.02398	0.02109
(48 97 73)	$\times 0.8.0.8^2.0.8^3.0.8^4$	0.02140	0.02030	0.02099	0.02326
(30 60 45)	$\times 0.9.0.9^2.0.9^3.0.9^4$	0.01907	0.01690	0.02470	0.02142

Table 4.3: MSE error for deterministic variance schemes. The schemes are outperformed by dynamic variance schemes (Fig. 4.4).

$(\sigma_\alpha^2 \sigma_\beta^2 \sigma_\gamma^2)$	Decreasing scheme	ISA_0	$ISA_{1/n}$	ISA_0	$ISA_{1/n}$
		Seq_2		Seq_2 with noise	
(3.5 5 8)	(-1 - 1 - 1)	0.00619	0.00632	0.00635	0.00629
(5 5 5)	(-1.5 - 1.5 - 1.5)	0.00614	0.00623	0.00640	0.00656
(3.5 5 6.5)	(-1 - 1.5 - 2)	0.00606	0.00626	0.00641	0.00642
(6.5 5 3.5)	(-2 - 1.5 - 1)	0.00648	0.00654	0.00651	0.00656
(7.5 7.5 7.5)	$-0.1.5.1.5^2.1.5^3$	0.00649	0.00657	0.00662	0.00662
(7.5 7.5 7.5)	$-0.1.5^3.1.5^2.1.5$	0.00636	0.00638	0.00646	0.00657
(1.2 1.2 1.2)	$\times 0.8.0.8^2.0.8^3.0.8^4$	0.00622	0.00623	0.00649	0.00639
(.75 .75 .75)	$\times 0.9.0.9^2.0.9^3.0.9^4$	0.00631	0.00607	0.00636	0.00641

Table 4.4: MSE error for deterministic variance schemes. The best dynamic variance schemes (Fig. 4.4) perform as well as the best deterministic variance schemes.

The decreasing scheme $-0 d^1 d^2 d^3$ gives the variance scheme $\sigma_{t,t}^2 = \sigma_{t,t-1}^2 - d^t$. The scheme $\sigma_{t,t}^2 = d^{t+1} \sigma_t^2$ is denoted by $\times d^1 d^2 d^3 d^4$.

The dynamic variance schemes are not only easier to handle since they depend only on one parameter c , but they also outperform the deterministic schemes provided that an appropriate parameter c is chosen. The best result for Seq_1 with MSE 0.01175 was obtained by $ISA_{1/n}$ with parameter $c = 0.3$. In comparison to the GPF , the MSE was reduced by more than 73%. We see that the error for Seq_2 was not significantly improved when comparing the best settings for constant, deterministic, and dynamic schemes. It indicates that the flow of Feynman-Kac distributions locates the global minimum and that the error is mainly caused by the particle approximation. Hence, an improvement is only expected by reducing the number of annealing runs yielding more particles for approximation or by increasing n_T .

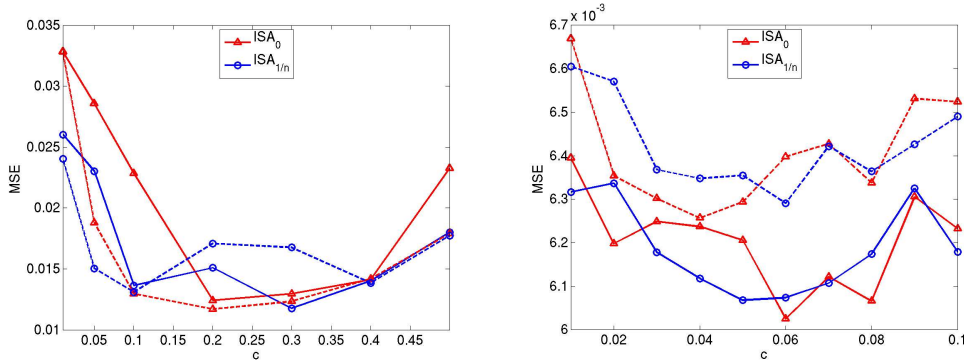


Figure 4.4: Performance for dynamic variance schemes with different values of c in the presence of noise (*dashed*) and without noise (*solid*). **Left:** MSE for Seq_1 . The error is significantly reduced in comparison to deterministic schemes (Tables 4.2 and 4.3). The best result is obtained by $ISA_{1/n}$ with $c = 0.3$. **Right:** MSE for Seq_2 . The best dynamic variance schemes perform as well as the best deterministic variance schemes (Tables 4.2 and 4.4).

Number of Annealing Runs and Particles The influence of the number of annealing runs for different values of n_T is plotted in Figs. 4.5 and 4.6. Seq_1 was tracked by ISA_0 and $ISA_{1/n}$ with a dynamic scheme with $c = 0.2$ and $c = 0.3$, respectively. The parameters for Seq_2 were 0.06 and 0.05, respectively. The curves for $ISA_{1/n}$ are quite stable with an unique optimal parameter $T = 6$ independent of n_T and noise, see Fig. 4.5. By contrast, the curves for ISA_0 contain deep local minima, in particular when the sequence

was disturbed by noise. Moreover, one can observe at $T = 7$ that the error for $ISA_{1/n}$ increases significantly when the number of particles is not clearly greater than g (3.2.3). This shows the impact of the condition on the results. The MSEs for Seq_2 are given in the diagram on the left hand side of Fig. 4.6. The error was reduced by reducing the number of annealing runs and by increasing n_T as expected whereas the differences between ISA_0 and $ISA_{1/n}$ were minimal. It also demonstrates the robustness of ISA to noise. As comparison, the error of the GPF is hardly reduced by increasing n_T . The MSE was still above 0.043 and 0.01 for Seq_1 and Seq_2 , respectively.

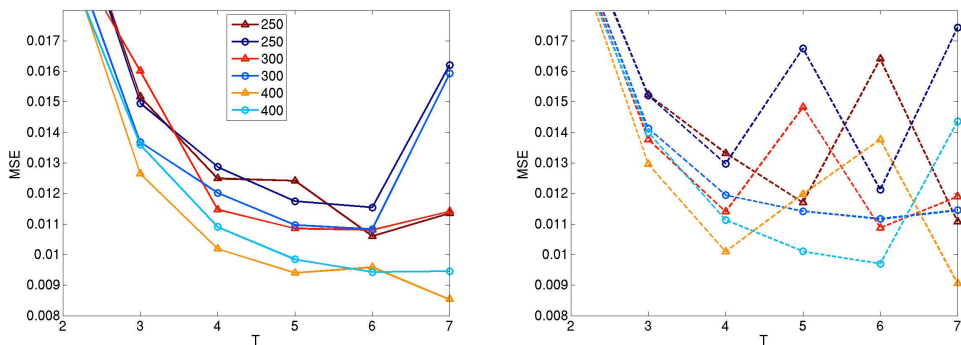


Figure 4.5: Performance of ISA_0 (*triangles*) and $ISA_{1/n}$ (*circles*) for different numbers of annealing runs T with $n_T = 250, 300,$ and 400 . The curves for $ISA_{1/n}$ are more stable with an unique optimal parameter $T = 6$, but the error increases at $T = 7$. More annealing runs are required than for Seq_2 (Fig. 4.6). **Left:** MSE for Seq_1 without noise. **Right:** MSE for Seq_1 with noise.

Real Sequences We applied ISA_0 and $ISA_{1/n}$ also to human motion capturing as visualized in Fig. 4.7. The diagram on the right hand side contains the estimated angles of the left and the right knee where the values acquired from the marker based system provide a ground truth. For the experiments that are described in [9], 250 particles and a geometric annealing scheme with $T = 11$ were used. We compared the root mean square errors (RMSE) for both knees obtained by ISA_0 with $c = 0.1$ for the dynamic scheme and $ISA_{1/n}$ with $c = 0.12$, where we repeated the simulations 25 times. While the average of the RMSE was not significantly improved by $ISA_{1/n}$ as expected from the previous results, the variance was reduced by 19.8% compared to ISA_0 .

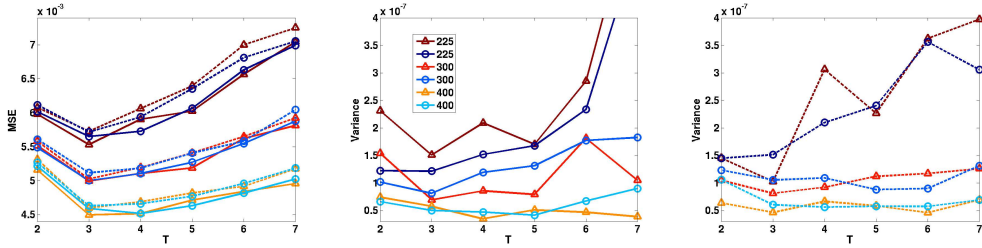


Figure 4.6: Performance of ISA_0 (triangles) and $ISA_{1/n}$ (circles) for different numbers of annealing runs T with $n_T = 225, 300,$ and 400 . **Left:** MSE for Seq_2 with noisy measurements (dashed) and without noise (solid). The error decreases with increasing n_T whereas the differences between ISA_0 and $ISA_{1/n}$ are minimal. The error is only slightly affected by noise. **Center:** Variance of MSE for Seq_2 without noise. The variance also decreases with increasing n_T . The curves for $ISA_{1/n}$ are more stable. **Right:** Variance with noise.

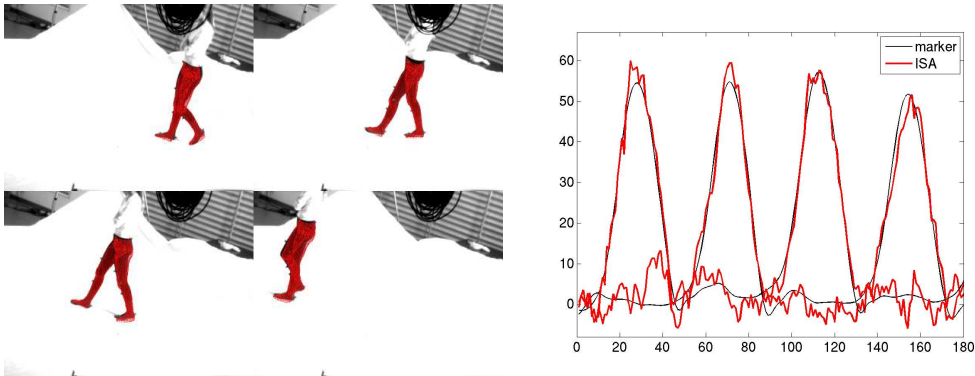


Figure 4.7: Tracking the lower part of a human body during walking. **Left:** The estimates (projected mesh) by the APF using a 3D model with 18 DOF (4 of 180 frames). **Right:** A comparison of the estimated joint angles of the right and left knee with a marker based motion capture system reveals the accuracy of $ISA_{1/n}$.

When we compare the result with our arm example in Fig. 4.6, we find no evidence that the variance reduction can be generalized. While the variance of the error is significantly lower for $ISA_{1/n}$ with $n_T = 225$, ISA_0 performs better with $n_T = 300$, and the differences are marginal for $n_T = 400$. The diagrams, however, reveal that the curves for $ISA_{1/n}$ are more stable and the variances are reduced by increasing n_T .

4.2 Mixing Condition

In this Section, we illustrate the impact of the mixing condition that is essential for the convergence results given in Sections 2.3 and 3.3. For this purpose, we track a stiff arm, i.e., $x = \alpha$. We suppose that the arm movement is given by the process $X_t := X_{t-1} + V_t$, where $X_0 := 0$ and V_t are i.i.d. uniform random variables on $[-10, 10]$. Let us examine the events where $V_t \in [9.75, 10]$ for $1 \leq t \leq 400$. Even though the probability that this occurs is very small, it is strictly greater than zero.

For the simulations, we used an *APF* with ISA_0 and parameters $n = 100$, $T = 2$, $\beta_0 = 3.2$. The initial distribution was δ_0 and the mutation kernels $K_t(x, \cdot)$ were uniform distributions on $[x - 2, x + 2]$. When uniform kernels were chosen for prediction in accordance with the process X_t , the *APF* was not capable of tracking the articulated arm as shown in Fig. 4.8. The algorithm lost track of the arm after some time and was not able to recover afterwards. For comparison, the uniform kernels were replaced by Gaussian kernels with variance 100, which satisfy the mixing condition since the state space is bounded. In this case, the arm was successfully tracked over a sequence of 400 images, see Fig. 4.9. We carried out the simulations 25 times. This shows that the *APF* may fail when the mixing condition is not satisfied, even though the particles are correctly predicted according to the dynamics.

4.3 Filtering Problem

We already mentioned that the *GPF* outperforms the *APF* for the filtering problem since the latter does not approximate the posterior distribution. An example is illustrated in Fig. 1.1, where we applied the algorithms to a one-dimensional nonlinear filtering problem. The signal and observation process are defined by

$$X_t = \frac{X_{t-1}}{4} + 5 \frac{X_{t-1}}{1 + X_{t-1}^2} + 2 \cos(1.2t) + V_t, \quad Y_t = \frac{X_t^2}{20} + \frac{X_t^3}{100} + W_t, \quad (4.3.1)$$

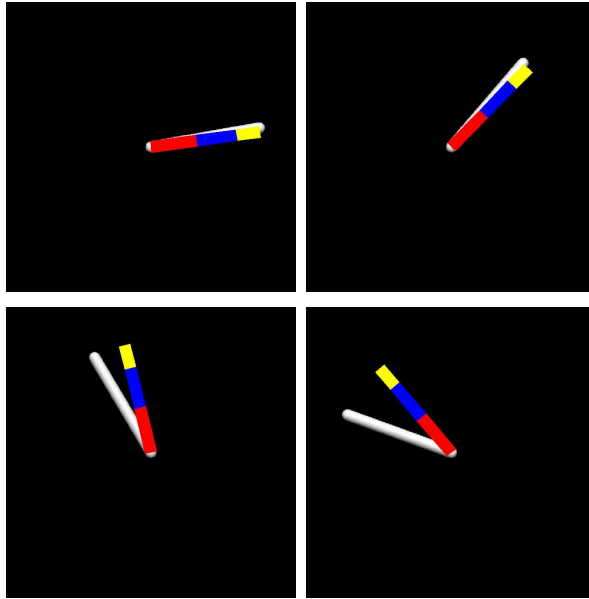


Figure 4.8: When the mixing condition is not satisfied, the *APF* loses track of the articulated arm after some time and is not able to recover. **From top left to bottom right:** $t = 1, 5, 158, 165$.

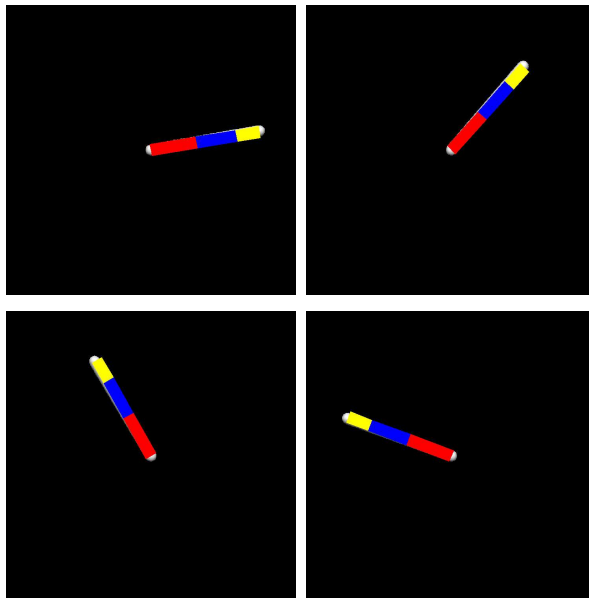


Figure 4.9: When the mixing condition is satisfied, the *APF* is able to track the articulated arm. **From top left to bottom right:** $t = 1, 5, 158, 165$.

where V_t and W_t are independent zero-mean Gaussian random variables with variances 10 and 1, respectively. The distribution of X_0 is a standard normal distribution. This example is similar to the studied problem in [12], where the extended Kalman filter performs poorly.

We evaluated the *APF* with various parameter settings as in the arm example and repeated each simulation with 200 time steps 100 times. The performance was measured by the resulting root mean square error from the true signal, where $n_T = 300$ was fixed. The best result of the *APF* is plotted in the diagram of Fig. 1.1 with RMSE of 2.7988. For comparison, the error of *GPF* was only 2.6037.

5 Conclusions

We have proposed two algorithms, namely interacting simulated annealing (*ISA*) and interacting annealing sampling (*IAS*), which combine interacting and annealing strategies. Based on Feynman-Kac models, we provided convergence results and conditions that are sufficient for convergence. While *ISA* converges to the regions of global minima, *IAS* approximates a given distribution.

We showed that the annealed particle filter (*APF*), which performs an *ISA* for each time step, does not solve the filtering problem since *ISA* does not approximate the posterior distribution in contrast to *IAS*. This was confirmed by an example where the generic particle filter (*GPF*) outperformed the *APF*. For a tracking application, however, the models for a filtering problem are often unknown whereas a fitness function can be easily designed from the available image features. In this case, *ISA* determines the global optimum of the fitness function, which leads to a good performance of the *APF* in contrast to *GPF* as we have demonstrated in our experiments.

The *ISA* approximates a flow of Feynman-Kac distributions that converges to the regions of global minima. The optimal parameters are therefore a trade-off between the approximation of the global minima by the flow and the approximation of the flow by particles. The first consequence of this are the dynamic variance schemes that outperform constant and deterministic variance schemes as we have shown. It also influences the optimal number of annealing runs provided that n_T is fixed. When the global optimum is easily determined by the flow, the error is mainly introduced by the particle approximation. Increasing the number of particles n by reducing the number of annealing runs improves the performance. More annealing runs, on the contrary, provide a better localization of the global optimum by the flow.

Based on two versions of the selection kernel, we compared the algorithms ISA_0 and $ISA_{1/n}$ where the latter gives a better convergence result of the particle approximation if $n > g$ is satisfied. In our experiments, however, we found no evidence that one kernel is better than the other. From the

practitioner’s point of view, the kernel can be selected as follows: When the number of particles is clearly greater than g , we recommend to use $ISA_{1/n}$ for finding the optimal setting since the error curves were more stable with respect to the parameters. Afterwards, it is useful to apply the ISA_0 with the final setting as it cannot be guaranteed for very complex weighting functions that the chosen parameters are optimal. The kernel with the best results is then selected.

Furthermore, we demonstrated in our experiments that the error declines by increasing n_T and that the ISA is robust to noise. Since the piecewise constant annealing scheme given in Theorem 3.3.1, which is sufficient for convergence, is too slow for most applications, we compared various annealing schemes where the logarithmic schemes performed best. Although we suspect that these schemes work well also for more complex applications, 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 experiments for a wide range of applications would be necessary.

Finally, we gave an example that illustrates the impact of the mixing condition on the APF , which is essential for the uniform convergence of the GPF and the convergence of the ISA .

Bibliography

- [1] D. Alspach and H. Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Trans. on Automatic Control*, 17(4):439–448, 1972.
- [2] P. Chigansky and R. Liptser. Stability of nonlinear filters in nonmixing case. *The Annals of Applied Probability*, 14(4):2038–2056, 2004.
- [3] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. on Signal Processing*, 50(3):736–746, 2002.
- [4] D. Crisan and M. 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.
- [5] D. Crisan, P. D. Moral, and T. Lyons. Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields*, 5(3):293–319, 1999.
- [6] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 1144–1149, 2000.
- [7] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *Int. J. of Computer Vision*, 61(2):185–205, 2005.
- [8] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.
- [9] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Learning for multi-view 3d tracking in the context of particle filters. In *Int. Sympo-*

- sium on Visual Computing (ISVC)*, Lecture Notes in Computer Science. Springer, 2006. To appear.
- [10] B. Gidas. *Topics in Contemporary Probability and Its Applications*, chapter 7: Metropolis-type Monte Carlo Simulation Algorithms and Simulated Annealing, pages 159–232. CRC Press, Boca Raton, 1995.
 - [11] F. L. Gland and N. Oudjane. Stability and uniform approximation of nonlinear filters using the hilbert metric and application to particle filters. *The Annals of Applied Probability*, 14(1):144–187, 2004.
 - [12] N. Gordon, D. Salmond, and A. Smith. Novel approach to non-linear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.
 - [13] J. Hammersley and D. Handscomb. *Monte Carlo Methods*. Methuen, London, 1967.
 - [14] W. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
 - [15] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. on Computer Vision*, volume 1, pages 343–356, 1996.
 - [16] A. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, London, 1970.
 - [17] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. Symposium on Aerospace/Defence Sensing, Simulation and Controls*, 1997.
 - [18] R. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
 - [19] K. Kanazawa, D. Koller, and S. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proc. of the Eleventh Annual Conf. on Uncertainty in AI (UAI)*, pages 346–351, 1995.
 - [20] S. Kirkpatrick, C. G. Jr., and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

- [21] G. Kitagawa and W. Gersch. *Smoothness Priors Analysis of Time Series*, volume 116 of *Lecture Notes in Statistics*. Springer, New York, 1996.
- [22] J. MacCormick. *Probabilistic models and stochastic algorithms for visual tracking*. PhD thesis, University of Oxford, 2000.
- [23] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. of Chemical Physics*, 21(6):1087–1092, 1953.
- [24] P. D. Moral. Measure-valued processes and interacting particle systems. application to nonlinear filtering problems. *Annals of Applied Probability*, 8(2):438–495, 1998.
- [25] P. D. Moral. *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Springer, New York, 2004.
- [26] P. D. Moral and A. Doucet. On a class of genealogical and interacting metropolis models. In *Séminaire de Probabilités XXXVII*, number 1832 in *Lecture Notes in Mathematics*. Springer, 2003.
- [27] P. D. Moral and A. 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):155–194, 2001.
- [28] P. D. 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*, number 1729 in *Lecture Notes in Mathematics*, pages 1–145. Springer, 2000.
- [29] P. D. Moral and L. Miclo. Annealed feynman-kac models. *Communications in Mathematical Physics*, 235:191–214, 2003.
- [30] R. Neal. Annealed inportance sampling. *Statistics and Computing*, 11:125–139, 2001.
- [31] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2002.
- [32] L. Rogers and D. Williams. *Diffusions, Markov Processes and Martingales*, volume 1. Cambridge University Press, Cambridge, 2 edition, 2001.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from [ftp.mpi-sb.mpg.de](ftp://ftp.mpi-sb.mpg.de) under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL <http://www.mpi-sb.mpg.de>. If you have any questions concerning ftp or WWW access, please contact reports@mpi-sb.mpg.de. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
 Library
 attn. Anja Becker
 Stuhlsatzenhausweg 85
 66123 Saarbrücken
 GERMANY
 e-mail: library@mpi-sb.mpg.de

MPI-I-2006-RG1-001	S. Hirth, C. Karl, C. Weidenbach	Automatic Infrastructure for Analysis
MPI-I-2006-5-005	R. Angelova, S. Siersdorfer	A Neighborhood-Based Approach for Clustering of Linked Document Collections
MPI-I-2006-5-004	F. Suchanek, G. Ifrim, G. Weikum	Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents
MPI-I-2006-5-003	V. Scholz, M. Magnor	Garment Texture Editing in Monocular Video Sequences based on Color-Coded Printing Patterns
MPI-I-2006-5-002	H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum	IO-Top-k: Index-access Optimized Top-k Query Processing
MPI-I-2006-5-001	M. Bender, S. Michel, G. Weikum, P. Triantafilou	Overlap-Aware Global df Estimation in Distributed Information Retrieval Systems
MPI-I-2006-4-009	J. Gall, J. Potthoff, B. Rosenhahn, C. Schnoerr, H. Seidel	Interacting and Annealing Particle Systems - Mathematics and Recipes
MPI-I-2006-4-008	I. Albrecht, M. Kipp, M. Neff, H. Seidel	Gesture Modeling and Animation by Imitation
MPI-I-2006-4-007	O. Schall, A. Belyaev, H. Seidel	Feature-preserving Non-local Denoising of Static and Time-varying Range Data
MPI-I-2006-4-006	C. Theobald, N. Ahmed, H. Lensch, M. Magnor, H. Seidel	Enhanced Dynamic Reflectometry for Relightable Free-Viewpoint Video
MPI-I-2006-4-005	S. Yoshizawa	?
MPI-I-2006-4-004	V. Havran, R. Herzog, H. Seidel	On Fast Construction of Spatial Hierarchies for Ray Tracing
MPI-I-2006-4-003	E. de Aguiar, R. Zayer, C. Theobald, M. Magnor, H. Seidel	A Framework for Natural Animation of Digitized Models
MPI-I-2006-4-002	G. Ziegler, A. Tevs, C. Theobald, H. Seidel	GPU Point List Generation through Histogram Pyramids
MPI-I-2006-4-001	R. Mantiuk	?
MPI-I-2006-2-001	T. Wies, V. Kuncak, K. Zee, A. Podelski, M. Rinard	On Verifying Complex Properties using Symbolic Shape Analysis
MPI-I-2006-1-007	I. Weber	?
MPI-I-2006-1-006	M. Kerber	Division-Free Computation of Subresultants Using Bezout Matrices
MPI-I-2006-1-004	E. de Aguiar	?
MPI-I-2006-1-001	M. Dimitrios	?
MPI-I-2005-5-002	S. Siersdorfer, G. Weikum	Automated Retraining Methods for Document Classification and their Parameter Tuning
MPI-I-2005-4-006	C. Fuchs, M. Goesele, T. Chen, H. Seidel	An Empirical Model for Heterogeneous Translucent Objects
MPI-I-2005-4-005	G. Krawczyk, M. Goesele, H. Seidel	Photometric Calibration of High Dynamic Range Cameras

MPI-I-2005-4-004	C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A., Magnor, H. Seidel	Joint Motion and Reflectance Capture for Creating Relightable 3D Videos
MPI-I-2005-4-003	T. Langer, A.G. Belyaev, H. Seidel	Analysis and Design of Discrete Normals and Curvatures
MPI-I-2005-4-002	O. Schall, A. Belyaev, H. Seidel	Sparse Meshing of Uncertain and Noisy Surface Scattered Data
MPI-I-2005-4-001	M. Fuchs, V. Blanz, H. Lensch, H. Seidel	Reflectance from Images: A Model-Based Approach for Human Faces
MPI-I-2005-2-004	Y. Kazakov	A Framework of Refutational Theorem Proving for Saturation-Based Decision Procedures
MPI-I-2005-2-003	H.d. Nivelle	Using Resolution as a Decision Procedure
MPI-I-2005-2-002	P. Maier, W. Charatonik, L. Georgieva	Bounded Model Checking of Pointer Programs
MPI-I-2005-2-001	J. Hoffmann, C. Gomes, B. Selman	Bottleneck Behavior in CNF Formulas
MPI-I-2005-1-008	C. Gotsman, K. Kaligosi, K. Mehlhorn, D. Michail, E. Pyrga	Cycle Bases of Graphs and Sampled Manifolds
MPI-I-2005-1-008	D. Michail	?
MPI-I-2005-1-007	I. Katriel, M. Kutz	A Faster Algorithm for Computing a Longest Common Increasing Subsequence
MPI-I-2005-1-003	S. Baswana, K. Telikepalli	Improved Algorithms for All-Pairs Approximate Shortest Paths in Weighted Graphs
MPI-I-2005-1-002	I. Katriel, M. Kutz, M. Skutella	Reachability Substitutes for Planar Digraphs
MPI-I-2005-1-001	D. Michail	Rank-Maximal through Maximum Weight Matchings
MPI-I-2004-NWG3-001	M. Magnor	Axisymmetric Reconstruction and 3D Visualization of Bipolar Planetary Nebulae
MPI-I-2004-NWG1-001	B. Blanchet	Automatic Proof of Strong Secrecy for Security Protocols
MPI-I-2004-5-001	S. Siersdorfer, S. Sizov, G. Weikum	Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning
MPI-I-2004-4-006	K. Dmitriev, V. Havran, H. Seidel	Faster Ray Tracing with SIMD Shaft Culling
MPI-I-2004-4-005	I.P. Ivriissimtzis, W.-. Jeong, S. Lee, Y.a. Lee, H.-. Seidel	Neural Meshes: Surface Reconstruction with a Learning Algorithm
MPI-I-2004-4-004	R. Zayer, C. Rssl, H. Seidel	r-Adaptive Parameterization of Surfaces
MPI-I-2004-4-003	Y. Ohtake, A. Belyaev, H. Seidel	3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs
MPI-I-2004-4-002	Y. Ohtake, A. Belyaev, H. Seidel	Quadric-Based Mesh Reconstruction from Scattered Data
MPI-I-2004-4-001	J. Haber, C. Schmitt, M. Koster, H. Seidel	Modeling Hair using a Wisp Hair Model
MPI-I-2004-2-007	S. Wagner	Summaries for While Programs with Recursion
MPI-I-2004-2-002	P. Maier	Intuitionistic LTL and a New Characterization of Safety and Liveness
MPI-I-2004-2-001	H. de Nivelle, Y. Kazakov	Resolution Decision Procedures for the Guarded Fragment with Transitive Guards
MPI-I-2004-1-006	L.S. Chandran, N. Sivadasan	On the Hadwiger's Conjecture for Graph Products
MPI-I-2004-1-005	S. Schmitt, L. Fousse	A comparison of polynomial evaluation schemes
MPI-I-2004-1-004	N. Sivadasan, P. Sanders, M. Skutella	Online Scheduling with Bounded Migration
MPI-I-2004-1-003	I. Katriel	On Algorithms for Online Topological Ordering and Sorting
MPI-I-2004-1-002	P. Sanders, S. Pettie	A Simpler Linear Time $2/3 - \epsilon$ Approximation for Maximum Weight Matching
MPI-I-2004-1-001	N. Beldiceanu, I. Katriel, S. Thiel	Filtering algorithms for the Same and UsedBy constraints