Particle Smoothing for Hidden Diffusion Processes: Adaptive Path Integral Smoother

Hans-Christian Ruiz and Hilbert J. Kappen

***Abstract*—Smoothing methods are used for inference of stochastic processes given noisy observations. The estimation of the marginal posterior distribution given all observations is typically a computationally intensive task. We propose a novel algorithm based on path integral control theory to efficiently estimate the smoothing distribution of continuous-time diffusion processes from partial observations. In particular, we use an adaptive importance sampling method to improve the effective sampling size of the posterior and the reliability of the estimation of the marginals. This is achieved by estimating a feedback controller, together with an adaptive initialization and an annealing scheme to sample efficiently from the joint smoothing distribution. We compare the results with estimations obtained from the standard Forward Filter/Backward Simulator (FFBSi) for two diffusion processes of different complexity. We show that the proposed method gives more accurate estimates than the standard FFBSi.**

***Index Terms*—Adaptive importance sampling, hidden processes, particle filtering, sequential Monte Carlo, smoothing methods, state-space models, stochastic optimal control, time series.**

I. Introduction

A. Problem Statement

In MANY fields of science and engineering access to physical time varying processes is limited to time series of noisy, indirect measurements. In order to extract information about the latent process, one estimates the so-called filtering or smoothing distributions. It is then possible to estimate the time evolution of the latent states, or estimate the parameters of a model, for example using an Expectation-Maximization procedure.

In this paper, we consider the smoothing problem for continuous time diffusion processes given a discrete number of observations. The latent process X_t is described by the following n-dimensional stochastic differential equation (SDE)

$$dX_t = F(X_t, t)dt + \sigma_{\rm dyn}(X_t, t)dW_t \qquad (1)$$

where dW_t is a m-dimensional Gaussian noise with $\mathbb{E}\left[dW_t\right] = 0$ and $\mathbb{E}\left[dW_r^i dW_s^j\right] = dt\delta_{i,j}\delta(r-s)$ and $\sigma_{\rm dyn}(x,t) \in \mathbb{R}^{n\times m}$ is a matrix that depends on the state x and time t. For given initial state x_0, (1) defines a distribution over processes $p_0(X_{(0,T]}|x_0)$. When the initial state x_0 is drawn from a distribution $p_0(X_0)$, this defines a prior distribution over processes $p_0(X_{[0,T]}) = p_0(X_0)p_0(X_{(0,T]}|X_0)$.

We assume an observation model $g(y|x)$ that denotes the probability of observation y at time t given the latent state x at time t. Given J observations y_{t_j} at times t_j, with $t_j \in [0,T]$ for all $j = 1,\ldots,J$ and $t_J = T$, this defines a likelihood $p(y_{0:T}|X_{[0,T]}) = \prod_{j=1}^{J} g(y_{t_j}|X_{t_j})$. The smoothing problem is to estimate marginals or statistics of the posterior distribution, also referred to as the smoothing distribution:

$$p(X_{[0,T]}|y_{0:T}) = \frac{1}{Z}p_0(X_{[0,T]})\exp\left(\sum_{j=1}^{J}\log[g(y_{t_j}|X_{t_j})]\right). \qquad (2)$$

with $Z = p(y_{0:T})$ the likelihood of the data.[1]

The smoothing problem is in general intractable when the dynamics (1) is non-linear or when the observation model is non-Gaussian. In those cases, it is needed to resort to approximate methods. One class of these methods is the deterministic approximation methods such as non-linear Kalman filtering [1], [2] and smoothing [3], or the variational method [4], which approximate the posterior by a simpler distribution. These methods are relatively efficient but may be inaccurate in some cases and will not be considered further in this paper.

In the remaining of this section, we will discuss three alternative classes of smoothing methods, first, particle filtering, second, adaptive importance sampling, and third, inference as a control problem. In the latter class, we will introduce our method “Adaptive Path Integral Smoother”.

B. Particle Filtering Methods

A prominent sampling based method, known as Sequential Monte Carlo (SMC) sampling or particle filtering is used to target the smoothing distribution. Particle filtering methods estimate the smoothing distribution by computing estimates of the filtered distribution and subsequently correct for these estimates. Each particle corresponds to an entire trajectory $X_{[0,T]}$. Among the various SMC methods for smoothing, one can distinguish broadly speaking three approaches; first, the bootstrap Filter-Smoother (FS) by [5], second, the forward-backward smoothers

Manuscript received May 25, 2016; revised November 27, 2016 and February 10, 2017; accepted February 28, 2017. Date of publication March 22, 2017; date of current version April 19, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pierre Borgnat. The work of H.-Ch. Ruiz is supported by the European Commission through the FP7 Marie Curie Initial Training Network 289146, NETT: Neural Engineering Transformative Technologies. *(Corresponding author: Hans-Christian Ruiz.)*

The authors are with the Biophysics Department, Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen 6525AJ, The Netherlands (e-mail: H.Ruiz@science.ru.nl; B.Kappen@science.ru.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2686340

[1] We denote time series of discrete observations by $y_{0:T} := (y_{t_1}, y_{t_2}, \ldots, y_{t_J})$ and continuous paths by $X_{[0,T]} := (X_s)_{s\in[0,T]\subset\mathbb{R}}$.

[6], [7]–with its many variations [8]–and third, the two-filter smoothers [9]–[11]. All these methods have their particular strengths and weaknesses. See e.g. [12]–[14] for a review on various filtering methods.

In naive particle smoothing each particle is sampled from forward simulation of $p_0(X_{[0,T]})$ and weighted with $w = \exp(\sum_{j=1}^{J} \log[g(y_{t_j}|X_{t_j})])$. With many observations (large J), the so-called degeneracy problem is introduced, where the weight w of one particle dominates all other weights. As a result, the representation of the smoothing distribution is very poor.

One can reduce the degeneracy by resampling the filtering particles. In its simplest form, the resampling step is done at each observation, but more sophisticated adaptive schemes exist [15]. Resampling is an effective way to improve the quality of the filtered estimates.

The trajectories of the resampled particles can also be used to estimate the smoothing distribution, as in the bootstrap Filter-Smoother (FS) [5]. However, the effect of resampling is that all trajectories arise from a very small number of common trajectories at early times. As a result of this "path degeneracy", the resampled trajectories give a poor representation of the smoothed marginals $p(X_t|y_{0:T})$ at early times $t \ll T$. The path degeneracy increases also exponentially fast as J increases [16]. In other words, resampling improves the filtered estimates but not the smoothed estimates.

The degeneracy problem is particularly severe when the observations deviate significantly from the prior process. In this case, the smoothing distribution may be very different from the filtering distribution causing weights with high variance and low effective sample size. As a result, the number of particles N needs to be prohibitively large to have moderate accuracy.

The quality of the smoothing estimates can be improved by adding a backward simulation, known as Forward Filter Backward Simulator (FFBSi) [6] which obtains trajectories approximately from the joint posterior. Applying the backward pass with M particles has a complexity $\mathcal{O}(MN)$. Since typically $M = \mathcal{O}(N)$ backward particles are required, the accuracy of this method is severely limited in practice by the computational cost. Several approaches have been developed to lower the computational effort while maintaining reliable estimates. For instance, in [17] a rejection sampling approach was suggested to avoid the computational complexity of evaluating all backward weights, effectively reducing the overall computational complexity to $\mathcal{O}(N)$ provided that N is sufficiently large. However in practice, this approach is less efficient than FFBSi for many problems and does not scale to high dimensions [8]. Recently [18], the rejection sampling approach has been used in a more efficient forward-only, online algorithm that computes expectations of additive functionals under the joint smoothing distribution.

The Forward Filter Backward Smoother [7] aims at approximating the marginal smoothing densities. This is done by reweighting the forward filter particles to target the posterior marginals. The computational complexity is $\mathcal{O}(N^2)$ due to the reweighting step.

An additional limitation of the backward methods, aside from the computational demands of some algorithms, is that they assume the existence of a non-degenerate backward kernel. In the case of the process (1), this means that the noise covariance matrix $\sigma_{\text{dyn}}\sigma'_{\text{dyn}}$ must be non-singular, which limits the applicability, for instance when the dynamics of some components of X_t is deterministic.

Finally, the forward-backward approaches have a further limitation in continuous time problems. The efficiency of the integration of SDEs can be increased significantly by replacing the standard Euler-Maruyama integration by a higher order scheme [19]. Since higher order schemes cannot be used in the backward computation step, the overall efficiency of backward methods can not be improved by these integration methods. See, however, [20] for some interesting work that allows higher order integration schemes using kernel density estimation.

Another approach within the SMC methods is the generalized two-filter smoother [10], which involves sampling from both a backward information filter and a forward particle filter. The particles of both filters are combined to obtain an approximation of the marginal $p(x_t|y_{0:T})$, which is used to sample approximately from the joint smoothing distribution. The method requires the choice of an artificial prior at each time point affecting the efficiency of the sampler. Besides, this method also requires $\mathcal{O}(N^2)$ samples.

As noted by [11], whenever the forward state transition probability $f(X_t|X_{t-dt})$ is approximately zero for most state pairs X^j_{t-dt}, X^i_t (sparse dynamics), the forward-backward smoother degenerates to being equivalent to the filter-smoother, albeit with substantially greater computational cost. The situation is worse for the two-filter smoother which fails completely as the forward and backward filter particles are sampled independently. This problem is particularly relevant for continuous time stochastic systems. Here, the variance of dX_t is proportional to dt thus, the transition probability from particle j at time $t - dt$ to particle i at time t is exponentially suppressed for all pairs $i \neq j$.

This issue is addressed in [11] by drawing new particles from the smoothing marginals directly. Although, the computational complexity of this approach is linear in the number of particles, it is not clear how to choose the required artificial densities in general. As a result, the method suffers from cumbersome design choices [21] which makes it impractical in many cases.

Other approaches that can ameliorate the particle degeneracy are developed in [22], [23]. Both methods propose to use Metropolis-Hastings moves to sample new positions and generate trajectories of the joint smoothing distribution given an existing particle system. In principle, this could move particles to higher density regions of the smoothing distribution and increase the effective sample size. In [22], the Metropolis-Hastings Improved Particle Smoother (MH-IPS) uses Gibbs sampling to sample a new state X_t given the remaining particle states. However, this method might be subject to strong dependencies between state variables, resulting in a poor mixing whenever the discretization time dt of the underlying SDE is sufficiently small.

Recently, [24] considered so called twisted models based on the idea of message passing through the Markov representation of the posterior (2). The messages are positive functions that need to be approximated iteratively. This is done by sampling from a “twisted” auxiliary particle filter and using the particles to estimate new messages. The disadvantage of this method is that in practice the transition density and messages are restricted to certain classes.

All the above methods have a particle filtering step in common.

C. Adaptive Importance Sampling

Importance sampling is a way of obtaining samples from a target distribution indirectly. The idea is to sample from a proposal distribution that is different from the target distribution and to weight the samples by importance sampling weights. Adaptive importance sampling [25], [26] adapts the parameters of the proposal distribution by minimizing some cost criterion, such as the Kullback-Leibler (KL) divergence or the chi-square distance between proposal and target distributions.

In [27] an adaptive importance sampling method is proposed for time-series models. This work uses an auxiliary particle filter [28] to construct adjustment multiplier weights that minimize the aforementioned risk criteria for a given proposal kernel. In addition, optimization techniques are proposed to adjust the proposal kernels by minimizing the risk criteria. For instance, the KL divergence is minimized using the cross-entropy method. To the best of our knowledge, this method has not been applied to the continuous time smoothing problem.

D. Inference as a Control Problem

A fundamentally different approach to address the smoothing and the degeneracy problem is to “steer” the particles through time based on future observations. Steering is optimal when the degeneracy problem is solved. In this sense, the smoothing problem can be viewed as a stochastic optimal control problem. The relationship between control and inference was first established by [29]–[31] who showed that the posterior inference for the smoothing problem (2) can be mapped onto a certain class of so called path integral control problems. In [32], it was shown how to compute the optimal control for these problems. Thus far, few authors have considered the application of this idea for smoothing. In this paper, we propose such an algorithm.

Nevertheless, we briefly review other approaches to inference that use ideas from control theory, but not from within the path integral control theory. In [33], it is shown that for a general non-linear diffusion with non-Gaussian observations, the optimal (state-dependent) Kalman gain can be computed at each time as an Euler-Lagrange boundary value problem. However, the approach is restricted to one-dimensional diffusion processes only. In [34], it is proposed to improve the posterior estimate by considering interacting particles. These so-called mean field game systems describe interacting particles whose density evolves according to a (forward) Fokker-Planck equation which is controlled by a (backward) Hamilton-Jacobi-Bellman equation. The disadvantage of this approach is that one needs to solve the HJB equation which is intractable for high dimensions.

In [30], the authors showed that the smoothing distribution (2) can be sampled with (3), which differs from (1) by a control term $u(x,t)$. The function $u(x,t)$ must be chosen optimally to minimize a control cost. The optimal control can be estimated for each x, t as a path integral. It can be shown that the optimal control gives the optimal (zero variance) importance sampler. In general, we cannot compute the optimal control function for all x, t. For the smoothing problem, we therefore propose a parametrized controller and learn the parameters by an iterative scheme, that was first proposed in [35]. We call this method Adaptive Path Integral Smoother (APIS). APIS iteratively reduces the variance of the weights for a given time-series and thus improves the sampling efficiency in terms of effective sample size. This improvement is limited mainly by the class of control functions that is considered. If the correct parametrization of the optimal control solution is available, the effective sample size is only limited by the numerical errors coming from the time discretization and the sample error. As a result, APIS requires increasing precision to maintain the sampling efficiency for longer time series, i.e. more particles and smaller integration steps are needed. In this paper, we restrict ourselves to linear state-feedback controllers and we show that these yield very reliable smoothing estimates even when used in non-linear systems.

An additional advantage of APIS for continuous time problems is that it does not contain a backward step, so it can be accelerated by using higher order integration schemes. Furthermore, there is not restriction on the degeneracy of the covariance matrix $\sigma_{\mathrm{dyn}}\sigma'_{\mathrm{dyn}}$. This is particularly useful for problems with mixed deterministic and stochastic dynamics. Finally, the variance of the estimates are not increased due to resampling because APIS does not require this step [12], [16].

In [36], preliminary results were shown on a small problem. In this paper, we provide the full detailed description of the implementation of the APIS method, and extend the method with a novel adaptive initialization of the particles and a novel annealing/bootstrapping scheme, which are both crucial for the sampling efficiency, in particular for large time series with many observations. In addition, we analyze in detail the quality of APIS in terms of effective sample size, we compare APIS with the vanilla flavor FFBSi and FS particle filtering algorithms and we analyze the scalability of APIS for up to 1000 observations.

E. Outline

This paper is organized as follows. In section II we review the main concepts in path integral (PI) control theory. We show how computing the joint smoothing distribution in continuous time is equivalent to a PI control problem. In section III, we discuss the importance sampling scheme for diffusion processes based on control. Then, we give an update rule to estimate a feedback controller and present the APIS algorithm. In section IV we present numerical examples. First, we consider the simple case of a one-dimensional linear diffusion process with Gaussian observations. We compare the accuracy and efficiently of FFBSi, the Bootstrap Filter-Smoother (FS) and APIS and show their performance as a function of the (un)likelihood of the observations. In addition, we demonstrate the benefit of the adaptive initialization proposed and examine the scaling of APIS up to 1000 observations. Then, we consider a 5-dimensional non-linear neural network model with multiple Gaussian observations. We demonstrate that the annealing procedure is significantly more efficient than the naive bootstrap and show that even a suboptimal linear feedback controller improves drastically the ESS. Moreover, we show that the estimation of the smoothing distribution is more reliable with APIS than with FFBSi and FS. In section V we comment on further considerations for the proposed algorithm. Finally, we outline possible extensions of this method that will be addressed in future work.

II. PI Control Theory and the Smoothing Distribution

We introduce the basic concepts regarding a subclass of stochastic control problems called Path Integral control problems, for more details see [32], [37], [38].

Stochastic optimal control theory considers systems under uncertain time evolution. The aim is to compute the optimal feedback control function to steer the system to a specified future goal. More formally, we have a continuous time stochastic process X_t ($t \in [0,T]$) described by the following n-dimensional SDE with the initial condition $X_0 = x_0$

$$dX_t = F(X_t,t)dt + \sigma_{\rm dyn}(X_t,t)[u(X_t,t)dt + dW_t] \tag{3}$$

where dW_t and $\sigma_{\rm dyn}(x,t)$ are as before in (1). We denote[2] the stochastic variable as X and the state as x. In addition to the drift $F(x,t)$, the process is driven by a feedback control signal $u(x,t) \in \mathbb{R}^m$.

We call realizations of the above process "particles". Each particle is a trajectory that accumulates a state cost $V(x,t)$ and a quadratic control cost. This accumulated cost is called the "path cost". The aim is to find the control function $u(x,t)$ that minimizes the expectation of the future path cost with respect to the process (3). The resulting optimal cost $J(x,t)$ at any time is called optimal cost-to-go,

$$J(x_t,t) = \min_u \mathbb{E}_u \left[\int_t^T V(X_s,s) + \frac{1}{2}||u(X_s,s)||^2 ds\right] \tag{4}$$

where the subscript u denotes the feedback control function[3] $u(x,s)$ for all $s \in [t,T]$ and $\|v\|^2 := \sum_{i=1}^m v_i^2$ denotes the usual Euclidean norm squared for a vector $v \in \mathbb{R}^m$. The expectation is defined as

$$\mathbb{E}_u\left[R(X_{(t,T]})\right] := \int dX_{(t,T]} p_u\left(X_{(t,T]}\right) R(X_{(t,T]})$$

for any function $R(X_{(t,T]})$ of continuous trajectories starting at a fixed x_t, $X_{(t,T]} := (X_s)_{s\in(t,T]\subset\mathbb{R}}|x_t$, and $p_u\left(X_{(t,T]}\right) := p(X_{(t,T]}|x_t,u)$. Notice that this density is conditioned on the control function $u(x,s)$ for all $s \in [t,T]$.

The optimal control

$$u^*(x_t,t) = \text{argmin}_u \mathbb{E}_u\left[\int_t^T V(X_s,s) + \frac{1}{2}||u(X_s,s)||^2 ds\right]$$

is the solution to this minimization.

We can express the expectation over the trajectories in (4) as a Kullback-Leibler divergence between a distribution over trajectories under the controlled dynamics (3) and the uncontrolled dynamics (1). To see this consider the following. In the limit of $ds \to 0$, the transition density between time s and $s + ds$ for the controlled process is given by a Gaussian

$$\hat{f}(x_{s+ds}|x_s,u) = \mathcal{N}\left(x_{s+ds}|x_s + \tilde{F}ds, \sigma_{\rm dyn}\sigma_{\rm dyn}'\right),$$

where $\tilde{F}{=}F(x_s,s) + \sigma_{\rm dyn}(x_s,s)u(x_s,s)$, $\sigma_{\rm dyn}{=}\sigma_{\rm dyn}(x_s,s)$ and $'$ denotes transpose. This density is proportional to (see e.g. [38, Appendix B])

$$\hat{f}(x_{s+ds}|x_s,u=0)\exp\left(\frac{1}{2}\|u(x_s,s)\|^2 ds + u(x_s,s)dW_s\right). \tag{5}$$

Multiplying (5) for all times s on the interval $(0,T]$, the distribution over controlled dynamics is proportional to the distribution over the uncontrolled dynamics (both conditioned on the initial state x_0) as

$$\begin{aligned} p_u\left(X_{(0,T]}|x_0\right) &= p_0\left(X_{(0,T]}|x_0\right) \times \ldots \\ &\exp\left(\frac{1}{2}\int_0^T \|u(X_s,s)\|^2 ds + \int_0^T u(X_s,s)dW_s\right). \end{aligned} \tag{6}$$

From this, we derive that[4]

$$\mathbb{E}_u\left[\log\frac{p_u\left(X_{(0,T]}|x_0\right)}{p_0\left(X_{(0,T]}|x_0\right)}\right] = \mathbb{E}_u\left[\int_0^T ds\frac{1}{2}||u(X_s,s)||^2\right]$$

is the KL divergence between the distribution over trajectories under the control $u(x,t)$ and the distribution over trajectories under the uncontrolled dynamics. Thus, the optimal cost-to-go at $t = 0$ is

$$J(x_0) = \min_u \mathbb{E}_u\left[V(X_{[0,T]}) + \log\frac{p_u\left(X_{(0,T]}|x_0\right)}{p_0\left(X_{(0,T]}|x_0\right)}\right] \tag{7}$$

where we define $V(X_{[0,T]}) := \int_0^T ds V(X_s,s)$.

Since the feedback control function $u(x,t)$ determines fully the distribution p_u, we can replace the minimization w.r.t. $u(x,t)$ with a minimization with respect to p_u subject to the normalization constraint $\int dX_{(0,T]}p_u\left(X_{(0,T]}|x_0\right) = 1$. The optimal control distribution conditioned on the initial state x_0 that minimizes (7) is then given by

$$p_{u^*}(X_{(0,T]}|x_0) = \frac{1}{\psi(x_0)}p_0(X_{(0,T]}|x_0)\exp\left(-V(X_{[0,T]})\right) \tag{8}$$

where the normalization constant is given by $\psi(x_0) := \mathbb{E}_{u=0}\left[\exp\left(-V(X_{[0,T]})\right)\right]$; see [38] for details. If we identify $V(X_{[0,T]}) = -\log\left[p(y_{0:T}|X_{[0,T]})\right]$ we see that the smoothing distribution for fixed initial state x_0 is identical to the optimal control distribution (8):

$$p(X_{(0,T]}|y_{0:T},x_0) = p_{u^*}(X_{(0,T]}|x_0) \tag{9}$$

When the initial state X_0 is drawn from a prior distribution $p_0(X_0)$ the smoothing distribution (2) is related to the optimal

[2] Note that we also distinguish between a deterministic function of state and time, e.g. $\sigma_{\rm dyn}(x,t)$, and its corresponding stochastic process $\sigma_{\rm dyn}(X_t,t)$.

[3] To simplify the notation in a formula, we omit the arguments of functions where the dependency is obvious from the context. Moreover, we some times write $J(x_t,t)$ to emphasize the dependency of a function $J(x,t)$ on the momentary value x_t of the trajectory $X_{[t,T]}$.

[4] Note that $\mathbb{E}_u\left[\int_t^T u(X_s,s)dW_s\right] = 0$ as a stochastic integral.

control distribution via

$$p(X_{[0,T]}|y_{0:T}) = p_{u^*}(X_{(0,T]}|X_0)p(X_0|y_{0:T}) \tag{10}$$

with

$$p(X_0|y_{0:T}) = \frac{\psi(X_0)p_0(X_0)}{p(y_{0:T})} \tag{11}$$

the posterior over the initial state. Thus, the posterior marginal at time $t = 0$ is proportional to the prior weighted with the expectation of the likelihood over all paths starting at X_0.

III. Importance Sampling as Controlled Diffusion

In this section, we show how sampling from the posterior can be done using controlled diffusions. We use a previous result that shows that when the control approaches the optimal control, the quality of the sampling, measured as the effective sample size, increases [35]. In general, we cannot compute the optimal control. We introduce the APIS method that adapts feedback controllers to optimize the sampling process.

A. Importance Sampling and the Relation to Optimal Control

Equation (2) suggest that we can sample from the smoothing distribution by sampling from the prior process and weighting each trajectory $X_{[0,T]}$ with $p(y_{0:T}|X_{[0,T]})$. We can use the control theory to improve the efficiency of the sampling. Combining (6) and (8)–(10) we can write

$$p(X_{[0,T]}|y_{0:T}) \propto p(X_0|y_{0:T})p_u(X_{(0,T]}|X_0) \times \dots \exp \\ \times \left(\sum_{j=1}^{J} \log[g(y_{t_j}|X_{t_j})] - \int_0^T \frac{1}{2}||u_s||^2 ds - \int_0^T u_s dW_s \right) \tag{12}$$

where we recall that $t_J = T$ and denote $u_s := u(X_s, s)$ for simplicity. We can thus sample from p_u and correct with the exponential term, i.e. an importance sampling procedure for diffusion processes [30], [31], [35], [36], [39], [40]. We call $u(x,t)$ the importance sampling control. In addition, we use importance sampling with a proposal distribution $q(X_0)$ to sample from (11).

We sample $i = 1, \dots, N$ particle trajectories. For each particle, we define an importance weight

$$\begin{aligned} \alpha_u &= \exp[-S_u] \qquad (13) \\ S_u &:= -\sum_{j=1}^{J} \log[g(y_{t_j}|X_{t_j})] + \int_0^T \frac{1}{2}||u_s||^2 ds \\ &\quad + \int_0^T u_s dW_s + S^0 \end{aligned}$$

and $S^0 := -\log[\frac{p_0(X_0)}{q(X_0)}]$ and normalize such that $\sum_{i=1}^N \alpha_u^i = 1$. Notice that the weights α_u depend on all observations $y_{0:T}$ and on $u(x,t)$ through S_u.

Since the cost assigned to the importance weight targeting $p(X_0|y_{0:T})$ differs from S^0 only by a constant shift, we can use S^0 as the importance correction, see (8), (10), (11).

The quality of the sampling can be quantified in terms of the effective sample size, which we define as [41]

$$ESS = \frac{N_{\text{eff}}}{N} = \frac{1}{Var(\alpha_u) + 1}. \tag{14}$$

with $Var(\alpha_u)$ the empirical variance in the N sample weights. We see that reducing the variance of the weights increases the efficiency of the sampling procedure. In [35] upper and lower bounds for $Var(\alpha_u)$ were found. The upper bound

$$Var(\alpha_u) \leq \int_0^T \mathbb{E}_u \left\{ ||\alpha_u [u^*(X_t,t) - u(X_t,t)]||^2 \right\} dt \tag{15}$$

shows that the optimal control function $u^*(x,t)$ is the optimal importance sampler in the sense that the importance weights have zero variance and the ESS becomes maximal. Hence, the better we approximate $u^*(x,t)$, the higher the efficiency of our importance sampler will be.

B. Adaptive Path Integral Smoother

Clearly, it is difficult to compute the optimal control in general. However, we can efficiently estimate a suboptimal control using the approach introduced in [35]. Assume that the optimal control can be approximately parametrized as

$$u^*(x,t) = A^*(t)h(x,t) \in \mathbb{R}^m \tag{16}$$

where $A^*(t) \in \mathbb{R}^{m \times k}$ are time-dependent parameters and $h : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^k$ are the k "basis" functions of the feedback controller. In addition, we choose the importance sampling control to be parametrized with the same basis functions: $u(x,t) = A(t)h(x,t)$. The main theorem in [35] implies

$$A^*(t) \langle h_t \otimes h_t \rangle = A(t) \langle h_t \otimes h_t \rangle + \lim_{\delta t \to 0} \frac{\left\langle \int_t^{t+\delta t} dW_s \otimes h_s \right\rangle}{\delta t} \tag{17}$$

where $h_t := h(X_t,t)$, $h_t \otimes h_t$ is the outer product $(h_t \otimes h_t)_{kk'} = h_k(X_t,t)h_{k'}(X_t,t)$ and $\langle \bullet \rangle = \mathbb{E}_u[\alpha_u \bullet]$ is the weighted average targeting the posterior.

In practice the limit in the right side of (17) may lead to numerical instability when estimated with a finite number of particles and time discretization $dt > 0$. Therefore, one may consider taking $\delta t \geq dt$ which yields a smoothed biased estimate of $u(x,t)$ with less variance. Around observations, the control may be a sensitive function of time and a small δt is required. In the reminder of the article we set $\delta t = dt$.

Equation (17) describes a procedure to compute an estimate of the optimal control $u^*(x,t)$ based on an importance sampling control $u(x,t)$. We can iterate this idea where in iteration r we estimate $A^r(t)$ as $A^*(t)$ in (17) with samples that we generate with a control function with parameters $A^{r-1}(t)$ from the previous iteration. Then, (17) becomes

$$A^{r+1}(t) = A^r(t) + \eta \frac{dQ(h_t)}{dt} H^{-1}(t) \tag{18}$$

where $H(t) = \langle h_t \otimes h_t \rangle_r \in \mathbb{R}^{k \times k}$ and $dQ(h_t) := \langle dW_t \otimes h_t \rangle_r \in \mathbb{R}^{m \times k}$. The learning rate $\eta < 1$ accounts for sample errors at the beginning of the learning procedure, when the ESS is low.

We need to estimate $H(t)$ and $dQ(h_t)$. Both can be obtained by sampling N particles via numerical integration of (3) and

Algorithm 1: Adaptive Path Integral Smoother.

```
Input: Observations $y_{0:T}$, prior $p_0(x_0)$, control
  parametrization $u^0(x,t) = A^0(t)h(x,t)$, learning rate
  $\eta < 1$, particles $N$, iterations $I_{\max}$, ESS threshold
  $\theta_{\text{ess}} \leq 1$, annealing factor $\beta > 1$ and annealing
  threshold $\gamma \geq 0$.
Output: Smoothing particle system $\{x^i_{[0,T]}, \alpha^i_u\}_{i=1:N}$
  and importance controller $u(x,t)$.
Set $n \leftarrow 0$
while $ESS < \theta_{\text{ess}}$ or $n \leq I_{\max}$ do
  if $n = 0$ then $x^i_0 \sim p_0(x_0)$ for $i = 1, \ldots, N$
  else
    $x^i_0 \sim \mathcal{N}(\hat{\mu}_0, \hat{\sigma}^2_0)$ for $i = 1, \ldots, N$
    $S^i_u = -\log(p_0(x^i_0)/\mathcal{N}(x_0|\hat{\mu}_0, \hat{\sigma}^2_0))$
  end if
  Generate: $\{x^i_{[0,T]}, \alpha^i_u\}_{i=1:N}$ according to (3) and (13).
  Estimate $ESS$ from (14)
  while $ESS < \gamma$ do
    $S^i_u \leftarrow S^i_u/\beta$ for $i = 1, \ldots, N$
    Estimate $\alpha_u$ from (13)
    Estimate $ESS$ from (14)
  end while
  Compute: $\hat{\mu}_0, \hat{\sigma}^2_0$ from $\{x^i_0, \alpha^i_u\}_{i=1:N}$ with (20).
  for $t = 0, \ldots, T$ do
    Estimate: $H(t)$ and $dQ(h_t)$ with (19)
    Update: $A_t \leftarrow A_t + \eta \frac{dQ(h_t)}{dt} H(t)^{-1}$
  end for
  $n \leftarrow n + 1$
end while
```

weighting each with its corresponding α_u. Then, the expectation at each time t is a weighted average over the particle system $\{X^i_t\}_{i=1,\ldots,N}$,

$$H(t) = \sum_{i=1}^{N} \alpha^i_u h(X^i_t, t) \otimes h(X^i_t, t) dQ(h_t)$$
$$= \sum_{i=1}^{N} \alpha^i_u dW^i_t \otimes h(X^i_t, t) \tag{19}$$

where $\sum_{i=1}^{N} \alpha^i_u = 1$ and dW^i_t is the noise realization of the i-th particle at time t.

The posterior initial state $p(X_0|y_{0:T})$ is sampled using a Gaussian adaptive importance sampling distribution $q(X_0) = \mathcal{N}(X_0|\hat{\mu}_0, \hat{\sigma}^2_0)$ with $\hat{\mu}_0$ and $\hat{\sigma}^2_0$ the mean and covariance of the marginal posterior at time $t = 0$. After the first iteration, we update

$$\hat{\mu}_{0,l} = \langle X_{0,l} \rangle \quad \left(\hat{\sigma}^2_0\right)_{jl} = \langle (X_{0,j} - \hat{\mu}_{0,j})(X_{0,l} - \hat{\mu}_{0,l}) \rangle \tag{20}$$

for each $l, j = 1, \ldots, m$.

Sampling from $q(X_0)$ ensures that we adapt the initialization of the particles to stay close to the posterior distribution. Since we sample from a Gaussian, the initialization procedure is simple and efficient for many cases. The update rules (20) follow by maximizing a variational lower bound on the likelihood $p(y_{0:T})$ with respect to a Gaussian distribution that describes the prior distribution at initial time. We call this initialization scheme the (Gaussian) *adaptive initialization*.

Equations (18)–(20) define the Adaptive Path Integral Smoother (APIS) that learns iteratively the feedback controller defined in (16). This is an adaptive importance sampling procedure to obtain samples from the joint smoothing distribution using controlled diffusion. Note that the control parameters $A(t)$ are estimated for each time t independently.

The APIS algorithm starts by sampling from the uncontrolled dynamics. We initialize the particles from $q(X_0) = p_0(X_0)$ if possible, otherwise they are initialized from a proposal distribution $q(X_0)$ of our choice, for instance a Gaussian. In this case, the cost S^0 for the initialization is non-zero also for the fist iteration. For subsequent iterations, the update rules (18)–(20) are repeated until the ESS reaches a threshold $\theta_{\text{ess}} \leq N_{\text{eff}}/N$ or a maximal number of iterations $I = I_{\max}$. Alternatively, one can check if the variance of the weights or the ESS has changed significantly in the last l iterations and stop if the change is small. The resulting weighted particle system gives an estimate of the smoothing distribution.

The number of particles N is one of the most important parameters. The variance of the estimates reduces with N. In practice, we need a large number of particles to ensure sufficiently good estimates. The complexity of APIS is $\mathcal{O}(IN)$, where I is the number of adaptation iterations and N the number of particles.

The learning rate η determines the rate at which the control function u increases from its initial value zero. We observe poor improvement of the importance sampler in terms of the ESS for large learning rate η. In our experiments, we find good results with $\eta \in [0.001, 0.05]$ depending on the variance of the estimations.

Special attention is required at the initial iterations. Since the initial importance sampler is very poor, the ESS is extremely low and the estimates (18) and (19) are very inaccurate. For this reason we artificially increase the ESS to a predetermined minimum number of particles N_0 by introducing an 'adaptive annealing procedure' with a temperature $\lambda > 1$ that scales the cost of each particle i as $S^i_u \to S^i_u/\lambda$. For a given set of particles we can then estimate the ESS for different values of lambda λ. The smallest value of λ such that $ESS \approx N_0/N$ is found by setting $\lambda = \beta^m$ with $m = 0, 1, 2, \ldots$ and $\beta > 1$. The annealing factor β should be chosen not too large to prevent overshoot, and not too small to restrict the number of m steps. We find that values of $\beta \in [1.05, 1.15]$ prove to work well and finding λ is very fast. The adaptive annealing procedure is done whenever the ESS is below the threshold $\gamma = N_0/N$. In our experiments we use $N_0 = 100 - 150$.

IV. Results

In this section, we present numerical results to show the efficiency and accuracy of APIS compared to FS and FFBSi. We illustrate the benefits of the adaptive initialization and the annealing scheme. Additionally, we demonstrate the scaling of APIS to very high number of observations when the importance control has the correct parametrization.

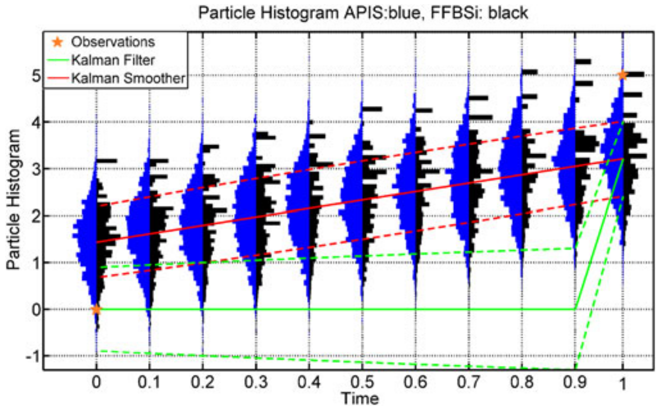

Fig. 1. Kalman Smoother solution. Notice the small overlap of the filtering (green) and smoothing (red) solutions due to the unlikely observation (orange) at $y_T = 5$. Violin plots (histograms) of particles obtained by FFBSi (black) and APIS (blue): snapshots every $\Delta t = 0.1$ starting at $t = 0$. Notice the poor particle representation in FFBSi. For APIS we used the following parameters: $N = 2000$ particles, learning rate $\eta = 0.2$, no annealing procedure and $I_{\max} = 15$ iterations. For FFBSi we used $N = M = 2000$ forward and backward particles. Color figures online.

For all numerical experiments, we fix the choice of the basis functions to a linear feedback term and an open-loop controller (no state dependence, only time dependence). For details on this choice and the implementation we refer the reader to the appendix.

A. Linear Quadratic System

1) Low Likelihood Observation: Consider a Brownian motion $X_{s+ds} \sim \mathcal{N}(x_s, \sigma^2_{\text{dyn}} dt)$ with $\sigma_{\text{dyn}} = 1, dt = 0.01$ and a Gaussian observation model $y_t \sim \mathcal{N}(x_t, \sigma^2_{\text{obs}} = 1)$ for $t = 0, T$. We fix the observations at $y_0 = 0$, $y_T = 5$ and the length of the series at $T = 1$. The initial distribution $p_0(X_0)$ is a Gaussian centered at $x_0 = 0$ with variance $\sigma_0^2 = 4$. The exact solution for this model is given by the Kalman smoother, Fig. 1. Notice the poor overlap of the filtering and smoothing distributions.

We compare the particle smoothing distribution given by APIS and FFBSi. In Fig. 1, we show violin plots[5] for a particular realization of the particles at times $t = 0, 0.1, 0.2, \dots, 1$. Although N is large, FFBSi poorly represents the Gaussian posterior marginal distributions. The effect worsens for large $t \lesssim T$, where filtered and smoothed marginals differ most. The histograms for the bootstrap filter-smoother (FS) are similar to those of FFBSi (not shown). On the contrary, APIS histograms represent much better the Gaussian distribution.

If the filtered particles do not represent the smoothing distribution well enough, the backward pass will have a low ESS and therefore the backward particles will mix poorly. Although we observe an increase in the averaged ESS of the backward pass from 2% at $t \lesssim T$ to 7% for times $0 \lesssim t$, this is not enough to improve the estimations, see Fig. 2. For comparison, APIS increases the ESS of the whole path from 1.5% to 98% in 15 iterations by adapting the trajectories from the initial filtering distribution to the smoothing distribution.

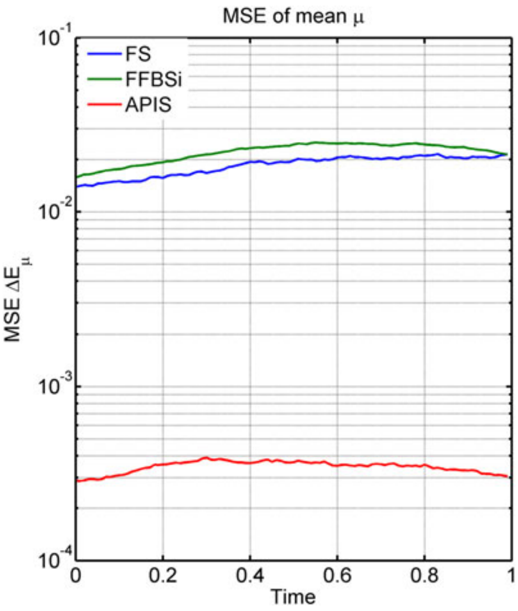

Fig. 2. MSE of mean $\hat{\mu}$ over time. Estimates averaged over $R = 250$ runs to avoid effects of the particular sampling realizations. We used $N = 2000$ forward particles in all methods and $M = 2000$ backward particles in FFBSi; observations at $y_0 = 0$ and $y_T = 5$. Notice the error for APIS (red) at all times; it is two orders of magnitude lower than FS and FFBSi. In APIS, estimations are made using the last particle system obtained after $I_{\max} = 15$ iterations and without annealing ($\gamma = 0$).

We can use the exact solution to compare the performance of all methods using the mean squared error (MSE)

$$\widehat{\Delta E}_\mu(t) = \frac{1}{R}\sum_{j=1}^{R}(\hat{\mu}_j(t) - \mu_{\text{KS}}(t))^2$$

where μ_{KS} is the mean of the ground truth obtained by the Kalman smoother, $\hat{\mu}_j$ is the estimated mean of each method in run j and R the number of runs. In each run, we used the same parameters as above. Fig. 2 shows $\widehat{\Delta E}_\mu$ versus t. We observe that APIS has an accuracy two orders of magnitude higher than FS and FFBSi. The errors of the variance estimates are very similar to the errors of the mean estimates.

Note the slight increase of the MSE of FFBSi vis-á-vis FS. This may be due to the small number of observations. In this example, there is no gain in applying a backward pass. For larger time series we observe an improvement of the estimates in FFBSi compared to FS, as is to be expected. However, APIS was consistently better in all the examples studied.

Fig. 3 shows the error of the mean $\widehat{E}_\mu = \frac{1}{T}\int_0^T \widehat{\Delta E}_\mu(s)ds$ as a function of the unlikely observation y_T. Notice how the performance of both FS and FFBSi are comparable to APIS if the observation is close to the high density region of the filtering ($y_T \in [0, 2]$) but deteriorates very fast for unlikely observations. On the contrary, the error in APIS is virtually independent of the position y_T of the observation. This is due the adaptation of APIS to the likelihood.

To show the importance of the adaptive initialization introduced above, we examine the ESS dependent on both, initializing the particles with the prior distribution–the non-adaptive

[5] We used distributionPlot.m for MATLAB

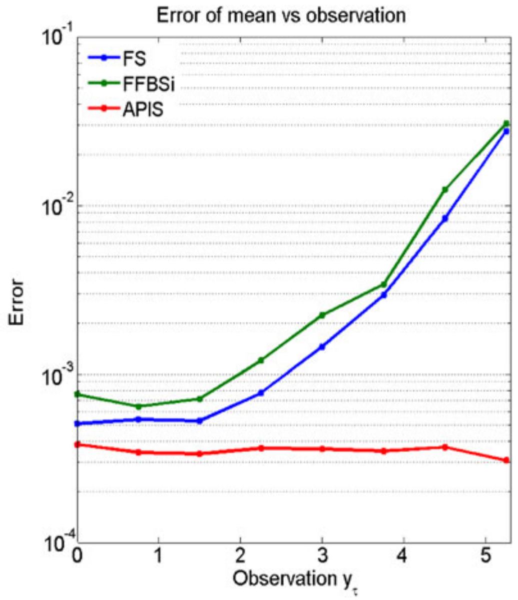

Fig. 3. Error $\hat{E}_\mu$ for $y_T \in \{0, 0.75, 1.5, 2.25, 3, 3.75, 4.5, 5.25\}$ and always $y_0 = 0$: For each y_T, we estimate $\widehat{\Delta E}_\mu(t)$ using 100 runs and $N = M = 2000$ particles. Notice the logarithmic scale in the error-axis. The error in the variance is similar. APIS has no annealing in this example.

TABLE I
IMPACT OF THE INITIALIZATION OF THE PARTICLES ON THE ESS FOR $J = 2$ OBSERVATIONS OF A BROWNIAN MOTION

Noise σ^2_{dyn}	0.05	1.4	6	8
ESS (adaptive)	0.996	0.985	0.94	0.93
ESS (prior)	0.08	0.49	0.67	0.66

initialization–and with the adaptive initialization. We show that the adaptive initialization is crucial for the sampling efficiency.

Similar to the experiment above, we consider $J = 2$ observations of a Brownian motion with $\sigma^2_{\text{obs}} = 0.5$. The prior is defined as a standard Gaussian. We use $N = 2000$ particles for a maximum number $I_{\max} = 500$ of iterations with $\eta = 0.01$ and y_0, y_T, γ, T, dt as before.

Table I shows the effect on the ESS for various σ^2_{dyn} using both initialization schemes. We see that the adaptive initialization yields significantly higher ESS for all noise levels. The effect is largest for low noise levels. The reason is that in this regime the maginal posterior at $t = 0$ is significantly different from the prior due to the large influence of the future observation y_T. Hence, sampling from the prior at $t = 0$ significantly reduces the ESS of the entire trajectories even if the controller is successfully learned. The small decrease of the ESS for the adaptive initialization is due to the increased precision needed to find the large magnitude control solutions in the high noise regime.

2) High Number of Observations: Now, we study in more detail the ESS of APIS for higher number of observations J. Consider again a Brownian motion with $\sigma^2_{\text{dyn}} = 0.75, \sigma^2_{\text{obs}} = 0.9$ and a time horizon $T = 3$. The prior $p_0(X_0)$ is as before. We generate a single time series of 300 observations on the time interval $[0, T]$ and define a posterior estimation problem given the first 100, 200 and all observations. For each problem we estimate the ESS after 100 iterations.

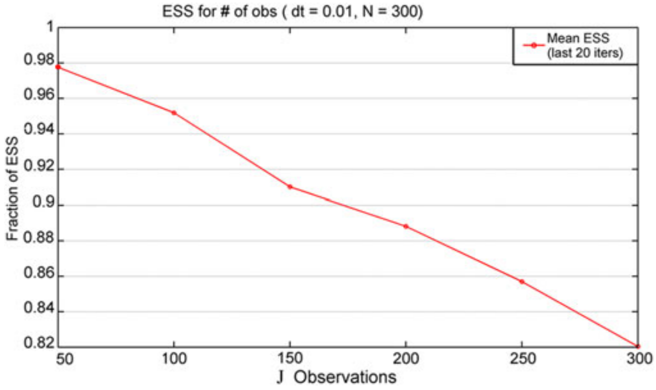

Fig. 4. ESS estimated for 100, 200 and 300 observations. We use a learning rate of $\eta = 0.05$, $N = 300$ particles, $I_{\max} = 100$ and no annealing ($\gamma = 0$). The ESS decays slowly with the number of observations.

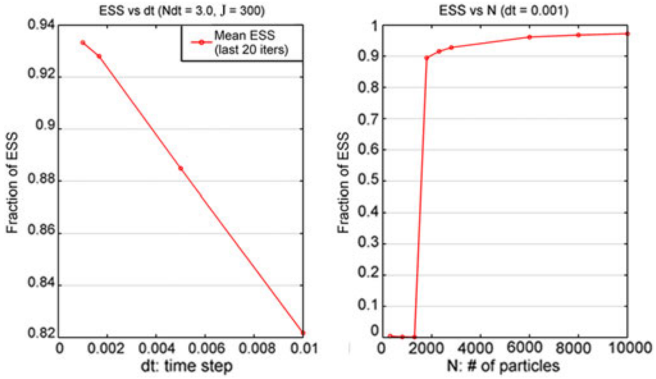

Fig. 5. Mean ESS over the last 20 iterations for a time series of 300 observations. Left: ESS for increasing precision. While dt decreases, N increases such that $Ndt = 3$. Right: ESS for increasing number of particles. The integration step is fixed at $dt = 10^{-3}$. The APIS parameters $\eta, I_{\max}$ and γ are the same as in Fig. 4.

In Fig. 4 we observe a slow decay of the ESS when the number of observations J increases. This decay can be compensated with an increase in the precision of the estimations. We illustrate this by increasing the number of samples N while decreasing the integration step dt such that Ndt remains constant. The consequence is an increase of the ESS for $dt \to 0$ as in Fig. 5 left. On the right of the same figure, we show the ESS for the same time series of 300 observations but with fixed dt and incrementing N. We observe a fast increase of the ESS and saturation[6] for higher N.

The excellent scaling with the number of observations is due to the correct parametrization of the importance control function. However, for small samples sizes N, the variance is too large to efficiently bootstrap APIS and an increase in ESS is not guaranteed. The minimum amount of particles needed to bootstrap APIS is problem dependent. However for a given problem,

[6] Accordingly, the MSE of both estimators decreased very fast until it saturated at a much lower value due to estimation errors in the controller (not shown).

the discretization step dt has a big impact on the choice of N. As a rule-of-thumb, we find that for a fixed dt, one must choose at least $N > 2/dt$ to have stable results.

More complex problems require higher number of samples N. In this case, the annealing procedure helps to avoid prohibitive large N. We considered now $J = 1000$ for the same system and parameters as above except that we anneal the weights if the ESS is below a threshold $\gamma = 0.01$ ($\beta = 1.15$). This allows us to use only $N = 10^4$ particles. Above γ there is no annealing anymore and the raw ESS converges to a value around 0.6 in less than 200 iterations. This is a remarkable increase of 3 orders of magnitude vis-á-vis the uncontrolled dynamics. After learning, the absolute error of the mean $|\hat{\mu}_{\mathrm{APIS}} - \mu_{\mathrm{KS}}|$ stays lower than 0.01 over time and the averaged absolute error is 1.8×10^{-3}. The absolute error in the variance is similar.

We can bootstrap APIS because we obtain a higher ESS from the annealed particle system. This allows us to estimate a control that improves the raw ESS incrementally. Without annealing, the ESS stays at 2×10^{-4} even after 1000 iterations. This result shows the importance of the annealing procedure when the number of observations is very large or the problem too complex. We refer to the next section for a more detailed illustration of the benefits of this procedure.

The above analysis shows that the ESS in APIS scales very well with the number of observations given the correct parametrization of the controller. Moreover, the error of the estimates stays small over the whole time interval.

B. A Neural Network Model

We consider a non-linear system and show the efficiency of bootstrapping with annealing in more detail. In addition, we examine the performance of APIS, FFBSi and FS. In this example the linear feedback control is clearly suboptimal. However, we show that the variance of the estimates is lower for APIS than for FFBSi and FS.

We consider a 5 dimensional non-linear neural network described by

$$dX_t = -X_t dt + \tanh(BX_t + \theta + A\sin(\omega t))dt + \sigma_{\mathrm{dyn}} dW_t$$

where $B \in \mathbb{R}^{5\times5}$ and $\theta \in \mathbb{R}^5$ are an antisymmetric connectivity matrix and a threshold vector respectively. The elements of the vector $A \in \mathbb{R}^5$ are the amplitudes of independent sinusoidal inputs with frequencies given by $\omega \in \mathbb{R}^5$. We choose the values randomly from Gaussian distributions $\theta_i \backsim \mathcal{N}(0, \sigma_\theta = 0.75)$, $A_i \backsim \mathcal{N}(0, \sigma_A = 2)$, $\omega_i \backsim \mathcal{N}(\pi/5, \sigma_\omega = \pi)$ and $B_{\mathrm{ij}} \backsim \mathcal{N}(0, \sigma_B = 2)$ with $B_{\mathrm{ij}} = -B_{ji}$, for all $i, j = 1, \ldots, 5$. In addition, we set $\sigma^2_{\mathrm{dyn}} = 0.05$ and an integration step of $dt = 0.01$.

Furthermore, we assume a Gaussian observation model with $Y_{t_i} \backsim \mathcal{N}(X_1(t_i), \sigma_{\mathrm{obs}} = 0.1)$ for $i = 1, \ldots, J$ and sample an observation every $\Delta_{\mathrm{obs}} = 10dt$. Note that only one of the five neural states is observed.

First, we examine the annealing procedure and show its benefit compared to the ”raw” bootstrap procedure with no annealing. We consider several smoothing problems for the above system with different number of observations $J = 20 - 42$. We fix the initial condition at $x_0 = 0$ to examine the ESS without the effect of importance sampling at the initial state. We generate a single time series using a random realization of the neural network and take the first J observations for each one of the 7 cases considered. We examine the performance in terms of the iteration number for which the ESS hits the 10% mark for the first time. We call this the *exit index*.

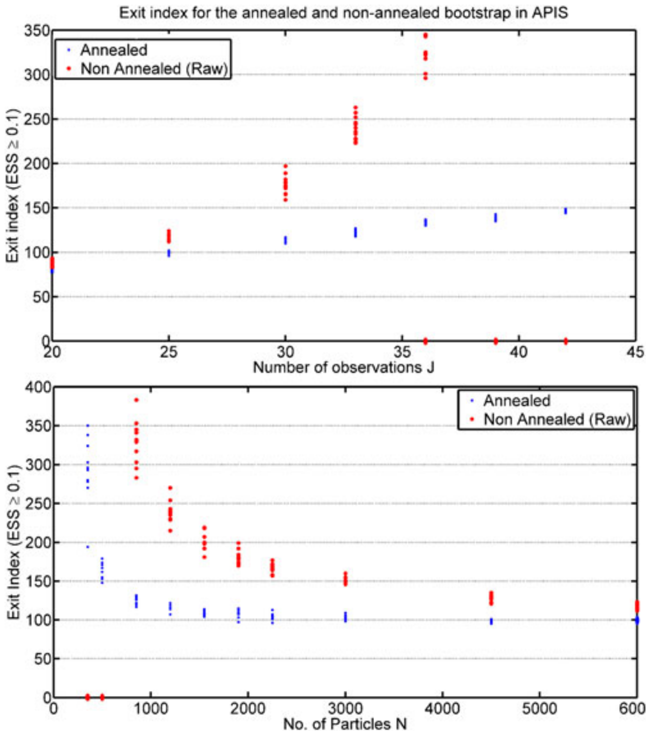

Fig. 6. Exit index for both bootstrapping schemes (blue: annealing with $\gamma = 0.02$; red: no annealing) shows the advantage of annealing. Top: ESS (15 independent runs) versus number of observations for fixed computational budget $N = 6000, I_{\max} = 350$. Bottom: ESS (10 independent runs) versus number of particles for fixed number of observations $J = 25$ and $I_{\max} = 600$. Note: For 39 and 42 observations (top figure), and for 350 and 500 particles (bottom figure), all runs with $\gamma = 0$ failed to bootstrap so there is no exit index defined. We mark these cases on the figures with an exit index of zero.

Fig. 6-Top shows the exit index for a fixed computational budget $N = 6000, I_{\max} = 350$. The sampling efficiency for short time series is similar in both bootstrapping schemes. Without annealing ($\gamma = 0$) however, the efficiency drops fast and after a certain number of observations, the raw scheme cannot bootstrap. In contrast, the annealed scheme allows APIS to bootstrap within a similar number of iterations, only slightly increasing for larger time series. Thus, the annealed estimates increase the ESS more reliably and in less iterations compared to the raw bootstrapping scheme.

Annealing not only helps bootstrapping for a fixed computational budget, it also reduces the number of particles needed. To show this, we examine how the exit index depends on the number of particles with $J = 25$ observations. Fig. 6-Bottom shows that with annealing, we are able to bootstrap with significantly less particles. In addition, annealing is more reliable and arrives much faster at the minimum exit index. For a similar performance, we would need an order of magnitude more particles for the raw bootstrap.

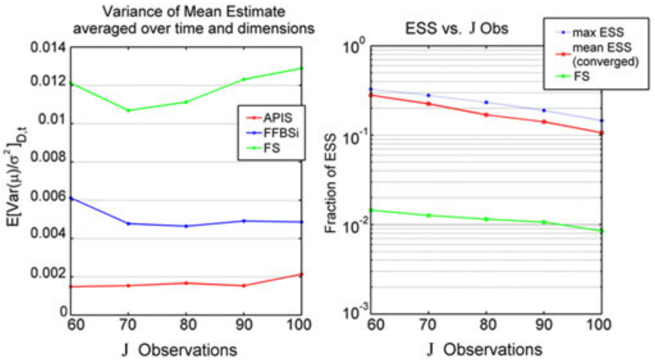

Fig. 7. Left: variance of the mean $\hat{\mu}$ across 10 estimations, scaled by the variance of the posterior. This is averaged over time and dimensions to give a single measure for each J. Right: ESS for different number of observations J. Notice the logarithmic scale. The ESS of FS is taken as the number of unique trajectories. We use the same amount of forward particles $N = 6000$ in all 3 methods, and the number of backward particles is set such that the CPU time spend on FFBSi and APIS is similar. The estimation of the posterior in APIS is accepted when a predefined threshold of $\theta_{\text{ess}} = 0.1$ is reached. Each algorithm was repeated $R = 10$ times to estimate the variance. We used a fixed initial condition $x_0 = 0$ and $\gamma = 0.02$ in APIS.

We see that annealing helps to scale up APIS to longer time series and reduces the number of particles needed. This efficiency is crucial whenever sampling from the hidden process is computationally expensive.

We compare now the performance of APIS and FS. The system has a fixed initial condition as in the example above. In Fig. 7 right, we compare the ESS of APIS and FS as a function of the number of observations.[7] The ESS is an order of magnitude higher for APIS than for FS, but the efficiency of both decrease with the number of observations. The ESS of APIS starts at around 30% for 60 observations and ends at around 10% for 100 observations. Moreover, the ESS cannot be increased much further with higher precision. This is the result of a suboptimal importance control.

Nevertheless, the efficiency and performance of APIS clearly increases compared to FS. As seen in Fig. 7 left, the variance of $\hat{\mu}$ in APIS is significantly lower than in FS and FFBSi. We have similar results for the variance of $\hat{\sigma}^2$.

Now, consider a Gaussian prior $p_0(X_0)$ with mean $\mu_0 = 0$ and variance $\sigma_0^2 = 1$. We study the performance of the three methods. Fig. 8 left shows the variance of the mean for the observed neuron. All three methods have reliable estimates but both FS and FFBSi have higher variance. APIS keeps the variance of the estimates consistently lower over the entire time interval.

Fig. 8 right shows that the variance of the mean for the hidden neurons is up to two orders of magnitude higher for FS and FFBSi than for APIS. The increased variance towards earlier times is in part an effect of the importance sampling procedure at the initialization, which affects all methods. Nevertheless, the adaptation of the proposal distribution $q(X_0)$ in APIS reduces significantly this effect.

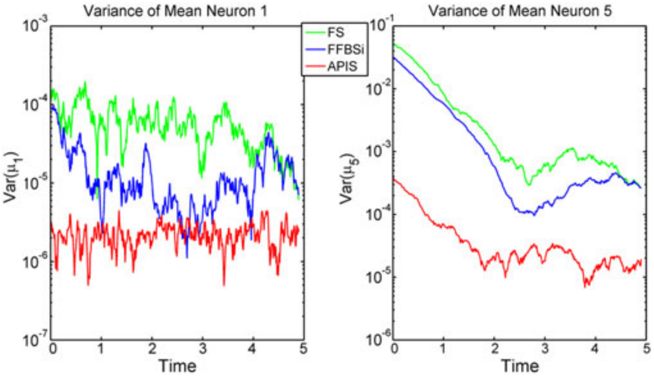

Fig. 8. Variance of mean estimate $\hat{\mu}$ for the partially observed neuron 1 (left) and for the hidden neuron 5 (right). The variance is obtained from $R = 12$ estimates. Green: FS; blue: FFBSi; red: APIS. Notice that APIS has a lower variance up to two orders of magnitude (log-scale). The estimations for all other neurons 2,3,4 are similar to neuron 5. The setting is similar as in Fig. 7 but with $J = 50$. In APIS, we use $\gamma = 0.02$ and $N = 7500$ forward particles. The ESS threshold is set to $\theta_{\text{ess}} = 0.2$. In FS and FFBSi we use $N = 5000$ forward particles and $M = 2500$ backward particles such that APIS and FFBSi have again the same CPU time available.

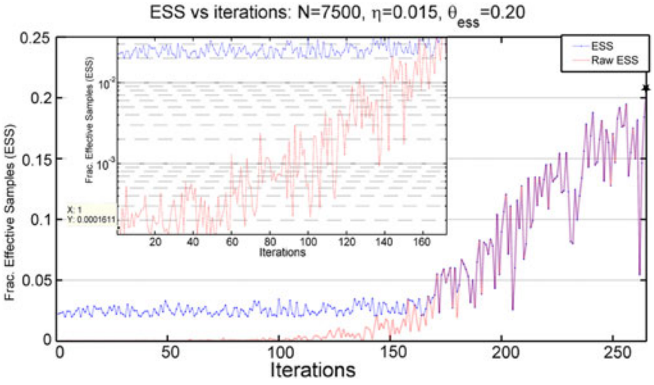

Fig. 9. Effective Sample Size (ESS) for a single run. Black marker symbolizes accepted samples used for the estimation of the smoothing distribution ($\theta_{\text{ess}} = 0.2$).

Finally in Fig. 9, we show the typical improvement of the ESS for this example. At the beginning, the ESS is around 2% due to annealing (blue line, $\lambda > 1$) and the raw ESS increases from 0.02% up to 2% (red dotted line, $\lambda = 1$). This increase is due to the control estimations obtained from the annealed particle system. After the ESS surpasses γ, APIS reaches the stopping threshold θ_{ess} in about 80 iterations. The final ESS of 20% is a remarkable improvement vis-á-vis the ESS of the posterior marginals in FS, which stays most of the time below 10% and around 2% for times close to $t = 0$.

V. Discussion

In this work, we present a new smoothing algorithm for diffusion processes in continuous time. This method estimates iteratively a feedback controller to target the posterior distribution. We show that having the correct parametrization of the control, we can sample the posterior with very high efficiency and observe excellent scaling with the number of observations.

[7] It is challenging to compute the ESS of the M backward trajectories in FFBSi, so we do not consider the ESS of FFBSi.

Furthermore, even with a suboptimal controller for a non-linear system the ESS increases by several orders of magnitude and the variance of the estimates is up to two orders of magnitude lower than the variance of FS and FFBSi.

We are aware of many important developments in particle methods, some of them having a linear computational complexity in the number of particles, e.g. [11], [22], [23]. However, we use the standard FFBSi to show the degeneracy of the backward pass and compare the results with APIS. We think that a comprehensive comparison of all state-of-the-art methods goes beyond the scope of this work but deserves to be addressed in the future.

More efficient proposal distributions are used in practice for FFBSi, e.g. by linearization of the discretized SDE [7]. However, in general this is only valid for sufficiently short time intervals, which might be shorter than the interval between observations and could lead to errors in the integration of (1) and the estimations. Also, similar schemes can be used to improve the efficiency of APIS.

The optimal resampling step at time $t-1$ in the $\mathcal{O}(N)$ two-filter algorithm [11] is given by marginals of the form $p(y_{t:T}|X_{t-1})w_{t-1}$ where w_{t-1} are the filter weights. This is approximated by a single observation ”look-ahead” distribution $p(y_t|X_{t-1})w_{t-1}$. It is interesting to notice that APIS effectively implements a “look-ahead” transition probability considering all observations. This makes APIS an attractive alternative to the $\mathcal{O}(N)$ two-filter algorithm.

One can apply the ideas of this paper also to discrete state, discrete time problems. A discussion on discrete systems in [38], [42] shows that we can frame discrete HMMs as a control problem. One can then adapt the uncontrolled dynamics using the Cross Entropy idea [25]. In addition, the application to continuous time hidden Markov jump processes is also interesting and possible. Both problems are equivalent to the optimization of a KL divergence similar to (7) [43]. But the details are different and one would need to work out the details on how a control theoretic approach would influence the process rates to perform importance sampling. Furthermore, details on the learning procedure applied to both types of systems and the parametrization of the feedback controller need to be worked out.

It is interesting to notice the similarities between the iterated auxiliary particle filter[8] [24] and optimal control solution in [38], [42]. The computation of the optimal twisted functions in [24] is related to the backward message passing involved in the computation of the optimal transition probability. However, the connection to optimal control was not pointed out in this work. Recall that, in practice, the functions and transition densities in [24] have to be restricted. On the contrary, APIS has great flexibility in the design of the controller. Hence, it may prove fruitful to explore the similarities between both approaches to develop better approximation schemes for the iterative auxiliary particle methods.

Similarly, it is interesting to contrast the ideas of this paper to those in [26], [27], where the aim was to minimize the KL divergence between the target density and a parametrized (mixture) model. These ideas are similar in flavor to the ideas here[9] but it is not obvious how both schemes relate. The difficulty in the comparison resides in the components learned in each method. While we aim at learning a parametrized controller to adapt the prior (uncontrolled) dynamics, [27] directly modifies the adjustment multiplier weights and the proposal kernels. These modifications may correspond to the importance control correction term and the introduction of a control term in the dynamics. However, a detailed comparison may help us to understand further the relation between adaptive importance sampling and control for general hidden state processes.

In our experiments we have initialized the importance sampler with $u(x,t)=0$. One can consider better initializations of the controller using other methods. For instance, one could initialize a linear feedback controller around the solution to the optimal trajectory as in [34]. The initialization of the controller will have an impact on the performance that is not to be taken lightly, for instance, bad importance control u might decrease the ESS and result in poor estimates which could lead to a further decrease in the quality of the controller. Our experience is, however, that a linear feedback controller initialized with the uncontrolled dynamics is a robust procedure when combined with annealing.

Naturally, the initialization of the particles at $t=0$ has an impact on the ESS. In this paper we have chosen (axis-aligned) Gaussians as proposal distributions to target the posterior marginal at $t=0$, but other initializations are possible. For instance, one can sample from a multivariate Gaussian with general covariance matrix, a general kernel density estimator to deal with multi-modal distributions or initialize the particles via the Cross Entropy method [25].

The choice of the number of particles N and the annealing threshold γ is an open question. On the one hand, we know that increasing N influences the efficiency of APIS in a non-linear way and that a value below some threshold prevents APIS from bootstrapping. However, it is not obvious how to choose N in an efficient way. On the other hand, with the annealing procedure introduced in this paper, it is possible to bootstrap APIS without increasing N to prohibitive sizes. Thus, it is important to understand how both, the number of particles and the annealing procedure, influence the learning of the control.

Unfortunately, there is no proof that APIS converges to the optimal control within the class of control solutions constrained by the parametrization. In practice we can use the ESS as a quality measure, which can be used to asses the “goodness” of the chosen parametrization. Still, the question of optimality given a parametrization is a very interesting question that deserves to be explored.

The efficiency of APIS depends on the choice of basis functions, which is problem dependent. This choice is an open question. The linear basis functions that we considered in this paper are very robust to learn, however, they might lead to problems whenever the posterior has multiple pronounced modes. Since the update rule (18) poses no restriction on the basis functions,

[8] We are grateful to a reviewer for pointing out this work and the similarity between APIS and this method.

[9] We are grateful to a reviewer for mentioning this approach and its possible relation to APIS for discrete HMMs.

more complex functions are possible. Thus, in problems with multiple modes, locally linear functions such as $x\exp(-x^2)$ may be a good choice.

More general strategies to estimate an efficient importance control are possible. For instance, we can use a nested set of functions. During the initial iterations a simple function (say linear) is learned and as soon as the ESS is sufficiently high, non-linear extensions are learned. In this context the application of universal function approximations such as deep neural networks may be promising.

Appendix
Implementation Details

The proposed APIS method was discussed in III-B in general form to remark that in principle any linear parametrization of the controller can be learned. Here, we discuss the implementation details for the results in IV.

We use a linear feedback controller standardized w.r.t. the target distribution, i.e. $h(x,t) := (1, z(x,t))'$, where $z(x,t), x \in \mathbb{R}^m$. Each component z_i is defined as

$$z_i(x,t) = \frac{x_i - \mu_i(t)}{\sigma_i(t)}$$

where $\mu_i(t) = \langle X_{t,i}\rangle, \sigma_i^2(t) = \langle (X_{t,i} - \mu_i(t))^2\rangle; i = 1,\ldots,m$ are the mean and variance of the state components w.r.t. smoothing marginal at time t. The values are initialized in the first iteration as $\mu_i(t) = 0$ and $\sigma_i^2(t) = 1$ for all times. This choice of basis functions splits (18) such that the updates for the open-loop and feedback controllers are independent and numerically more stable.

For completeness, we give the explicit update rules for the standardized linear feedback controller. The control has a very simple form $u(x,t) = a(t)z(x_t,t) + b(t)$, where $a(t) \in \mathbb{R}^{m\times m}$ is a square matrix of the same dimension as the state and $b(t) \in \mathbb{R}^m$ is an open-loop controller. Then, the cross-correlation matrix becomes

$$H(t) = \begin{bmatrix} 1 & 0 \\ 0 & C(t) \end{bmatrix}$$

where $C(t)$ is the correlation matrix of the state variables. We have component-wise,

$$C^{\text{ij}}(t) = \left\langle \frac{[X_{t,i} - \mu_i(t)][X_{t,j} - \mu_j(t)]}{\sigma_i(t)\sigma_j(t)} \right\rangle_r .$$

For $dQ_r(h_t)$ we have a matrix in $\mathbb{R}^{m\times(m+1)}$ with elements

$$[dQ_r(1)]^{i1} = \langle dW_{t,i}\rangle_r$$
$$[dQ_r(z_t)]^{i(j+1)} = \left\langle dW_{t,i}\frac{X_{t,j} - \mu_j(t)}{\sigma_j(t)} \right\rangle_r$$

for each $i, j = 1,\ldots,m$.

This gives the explicit update rules,

$$b^{r+1}(t) = b^r(t) + \eta\frac{\langle dW_t\rangle_r}{dt}$$
$$a^{r+1}(t) = a^r(t) + \eta\frac{dQ_r(z_t)}{dt}C^{-1}(t)$$

We use as prior distribution $p_0(X_0)$ for the state at time $t = 0$ a Gaussian with mean and variance μ_0, σ_0^2, respectively. In addition, we use an adaptive Gaussian as proposal distribution $q(X_0)$. Thus, we initialize at each iteration of APIS the cost to correct for this importance sampling step. This initial value is given for particles $l = 1,\ldots,N$ by

$$S^{0,l} = \sum_{i=1}^{m} \frac{(X_{0,i}^l - \hat{\mu}_{0,i})^2}{2\hat{\sigma}_{0,i}^2} - \frac{(X_{0,i}^l - \mu_{0,i})^2}{2\sigma_{0,i}^2}$$

where $\hat{\mu}_0 = \mu(0)$ and $\hat{\sigma}_{0,i}^2 = \sigma_i(0)^2$ are the mean and variance of the posterior marginal at $t = 0$.

The control cost over the entire interval $[0,T]$ is given by the linear feedback controller and approximated by the discretization step dt,

$$\int_0^T \frac{1}{2}\|u(X_s,s)\|^2\,ds + \int_0^T u(X_s,s)'dW_s$$
$$\approx \sum_{i=1}^{L} \|a(t_i)z_{t_i} + b(t_i)\|^2\frac{dt}{2} + (a(t_i)z_{t_i} + b(t_i))'\,dW_{t_i}$$

where $z_{t_i} = z(X_{t_i}, t_i)$ and dW_{t_i} the noise realization at time t_i with variance $\sigma^2 = dt$. The summation goes over the $L = T/dt$ integration steps.

In general, there is a trade-off between the amount of iterations and the number of particles needed, but we observe that the convergence of the ESS to a maximal value is very fast once it has bootstrapped, so usually a number of iterations $I_{\max} \ll N$ can be chosen and–as a rule of thumb–higher N allows for less iterations. The particles can be sampled independently so this step is parallelizable. Nevertheless, when reducing the learning rate, it is possible to reduce by at least one order of magnitude the number of particles needed to bootstrap APIS. Naturally, this will increase the number of iterations needed. In addition, we found that an annealing procedure with γ in the range $0.02 - 0.05$ works well for low N.

Finally, for FFBSi and FS, we use the algorithms as described in [8, Algorithm 4] with the numerical integration of the SDE (1) as proposal distribution for the filtering. We initialized particles according to $p_0(X_0)$ unless noted otherwise.

Acknowledgment

The first author would like to thank S. Thijssen, D. Thalmeier, and G. S. Stamatescu for helpful discussions. He would also like to show his gratitude to S. Menchon and the reviewers for their useful comments on an earlier version of the manuscript.

References

[1] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. Int. Symp. Aerospace/Defense Sensing, Simul. Controls*, 1997, pp. 182–193.

[2] S. Sarkka, "On unscented Kalman filtering for state estimation of continuous-time nonlinear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1631–1641, Sep. 2007.

[3] S. Särkkä, "Unscented Rauch–Tung–Striebel smoother," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 845–849, Apr. 2008.

[4] T. Sutter, A. Ganguly, and H. Koeppl, "A variational approach to path estimation and parameter inference of hidden diffusion processes," *J. Mach. Learn. Res.*, vol. 17, 2016.

[5] G. Kitagawa, “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,” *J. Comput. Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[6] S. J. Godsill, A. Doucet, and M. West, “Monte Carlo smoothing for nonlinear time series,” *J. Amer. Statist. Assoc.*, vol. 99, no. 465, pp. 156–168, Mar. 2004. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1198/016214504000000151

[7] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, Jul. 2000.

[8] F. Lindsten, “Backward simulation methods for Monte Carlo statistical inference,” *Foundations Trends Mach. Learning*, vol. 6, no. 1, pp. 1–143, 2013. [Online]. Available: http://www.nowpublishers.com/articles/foundations-and-trends-in-machine -learning/MAL-045

[9] Y. Bresler, “Two-filter formulae for discrete-time non-linear Bayesian smoothing,” *Int. J. Control*, vol. 43, no. 2, pp. 629–641, 1986.

[10] M. Briers, A. Doucet, and S. Maskell, “Smoothing algorithms for state-space models,” *Ann. Inst. Statist. Math.*, vol. 62, no. 1, pp. 61–89, Feb. 2010. [Online]. Available: http://link.springer.com/10.1007/s10463-009-0236-2

[11] P. Fearnhead, D. Wyncoll, and J. Tawn, “A sequential smoothing algorithm with linear computational cost,” *Biometrika*, vol. 97, no. 2, pp. 447–464, Jun. 2010. [Online]. Available: http://biomet.oxfordjournals.org/cgi/doi/10.1093/biomet/asq013

[12] A. Doucet and A.M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later,” *Handbook Nonlinear Filtering*, vol. 12, pp. 656–704, 2009.

[13] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Berlin, Germany: Springer Science & Business Media, 2008.

[14] P. Fearnhead, “Computational methods for complex stochastic systems: A review of some alternatives to MCMC,” *Statist. Comput.*, vol. 18, no. 2, pp. 151–171, 2008.

[15] P. Del Moral *et al.*, “On adaptive resampling strategies for sequential Monte Carlo methods,” *Bernoulli*, vol. 18, no. 1, pp. 252–278, 2012.

[16] N. Chopin, “Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference,” *Ann. Stat.*, vol. 32, no. 6, pp. 2385–2411, 2004.

[17] R. Douc, A. Garivier, E. Moulines, and J. Olsson, “Sequential Monte Carlo smoothing for general state space hidden Markov models,” *Ann. Appl. Probability*, vol. 21, no. 6, pp. 2109–2145, Dec. 2011. [Online]. Available: http://projecteuclid.org/euclid.aoap/1322057317

[18] J. Westerborn and J. Olsson, “Efficient particle-based online smoothing in general hidden Markov models,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 8003–8007.

[19] P. E. Kloeden, E. Platen, and H. Schurz, *Numerical Solution of SDE Through Computer Experiments*. Berlin, Germany: Springer-Verlag, 2012.

[20] L. Murray and A. Storkey, “Particle smoothing in continuous time: A fast approach via density estimation,” *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1017–1026, Mar. 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5654660

[21] E. Taghavi, “A study of linear complexity particle filter smoothers,” Master’s thesis, Signals and Systems Dept., Signal Processing Group, Chalmers Univ. Technology, 2012. [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/156741.pdf

[22] C. Dubarry and R. Douc, “Improving particle approximations of the joint smoothing distribution with linear computational cost.” IEEE, Jun. 2011, pp. 209–212. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5967661

[23] P. Bunch and S. Godsill, “Improved particle approximations to the joint smoothing distribution using Markov chain Monte Carlo,” *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 956–963, Feb. 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6359869

[24] P. Guarniero, A. M. Johansen, and A. Lee, “The iterated auxiliary particle filter,” *J. Amer. Statist. Assoc.*, accepted for publication, 2016.

[25] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A tutorial on the cross-entropy method,” *Ann. Oper. Res.*, vol. 134, no. 1, pp. 19–67, 2005.

[26] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert, “Adaptive importance sampling in general mixture classes,” *Statist. Comput.*, vol. 18, no. 4, pp. 447–459, 2008.

[27] J. Cornebise, É. Moulines, and J. Olsson, “Adaptive methods for sequential importance sampling with application to state space models,” *Statist. Comput.*, vol. 18, no. 4, pp. 461–480, 2008.

[28] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *J. Amer. Statist. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.

[29] E. Pardoux, “The solution of the nonlinear filtering equation as a likelihood function.” IEEE, Dec. 1981, pp. 316–319. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4046946

[30] W. H. Fleming and S. K. Mitter, “Optimal control and nonlinear filtering for nondegenerate diffusion processes,” *Stochastics*, vol. 8, no. 1, pp. 63–77, 1982.

[31] S. K. Mitter, *Nonlinear Filtering of Diffusion Processes a Guided Tour* (Lecture Notes in Control and Information Sciences 42). New York, NY, USA: Springer-Verlag, 1982, ch. 23, pp. 256–266. [Online]. Available: http://www.springerlink.com/index/10.1007/BFb0004544

[32] H. J. Kappen, “Linear theory for control of nonlinear stochastic systems,” *Phys. Rev. Lett.*, vol. 95, no. 20, Nov. 2005. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.95.200201

[33] T. Yang, P. G. Mehta, and S. P. Meyn, “Feedback particle filter,” *IEEE Trans. Autom. Control*, vol. 58, no. 10, pp. 2465–2480, Oct. 2013.

[34] S. Pequito, P. Aguiar, B. Sinopoli, and D. Gomes, “Nonlinear estimation using mean field games,” in *Proc. Int. Conf. NETw. Games, Control Optim*. IEEE, 2011, pp. 1–5.

[35] S. Thijssen and H. J. Kappen, “Path integral control and state-dependent feedback,” *Phys. Rev. E*, vol. 91, no. 3, Mar. 2015. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.91.032104

[36] H. J. Kappen and H. C. Ruiz, “Adaptive importance sampling for control and inference,” *J. Statist. Phys.*, vol. 162, no. 5, pp. 1244–1266, 2016.

[37] H. J. Kappen, “Optimal control theory and the linear Bellman equation,” in *Inference Learning Dynamic Models*, pp. 363–387, 2011. [Online]. Available: http://hdl.handle.net/2066/94184

[38] H. Kappen, V. Gómez, and M. Opper, “Optimal control as a graphical model inference problem,” *Mach. Learn.*, vol. 87, no. 2, pp. 159–182, May 2012. [Online]. Available: http://link.springer.com/10.1007/s10994-012-5278-7

[39] W. Zhang, H. Wang, C. Hartmann, M. Weber, and C. Schuette, “Applications of the cross-entropy method to importance sampling and optimal control of diffusions,” *SIAM J. Sci. Comput.*, vol. 36, no. 6, pp. A2654–A2672, Jan. 2014. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/14096493X

[40] I. V. Girsanov, “On transforming a certain class of stochastic processes by absolutely continuous substitution of measures,” *Theory Probab. Appl.*, vol. 5, no. 3, pp. 285–301, 1960.

[41] J. S. Liu, “Metropolized independent sampling with comparisons to rejection sampling and importance sampling,” *Statist. Comput.*, vol. 6, no. 2, pp. 113–119, 1996.

[42] E. Todorov, “Linearly-solvable Markov decision problems,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1369–1376.

[43] M. Opper and G. Sanguinetti, “Variational inference for Markov jump processes,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1105–1112.

Hans-Christian Ruiz received the degree Diplom-Physiker in theoretical physics from the University of Munich in 2011. In 2012, he worked on generalizations of spin networks during an internship at the Max-Planck Institute for Gravitational Physics, Potsdam, Germany. He is currently working as a Ph.D. student in the Donders Institute, Nijmegen, The Netherlands. His research interests include stochastic optimal control, Bayesian inference, and neural networks. He received a Marie Curie fellowship of the Neural Engineering Transformative Technologies Initial Training Network.

Hilbert J. Kappen received the Ph.D. degree in particle physics from the Rockefeller University, New York, USA, in 1987. From 1987 to 1989, he worked as a Scientist in the Philips Research Laboratories, Eindhoven, the Netherlands. Since 1989, he has been conducting research on neural networks at the laboratory for biophysics of the University of Nijmegen, the Netherlands. Since 1997, he has been an Associate Professor and, since 2004, a Full Professor at this university. His group is involved in research on Bayesian machine learning, stochastic control theory, computational neuroscience, and several applications in collaboration with industry. His research was awarded in 1997 the prestigious national PIONIER research subsidy. In 1998, he co-founded the company Smart Research, which sell prediction software based on neural networks. He has developed a medical diagnostic expert system called Promedas, which assists doctors to make accurate diagnosis of patients. He is the Director of the Dutch Foundation for Neural Networks (SNN), which coordinates research on neural networks and machine learning in the Netherlands. Since 2009, he has been an honorary faculty in Gatsby Computational Neuroscience Unit, University College London, London, U.K.