International Journal of Control

ISSN: 0020-7179 (Print) 1366-5820 (Online) Journal homepage: http://www.tandfonline.com/loi/tcon20

Learning effective state-feedback controllers through efficient multilevel importance samplers

S. A. Menchón & H. J. Kappen

To cite this article: S. A. Menchón & H. J. Kappen (2018): Learning effective state-feedback controllers through efficient multilevel importance samplers, International Journal of Control, DOI: 10.1080/00207179.2018.1459857

To link to this article: https://doi.org/10.1080/00207179.2018.1459857

Published online: 21 May 2018.

INTERNATIONAL JOURNAL OF CONTROL, 2018
https://doi.org/10.1080/00207179.2018.1459857

Taylor & Francis
Taylor & Francis Group

Learning effective state-feedback controllers through efficient multilevel importance samplers

S. A. Menchón [a,b] and H. J. Kappen[b]

[a]IFEG-CONICET and FaMAF-Universidad Nacional de Córdoba, Ciudad Universitaria, Córdoba, Argentina; [b]SNN Machine Learning Group, Biophysics Department, Donders Institute for Brain Cognition and Behavior, Radboud University, Nijmegen, The Netherlands

ABSTRACT

Monte Carlo sampling can be used to estimate the solution of path integral control problems, which are a restricted class of nonlinear control problems with arbitrary dynamics and state cost, but with a linear dependence of the control on the dynamics and quadratic control cost. Although importance sampling is used to improve numerical computations, the effective sample size may still be low or many samples could be required. In this work, we propose a method to learn effective state-feedback controllers for nonlinear stochastic control problems based on multilevel importance samplers. In particular, we focus on the question of how to compute effective importance samplers considering a multigrid scenario. We test our algorithm in finite horizon control problems based on Lorenz-96 model with chaotic and non-chaotic behaviour, showing, in all cases, that our multigrid implementation reduces the computational time and improves the effective sample size.

ARTICLE HISTORY
Received 8 August 2017
Accepted 22 March 2018

KEYWORDS
Path integral control problems; multilevel Monte Carlo method; importance sampling

1. Introduction

When the system dynamics is linear and the cost is quadratic (LQ control), the solution of the stochastic control problem is given in terms of a number of coupled ordinary differential equations that can be solved efficiently (Stengel, 1993). Although LQ control is useful in some situations, it is a linear theory and it does not allow to model complex systems. However, there is a class of continuous nonlinear stochastic control problems that can be solved more efficiently than the general case (Kappen, 2005). These are control problems with a finite time horizon, where the control acts linearly and additive on the dynamics and the control cost is quadratic. These control problems are essentially reduced to the computation of a path integral. Since path integrals involve an expected value with respect to a dynamical system, the optimal control can be estimated by using Monte Carlo (MC) sampling. In order to efficiently compute the optimal control, samples might be generated from a different distribution by importance sampling (IS). It has also been shown an intimate relation between optimal IS and optimal control (Thijssen & Kappen, 2015). The optimal control solution is related with the optimal sampler, and better samplers in terms of effective sample size (ESS) are also better controllers in terms of control cost. This allows to iteratively improve the importance sampler.

The weakness of MC simulation is that its computational cost can be very high; in particular, this is the case when each sample might require many time steps. In order to improve the computational efficiency, Giles introduced the multilevel Monte Carlo (MLMC) method (Giles, 2008). This method considers a geometric sequence of time discretisations (grids). On the coarsest grid, the accuracy of the approximate solution of a partial differential equation and its computational cost are low. On the other hand, on a finer grid, its accuracy is greater, but so is its cost. The multigrid approach of MLMC combines the solution for all levels in a clever manner reducing the total computational cost and the estimate variance. Since MLMC estimator involves independent MC estimators at each grid level, IS for each level can be considered independent on the others. This feature and the fact that IS estimators are unbiased give some freedom about how to implement IS for the MLMC method. For instance, the same importance sampler can be considered for all levels, or each level may have its own importance sampler.

In this work, we discuss how to implement efficiently multilevel IS on the computation of effective controllers. In particular, we show that implementations using the same importance sampler for all levels give more accurate controller updates; and implementations considering an importance sampler for each level contribute with fast controller updates at the early iteration steps.

In the next sections, we review path integral control theory in the finite horizon case, the cross entropy method as method for adaptive IS and the MLMC method. Subsequently, we introduce two algorithms for applying IS to MLMC and test their efficiency in the search of an optimal controller. In particular, we apply our algorithms to particle smoothing problems considering dynamical systems based on Lorenz-96 model and one observation. Motivated by our results, we propose a combined algorithm, which has a much better performance than MC, to estimate solutions of path integral control problems. A significant increase in the final ESS and a reduction of the CPU time are the reasons that make our combined algorithm a powerful tool to find better samplers; and therefore, better controllers.

CONTACT S. A. Menchón silvia.menchon@unc.edu.ar

2. Path integral control

Consider the dynamical system

$$dX(t) = f(t, X(t))dt + g(t, X(t))\left[u(t, X(t))dt + dW(t)\right], \tag{1}$$

for $t_0 \le t \le T$ with initial condition $X(t_0) = x_0$, where $X(t) \in \mathbb{R}^n$, $f(t, X(t)) : [t_0, T] \times \mathbb{R}^n \to \mathbb{R}^n$, $g(t, X(t)) : [t_0, T] \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$, $u(t, X(t)) : [t_0, T] \times \mathbb{R}^n \to \mathbb{R}^m$ and $dW(t)$ is an m-dimensional Gaussian noise with $\mathbb{E}[dW(t)] = 0$ and $\mathbb{E}[dW(t)dW(\hat{t})] = \nu dt\delta(t - \hat{t})$, here ν is an $m \times m$ positive definite covariance matrix, and δ is Dirac's function. Given a function $u(t, x)$ that defines the control for each state x at each time $t \in [t_0, T]$, the cost function, S, is defined by

$$\begin{aligned} S(t, x, u) = \frac{1}{\lambda}\Bigg(\Phi(X(T)) + \int_t^T \Bigg(V(s, X(s)) \\ + \frac{1}{2}u(s, X(s))' R u(s, X(s))\Bigg)ds \\ + \int_t^T u(s, X(s))' R dW(s)\Bigg), \end{aligned} \tag{2}$$

where t and x are the current time and state and $'$ denotes transpose. The cost function consists of an end cost, $\Phi(x)$, that gives the cost of ending in the configuration x, and a path cost that is an integral over the time trajectories. In this work, we consider Φ and V to be bounded from below and piecewise-defined functions. In this case, ν and R are related by $\lambda I = R\nu$, with $\lambda > 0$ a scalar (Kappen, 2005).

The main goal is to find the optimal control, u^*, that minimises the expected cost:

$$\begin{aligned} J(t, x) &= \min_u \mathbb{E}[S(t, x, u)], \\ u^* &= arg\min_u \mathbb{E}[S(t, x, u)], \end{aligned} \tag{3}$$

where $\mathbb{E}$ denotes the expected value with respect to the stochastic problem from Equation (1) with initial condition $X(t_0) = x_0$ and control u. The optimal cost-to-go, $J(t, x)$, is the optimal cost for any intermediate time t until the fixed end time T, starting at any intermediate state x. The optimal cost-to-go satisfies a partial differential equation known as the stochastic Hamilton–Jacobi–Bellman equation, whose solution, in the particular case of path integral control problems, is given by the following theorem, proved by Thijssen and Kappen (2015):

Theorem 2.1: *The solution of the control problem specified by Equation (3) is given by*

$$J(t, x) = -\log(\psi(t, x)), \tag{4}$$

$$u^*(t, x) = u(t, x) + \lim_{s\to t} \frac{1}{s - t} \frac{\mathbb{E}[dW(s)\exp(-S(t, x, u))]}{\mathbb{E}[\exp(-S(t, x, u))]}, \tag{5}$$

where $\psi(t, x) = \mathbb{E}[\exp(-S(t, x, u))]$.

Theorem 2.1 not only gives an expression to obtain the optimal control solution, u^*, it also shows that we can use any function u to compute it. In order to solve the control problem numerically, we need to estimate the expected values in Equations (4) and (5). In general, the easiest way to do it is by MC sampling. We refer to the control u as the sampling control, when implementing numerical computations. Of course, the choice of u affects the efficiency of MC sampling. Thus, it is very important to choose an appropriate sampling control and iteratively improve it, otherwise the efficiency of MC sampling would be very poor.

In order to improve the sample efficiency, we can use IS. Consider the probability distribution $p(x)$ and suppose that we are interested in the expected value of $\gamma(x)$, i.e. we want to estimate the quantity:

$$\mathbb{E}_p[\gamma(X)] = \int_\Omega \gamma(x)p(x)dx,$$

where Ω represents the support of p. The naive MC estimator is given by

$$\hat{\gamma} = \frac{1}{N}\sum_{i=1}^N \gamma(X^{(i)}),$$

where the N samples are drawn from p. If p does not have much weight on the support of $\gamma(x)$, only a few samples will contribute to the computation of the estimator. However, we can consider another distribution $q(x)$ with support Υ, such that $\Omega \subseteq \Upsilon$, and such that q has much more weight on the support of $\gamma(x)$ than p. The expected value $\mathbb{E}_p[\gamma(X)]$ can be expressed in terms of $q(x)$ by

$$\mathbb{E}_p[\gamma(X)] = \int_\Upsilon \gamma(x)\frac{p(x)}{q(x)}q(x)dx = \mathbb{E}_q\left[\gamma(X)\frac{p(X)}{q(X)}\right], \tag{6}$$

and its MC estimator, which is also unbiased, is

$$\hat{\gamma} = \frac{1}{N}\sum_{i=1}^N \gamma(X^{(i)})\frac{p(X^{(i)})}{q(X^{(i)})},$$

where the N samples are now drawn from q. The basic idea of IS is to sample from a different distribution to reduce the variance of the estimator $\mathbb{E}_p[\gamma(X)]$ or because sampling from p is difficult (for instance, in Section 4, we use IS because we do not know the optimal distribution p^* explicitly). The ratio $p(x)/q(x)$ is referred to as Radon–Nikodym derivative. An important measure of the efficiency of applying IS using q is the normalised ESS, that is defined by $ESS = 1/(1 + \mathbb{V}_q[p(X)/q(X)])$ (Kong, 1992; Liu, 1996). The optimal importance sampler has zero variance, its ESS is one, and it is given by

$$q^*(x) = \frac{\gamma(x)p(x)}{\mathbb{E}[\gamma(X)]}.$$

However, we cannot use it in practice, since it requires prior knowledge of $\mathbb{E}[\gamma(X)]$, which is the quantity we want to compute. Nevertheless, our main goal is to give a numerical expression for Equations (4) and (5), i.e. we want to compute good estimators for the expected values in those equations by sampling from an efficient sampling control u. Although finding the optimal control and the most efficient sampling control seems to be two different problems, Thijssen and Kappen (2015) proved that the best control in terms of optimal control is also the best control in terms of sampling control. In the next section, we review the construction of an adaptive algorithm to find an effective controller.

3. Path integral control and cross entropy method

In this section, we review the cross entropy method. The results shown in this section have already been discussed by Kappen and Ruiz (2016), but we decided to include them with the purpose of having a self-contained article. In order to apply the cross entropy method (de Boer, Kroese, Mannor, & Rubinstein, 2005) to the path integral control theory, it is necessary to reformulate the control problems in terms of a Kullback–Leibler (KL) divergence. In the limit $dt \to 0$, the conditional probability of $X(t+dt)$ given $X(t)$ is a Gaussian with mean $\mu_t = X(t) + f(t, X(t))dt + g(t, X(t))u(t, X(t))dt$ and variance $\Xi_t dt = g(t, X(t))\nu g(t, X(t))' dt$. Therefore, the conditional probability of a trajectory $\tau = X_{t_0:T}|x_0$ with initial state $X(t_0) = x_0$ is given by

$$p_u(\tau) = p_0(\tau)\exp\left(\int_{t_0}^{T} -\frac{1}{2}u(t, X(t))'\nu^{-1}u(t, X(t))dt + \int_{t_0}^{T}(dX - f(t, X(t))dt)'\Xi_t^{-1}g(t, X(t))u(t, X(t))\right), \tag{7}$$

where p_0 is the distribution over trajectories for the uncontrolled dynamics (Kappen, Gómez, & Opper, 2012; Kappen & Ruiz, 2016). The quadratic control cost in the path integral control problem represented by Equation (3) can be expressed as a KL divergence by combining Equations (3) and (7). Thus,

$$\begin{aligned} J(t, x) &= \min_u \mathbb{E}[S(t, x, u)] \\ &= \min_u \int p_u(\tau)\left(\log\left(\frac{p_u(\tau)}{p_0(\tau)}\right) + \hat{V}(\tau)\right), \end{aligned} \tag{8}$$

where $\hat{V} = 1/\lambda(\Phi(X(T)) + \int_{t_0}^{T} V(t, X(t))dt)$. Since there is one-to-one correspondence between u and p_u, we can replace the minimisation with respect to u by a minimisation with respect to p_u subject to the normalisation constrain $\int p_u(\tau)d\tau = 1$. Taking this minimisation, the optimal solution, p^*, can be expressed in terms of p_u, (by using Equation (7) to relate p_0 with p_u), as follows:

$$p^*(\tau) = \frac{1}{\psi(t, x)}p_u(\tau)\exp(-S(t, x, u)). \tag{9}$$

Although we have an expression for the optimal solution, we cannot compute it, because it would require prior knowledge of $\psi(t, x) = \mathbb{E}[\exp(-S(t, x, u))]$. However, we can compute a near-optimal control $\hat{u}$, such that $p_{\hat{u}}$ is close to p^*. Following the cross entropy method, we minimise the KL divergence

$$\begin{aligned} KL(p^*|p_{\hat{u}}) &\propto -\mathbb{E}_{p^*}\log(p_{\hat{u}}), \\ &\propto \mathbb{E}_{p^*}\left[\int_{t_0}^{T}\frac{1}{2}\hat{u}(t, X(t))'\nu^{-1}\hat{u}(t, X(t))dt \right. \\ &\quad \left. - (dX - f(t, X(t))dt)'\Xi_t^{-1}g(t, X(t))\hat{u}(t, X(t))\right], \\ &\propto \frac{1}{\psi(t, x)}\mathbb{E}_{p_u}\left[\exp(-S(t, x, u)) \right. \\ &\quad \times \int_{t_0}^{T}\left(\frac{1}{2}\hat{u}(t, X(t))'\nu^{-1}\hat{u}(t, X(t)) \right. \\ &\quad \left.\left. - \left(u(t, X(t)) + \frac{dW(t)}{dt}\right)'\nu^{-1}\hat{u}(t, X(t))\right)dt\right], \end{aligned} \tag{10}$$

with respect to the functions $\hat{u}_{t_0:T} = \{\hat{u}(t, X(t)), t_0 \le t \le T\}$. In order to obtain the final expression of Equation (10), we have discarded the constant terms and expressed the expectation with respect to the optimal distribution p^* controlled by u^* in terms of an arbitrary distribution p_u controlled by u. In particular, we have used Equation (7) (with $u = \hat{u}$) and Equation (9) (to change $\mathbb{E}_{p^*}$ by $\mathbb{E}_{p_u}$) in the second and last lines, respectively. In the last step, we have used IS due to the fact that we cannot compute the expected value under the unknown distribution p^*, but it is possible to do it under p_u, for any arbitrary u.

We can compute the gradient of the KL divergence assuming that $\hat{u}$ is a parametrised function with the parameter θ. Thus,

$$\frac{\partial KL(p^*|p_{\hat{u}})}{\partial\theta} = \left\langle \int_{t_0}^{T}\left(\hat{u}(t, X(t)) - u(t, X(t)) - \frac{dW(t)}{dt}\right)' \times \nu^{-1}\frac{\partial\hat{u}(t, X(t))}{\partial\theta}dt\right\rangle_u, \tag{11}$$

where we have introduced the notation $\langle F\rangle_u = \psi(t, x)^{-1}\mathbb{E}_{p_u}[\exp(-S(t, x, u))F]$. The equation above represents the gradient in the point $\hat{u}(t, X(t))$ for an arbitrary $u(t, X(t))$. Since the KL divergence is a nonlinear function of θ, we can minimise it by using a gradient-based procedure. Thus, we expect that $\hat{u}(t, X(t))$ improves in each iteration and since a better control in terms of optimal control is also a better control in terms of sampling control (Thijssen & Kappen, 2015), its current estimated value is a good candidate for $u(t, X(t))$. In this case,

$$\frac{\partial KL(p^*|p_{\hat{u}})}{\partial\theta} = -\left\langle \int_{t_0}^{T} dW(t)'\nu^{-1}\frac{\partial\hat{u}(t, X(t))}{\partial\theta}\right\rangle_{\hat{u}}, \tag{12}$$

and the gradient descent update at iteration n is given by

$$\begin{aligned} \theta_{n+1} &= \theta_n - \eta\frac{\partial KL(p^*|p_{\hat{u}})}{\partial\theta_n} \\ &= \theta_n + \eta\left\langle \int_{t_0}^{T} dW(t)'\nu^{-1}\frac{\partial\hat{u}(t, X(t))}{\partial\theta}\right\rangle_{\hat{u}}, \end{aligned} \tag{13}$$

with $\eta > 0$ a small parameter.

4. MLMC and IS

MLMC method combines MC path simulations with different time steps through a telescopic sum. It was introduced by Giles to be applied to a variety of financial models (Giles, 2008). Here we include the main theorem of Giles' work, which is quite general; and we keep his notation. Afterward, we identify the functionals that appear at the main theorem with the functionals we are interested in, i.e. the functionals that are present in our adaptive algorithm (Equation (13)). Although we include the most important information for our work, we suggest readers to consult Giles' paper for more specific details about MLMC method.

According to Section 3 of Giles (2008), let $h_l = h_0/M^l$, $l = 0, 1, \ldots, L$ be the time step for level l, where h_0 corresponds to the initial discretisation. Thus, the MLMC theorem states:

Theorem 4.1: *Let P denote a functional of the solution of the stochastic differential equation (1) for a given Brownian path $W(t)$ and let $\hat{P}_l$ denote a corresponding approximation using a numerical discretisation with time step h_l.*

If there exist independent estimators $\hat{Y}_l$ based on N_l MC samples, and positive constants $\alpha \geq 0.5$, β, c_1, c_2 and c_3, such that

(1) $\mathbb{E}[\hat{P}_l - P] \leq c_1 h_l^{\alpha}$
(2) $\mathbb{E}[\hat{Y}_l] = \begin{cases} \mathbb{E}[\hat{P}_0], & l = 0 \\ \mathbb{E}[\hat{P}_l - \hat{P}_{l-1}], & l > 0 \end{cases}$
(3) $\mathbb{V}[\hat{Y}_l] \leq c_2 N_l^{-1} h_l^{\beta}$
(4) *C_l the computational complexity of $\hat{Y}_l$, is bounded by $C_l \leq c_3 N_l h_l^{-1}$,*

then there is a constant c_4 such that for any $\epsilon < e^{-1}$, there are values L and N_l for which the multilevel estimator $\hat{Y} = \sum_{l=0}^{L} \hat{Y}_l$ has a mean square error (MSE) with bound $MSE \equiv \mathbb{E}[(\hat{Y} - \mathbb{E}[P])^2] < \epsilon^2$, with a computational complexity C with bound

$$C \leq \begin{cases} c_4 \epsilon^{-2}, & \beta > 1 \\ c_4 \epsilon^{-2} (\log(\epsilon))^2, & \beta = 1 \\ c_4 \epsilon^{-2-(1-\beta)/\alpha} & 0 < \beta < 1 \end{cases}$$

The simplest estimators $\hat{Y}_l$ are given by the means of N_l independent samples. Thus,

$$\hat{Y}_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} \hat{P}_0^{(i)}, \text{ and } \hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)} \right), \quad (14)$$

with $l = 1, \ldots, L$. It is important to remark that the quantity $\hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)}$ comes from two discrete approximations with different time steps, but the same Brownian path. Observe that they are not the same Brownian paths for all l, they are different Brownian paths for each $\hat{Y}_l$, i.e. $\hat{P}_l^{(i)}$ comes from a discrete approximation with a given Brownian path for the computation of $\hat{Y}_l$, but with a different Brownian path for $\hat{Y}_{l+1}$, (see Giles, 2008). In order to equilibrate statistical and spatio-temporal discretisation errors, the number of samples on mesh level l is related with the number of samples of level L (Giles, 2008; Šukys, 2014) by the following equation:

$$N_l = N_L M^{(\beta+1)(L-l)/2}. \quad (15)$$

Using MLMC and an Euler scheme to solve the involved stochastic differential equation, the computational complexity and the variance are reduced (in comparison to those for MC), leaving unchanged the bias due to the Euler discretisation. Although MLMC reduces the variance and the computational complexity, it does not solve the problem of sample efficiency. However, we can combine MLMC with IS.

In this work, Equation (13) represents the main step of our adaptive algorithm, which updates the sampling control. At each adaptive step, we have a different sampling control and, therefore, a different distribution $q(x)$ for applying IS. In our particular case, p and q from Section 2 are related with p^* and $p_{\hat{u}}$ from Section 3, respectively; and our functional, P, is $\int_{t_0}^{T} dW(t)' \nu^{-1} \partial \hat{u}(t, X(t))/\partial\theta$. From Equation (9), we know that the Radon–Nikodym derivative is given by $\exp(-S(t, x, \hat{u}))/\psi(t, x)$. Since in a numerical approach, the Radon–Nikodym derivative depends on the control $\hat{u}$ as well as on the discretisation, we denote it as $r_{\hat{u},l}$, when using the discretisation of level l.

There are two obvious ways to implement the gradient descent update using MLMC and IS; it can be done updating all the levels at the same time, or updating each level independently. In both cases, the telescopic sum, that is involved in MLMC, remains valid due to the IS unbiased estimators.

If we implement IS through all the levels at the same time, the MLMC estimator is given by

$$\hat{Y}_{IS,L} = \frac{1}{N_0} \sum_{i=1}^{N_0} \hat{P}_0^{(i)} r_{\hat{u},0}^{(i)} + \sum_{l=1}^{L} \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\hat{P}_l^{(i)} r_{\hat{u},l}^{(i)} - \hat{P}_{l-1}^{(i)} r_{\hat{u},l-1}^{(i)} \right). \quad (16)$$

In other words, we compute the expected value in Equation (13) using all levels of discretisation, and then we update the control parameters. This is presented with more detail in the pseudocode Algorithm 1. In this case, the same controller $\hat{u}$ is used to sample at all levels.

Algorithm 1. MLMC and IS updating all levels at the same time

Require: Dynamic equation (i.e. Equation (1)), $S(t, x, u)$, $\hat{u}$ (i.e. θ_1 and $h(t, x)$), M, L, $N_{\max}$, ζ, N_L
  while $n < N_{\max}$ **or** $\Delta ESS < \zeta$ **do**
    Perform MLMC until level L, with $N_L M^{(\beta+1)(L-l)/2}$ samples at level $l = 0 \cdots L$
    Compute $\mathbb{E}[P_0 r_{\hat{u},0}]$, $\mathbb{V}[P_0 r_{\hat{u},0}]$, $\mathbb{E}[P_l r_{\hat{u},l} - P_{l-1} r_{\hat{u},l-1}]$ and $\mathbb{V}[P_l r_{\hat{u},l} - P_{l-1} r_{\hat{u},l-1}]$ for $l = 1 \cdots L$
    Compute $\mathbb{E}[P] = \mathbb{E}[P_0 r_{\hat{u},0}] + \sum_{l'=1}^{L} \mathbb{E}[P_{l'} r_{\hat{u},l'} - P_{l'-1} r_{\hat{u},l'-1}]$
    Compute $\mathbb{V}[P] = \mathbb{V}[P_0 r_{\hat{u},0}] + \sum_{l'=1}^{L} \mathbb{V}[P_{l'} r_{\hat{u},l'} - P_{l'-1} r_{\hat{u},l'-1}]$
    Update the control $\hat{u} : \theta_{n+1} \leftarrow \theta_n + \eta \mathbb{E}[P]$
  end while

Algorithm 1 computes all updates with the finest resolution; it means, the expected values that are involved in the update step are computed considering a telescopic sum until the finest grid,

i.e. level L (see Equation (16)). However, during the first iteration steps, we might be far away of the optimal solution, and, therefore, having a very accurate update may not be necessary. We can take advantage of the facts that MLMC considers different levels of discretisation with N_l independent samples at each level, and IS estimators are unbiased, by implementing IS independently at each grid level. We propose to use the iteration update (Equation (13)), first using IS for the coarsest levels only and for the finer levels later. We initially do IS for levels zero and one only, computing the control on the level one discretisation. Once we obtain a non-significant increase in the ESS, we set the time discretisation for the control update to level two, initialised with the current control solution. We can iterate this action at different levels: as soon as we obtain a non-significant increase in the ESS at level l, we reduce the time discretisation of the control update to level $l+1$, initialised with the level l control solution. Thus, in this case, the MLMC estimator at level $\mathcal{L}$ is given by

$$\hat{Y}_{IS,\mathcal{L}} = \frac{1}{N_0}\sum_{i=1}^{N_0} \hat{P}_0^{(i)} r_{u_1,0}^{(i)} + \sum_{l=1}^{\mathcal{L}} \frac{1}{N_l}\sum_{i=1}^{N_l} \left(\hat{P}_l^{(i)} r_{u_l,l}^{(i)} - \hat{P}_{l-1}^{(i)} r_{u_l,l-1}^{(i)}\right), \tag{17}$$

where $\mathcal{L} = 1, 2, \ldots, L$ and u_l corresponds to the last $\hat{u}$ that has been found, when doing iterations for level l. This implementation is described with more detail in the pseudo-code Algorithm 2.

Although we assume that the control is parametrised by K basic functions $h_k(t,x)$, $k = 1, \ldots, K$, i.e. $\hat{u} = \sum_{k=1}^{K} \theta_k h_k(t,x)$, for simplicity, we suppress the k index in the pseudo-codes being $\hat{u} = \theta h(t,x)$. Thus, the sub- and supra-index in θ indicate the update steps and the level where the update is performed, respectively. As stop criteria, we consider a maximum number of iterations $N_{\max}$, that in Algorithm 2 it may also depend on l; or a non-significant increase in the ESS.

When performing MLMC until level L, the number of samples at level l is related with N_L by Equation (15). These amount of samples are necessary to ensure that the square root of the estimated variance of the combined multilevel estimator has a superior bound similar to that for the bias, $\mathcal{O}(h_L)$. Since only levels zero and one are considered in the first step of Algorithm 2, i.e. the final discretisation is h_1 instead of h_L, we can use less samples. In particular, we use N_{update} samples for level one and $N_{update}M^{(\beta+1)/2}$ samples for level zero, with $N_{update} < N_1$. However, for the telescopic sums that are involved at levels $l > 1$, these amounts of samples are not enough; thus, once when we get the final expression for the control u at level one, we need to perform N_0 and N_1 samples at levels zero and one, respectively. The fact that the number of samples can be reduced at the early iteration steps is one of the main advantages of Algorithm 2 especially when $\beta > 1$.

In the next section, we analyse and compare the performances of MC, and Algorithms 1 and 2 when they are applied to finite horizon control problems based on Lorenz-96 model (Lorez, 1996). In general, Algorithm 1 reaches the maximum ESS value and Algorithm 2 performs faster than the others. In fact, if we combine Algorithms 1 and 2, we obtain the best performance in time and ESS.

Algorithm 2. MLMC and IS independently at each level

Require: Dynamic equation (i.e. Equation (1)), $S(t,x,u)$, $\hat{u}$ (i.e. θ_1^1 and $h(t,x)$), M, L, $N_{\max}^l$, N_{update}, ζ, N_L
  for $l = 1$ **to** L **do**
    while $n < N_{\max}^l$ **or** $\Delta ESS < \zeta$ **do**
      if $l = 1$ **then**
        Perform MLMC at levels zero and one with $N_{update}M^{(\beta+1)/2}$ and N_{update} samples, respectively
        Compute $\mathbb{E}[P_0 r_{\hat{u},0}]$, $\mathbb{V}[P_0 r_{\hat{u},0}]$, $\mathbb{E}[P_1 r_{\hat{u},1} - P_0 r_{\hat{u},0}]$ and $\mathbb{V}[P_1 r_{\hat{u},1} - P_0 r_{\hat{u},0}]$
      else
        Perform MLMC at level l, with $N_L M^{(\beta+1)(L-l)/2}$ samples
        Compute $\mathbb{E}[P_l r_{\hat{u},l} - P_{l-1} r_{\hat{u},l-1}]$ and $\mathbb{V}[P_l r_{\hat{u},l} - P_{l-1} r_{\hat{u},l-1}]$
      end if
      Compute $\mathbb{E}[P] = \mathbb{E}[P_0 r_{u_1,0}] + \sum_{l'=1}^{l-1} \mathbb{E}[P_{l'} r_{u_{l'},l'} - P_{l'-1} r_{u_{l'},l'-1}] + \mathbb{E}[P_l r_{\hat{u},l} - P_{l-1} r_{\hat{u},l-1}]$
      Compute $\mathbb{V}[P] = \mathbb{V}[P_0 r_{u_1,0}] + \sum_{l'=1}^{l-1} \mathbb{V}[P_{l'} r_{u_{l'},l'} - P_{l'-1} r_{u_{l'},l'-1}] + \mathbb{V}[P_l r_{\hat{u},l} - P_{l-1} r_{\hat{u},l-1}]$
      Update the control $\hat{u}$ at level l: $\theta_{n+1}^l \leftarrow \theta_n^l + \eta \mathbb{E}[P]$
    end while
    if $l = 1$ **then**
      $u_1 \leftarrow \hat{u}$
      Perform MLMC at levels zero and one with $N_0 = N_L M^{(\beta+1)L/2}$ and $N_1 = N_L M^{(\beta+1)(L-1)/2}$ samples, respectively
      Compute $\mathbb{E}[P_0 r_{u_1,0}]$, $\mathbb{V}[P_0 r_{u_1,0}]$, $\mathbb{E}[P_1 r_{u_1,1} - P_0 r_{u_1,0}]$ and $\mathbb{V}[P_1 r_{u_1,1} - P_0 r_{u_1,0}]$
      $\theta_1^{l+1} \leftarrow \theta_{n+1}^l$
    else
      $u_l \leftarrow \hat{u}$
      Perform MLMC at level l, with $N_l = N_L M^{(\beta+1)(L-l)/2}$ samples
      Compute $\mathbb{E}[P_l r_{u_l,l} - P_{l-1} r_{u_l,l-1}]$ and $\mathbb{V}[P_l r_{u_l,l} - P_{l-1} r_{u_l,l-1}]$
      $\theta_1^{l+1} \leftarrow \theta_{n+1}^l$
    end if
  end for

5. Numerical examples

In this section, we apply our algorithms to a set of finite horizon control problems. In particular, we illustrate the path integral cross entropy method for particle smoothing, considering a noisy observation and proceeding as Kappen and Ruiz did (Kappen & Ruiz, 2016; Ruiz & Kappen, 2017). Particle smoothing method is used for inference of stochastic processes, given noisy observations, i.e. it reconstructs latent time series from observations. In order to apply it here, we generate some data from the Lorenz-96 model (Lorez, 1996) and try to reconstruct the involved time series. We consider one observation only, but, of course, the procedure is valid for many observations as well.

Lorenz-96 model was originally introduced to describe the weather at mid-latitude circles considering κ discrete sectors and a continuous dynamics in time. This system may present chaotic behaviour and it has been analysed for different dimensions and external forces by Karimi and Paul (2010). In this work, we consider the dynamics of a periodic lattice of κ

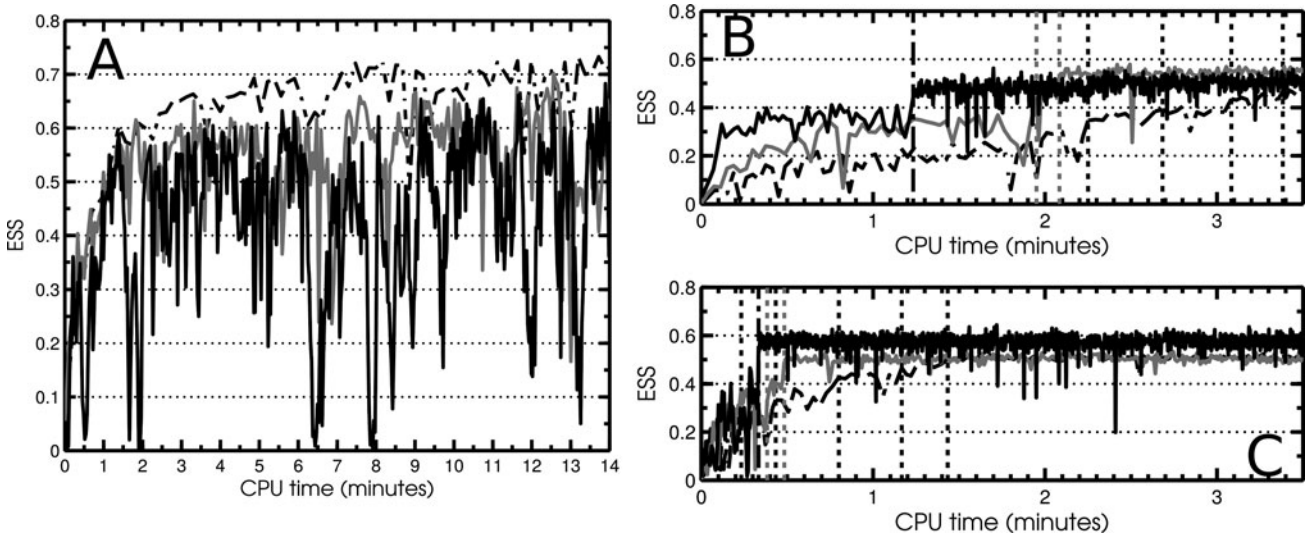

Figure 1. ESS vs. CPU time for Algorithm 1 (A), and Algorithm 2 (B and C), considering the dynamic equation (18) with $\kappa = 5$ and $F = 5$. For panel B, $N_{update} = N_1$, and for panel C $N_{update} = 1/h_1 ESS_0$. The algorithms were performed with: $M = 2$, $L = 6$, dashed black lines; $M = 4$, $L = 3$, solid grey lines; $M = 8$, $L = 2$, solid black lines. Vertical lines correspond to the end of the first loop in Algorithm 2, where the level is increased for (2,6), dotted black lines; (4,3), dotted grey lines; and (8,2), dashed-dotted black lines. All these realisations have the same initial and final discretisations, $h_0 = 0.1$ and $h_L = h_0/M^L$, respectively.

coupled variables, $X_k(t)$, with $k = 1, \ldots, \kappa$, where the dynamics of the kth variable is given by

$$dX_k(t) = ((X_{k+1} - X_{k-2})X_{k-1} - X_k + F + u_k(t, X(t)))\, dt + dW_k, \tag{18}$$

where F represents a constant external driving, $X_{-1} = X_\kappa$, $X_{-2} = X_{\kappa-1}$, and $X_{\kappa+1} = X_1$. In order to apply particle smoothing method, we define the cost function by

$$S(t_0, x, u) = \frac{1}{\lambda}\left(Q||X(T) - x_{obs}||^2 + \int_{t_0}^{T} \left(\frac{1}{2} u(s, X(s))' R u(s, X(s))\right) ds + \int_t^T u(s, X(s))' R dW(s)\right), \tag{19}$$

where now $u(t, X(t)) : [t_0, T] \times \mathbb{R}^\kappa \to \mathbb{R}^\kappa$, $dW(t)$ is an κ-dimensional Gaussian noise, and the end cost, $\Phi(x)$, is given by $Q||X(T) - x_{obs}||^2$, with Q constant and x_{obs} the observation.

In order to generate an observation, we integrate the uncontrolled and noiseless dynamics from $t_0 = 0$ until $T = 1$, with initial condition x_0. Our 'observation' is obtained by adding white Gaussian measurement noise, $\mathcal{N}(0, 0.5)$, to the final state. To ensure that all transients have decayed, the initial condition x_0 is defined as the final state after integrating forward for 1000 time units the uncontrolled and noiseless dynamics from random initial conditions. We have performed simulations for a huge variety of parameters, κ and F and for all of them $\alpha \approx 1$ and $\beta \approx 2$.

We start analysing the system with $F = 5$ and $K = 5$. For these parameters, the system does not have chaotic behaviour. We performed simulations with different pairs (M, L) in such a way all of them have the same final discretisation. In Figure 1(A), we show the ESS versus the CPU time when we used Algorithm 1 for (2,6 – dashed black line), (4,3 – solid grey line), (8,2 – solid black line) with $h_0 = 0.1$. We observe that the initial improvement of the ESS is similar for all schemes, but that adding more levels between the coarsest and finest discretisation significantly improves the asymptotic efficiency.

If instead we apply Algorithm 2, the simulations are performed faster, but we obtain, in general, less ESS. This is shown in Figure 1(B) for (2,6 – dashed black line), (4,3 – solid grey line), and (8,2 – solid black line), with $h_0 = 0.1$ and $N_{update} = N_1$. We observe that the initial increase of the ESS is significantly faster for Algorithm 2 than for Algorithm 1. However, its asymptotic performance is significantly worse. Vertical lines indicate the times at which the computation changes from level l to $l + 1$ (first loop of Algorithm 2, see pseudo-code).

As we expressed above, when only levels zero and one are considered, the discretisation error due to bias is $\mathcal{O}(h_1)$ instead of $\mathcal{O}(h_L)$. Thus, we can consider $N_{update} \propto 1/(h_1 ESS_0)$, where ESS_0 is an estimated initial ESS (we took it $\sim 5 \times 10^{-4}$). Figure 1(C) shows the simulations that were performed using Algorithm 2 for (2,6 – dashed black line), (4,3 – solid grey line), and (8,2 – solid black line), with $h_0 = 0.1$ and $N_{update} = 1/(h_1 ESS_0)$. We observe that initially there is a good ESS improvement and that asymptotic ESSs have similar values for Figure 1(B,C), supporting the fact that we can use the proposed N_{update} instead of N_1.

In order to implement our stopping criterion, $\Delta ESS < \zeta$, we defined a minimum number of iteration at each level; for iterations beyond this minimum, we computed the relative variation of the ESS in a window of 10 iterations, and we asked this to be less than 30%.

The Radon–Nikodym derivative that is involved in both algorithm is given by $\exp(-S(t, x, \hat{u}))/\psi(t, x)$, where $\psi(t, x) = \mathbb{E}[\exp(-S(t, x, \hat{u}))]$ has to be estimated when implementing the numerical computation. Since Algorithm 1 always considers L grid levels, the estimation of $\psi(t, x)$, and thus the update, are always better than those for Algorithm 2 that takes into account

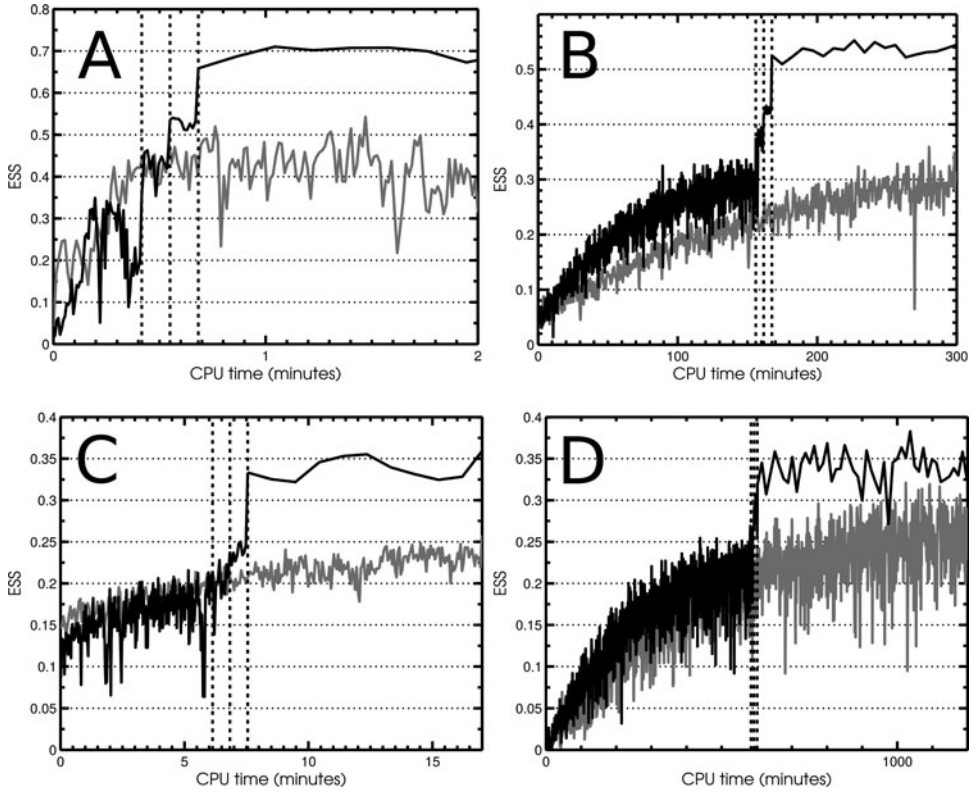

Figure 2. ESS vs. CPU time for a combined algorithm (black lines), and MC (grey lines), considering the dynamic equation (18) with: $\kappa = 5$, $F = 5$ (A); $\kappa = 16$, $F = 5$ (B); $\kappa = 5$, $F = 7$ (C); and $\kappa = 22$, $F = 5$ (D). All MLMC realisations have the same initial and final discretisations, $h_0 = 0.1$ and $h_L = h_0/M^L$, respectively. The time step for MC realisations is h_L.

$\mathcal{L}(\leq L)$ grid levels. Thus, since Algorithm 1 is more accurate and Algorithm 2 is faster, we propose a combined algorithm, implementing Algorithm 2 first, and then Algorithm 1. In this way, a good controller is obtained faster, and later the ESS is improved.

In Figure 2, we show the results of applying a combined algorithm (black) and standard MC (grey), for different F and κ values. In these particular cases, we have chosen to implement first Algorithm 2 with $M = 4$ and $L = 3$ and then Algorithm 1 with $M = 2$ and $L = 6$. Similar results are also obtained for other choices of M and L when performing Algorithm 2. Our combined algorithm is clearly much more effective and has less fluctuations than MC, even when the system presents chaotic behaviour. The ESS that is reached with the combined algorithm for $F = 5$ and $K = 5$, (Figure 2(A)) is the same than that when using only Algorithm 1 (Figure 1(A)); however, it takes much less CPU time with the combined algorithm. We have implemented all algorithms considering a control parametrised by the following functions: $u_k = a_k + b_k' Xt + c_k' Xt^2$, where $a_k \in \mathbb{R}$, $b_k,\ c_k \in \mathbb{R}^\kappa$. A more complex controller will lead to higher sampling efficiency, which also depends on the discretisation.

6. Conclusions

Theorem 2.1 gives an expression for the optimal control, $u^*(t, x)$, in terms of any sampling control, that can be estimated by MC sampling. Of course, the efficiency of the sampling strongly depends on the sampling control. This efficiency can be improved by using IS; but even then, the number of samples that are required may be large and initially it can take a long time to obtain a workable ESS.

The estimation of the optimal control, through Theorem 2.1, involves the computation of expected values. It has been shown that the MLMC method has less computational complexity than MC giving significant computational savings. IS can be applied in different ways, since MLMC considers different levels of discretisation with independent samples at each level, and IS estimators are unbiased. In particular, an improvement in the sampling control can be obtained even with a poor discretisation due to the variance reduction through MLMC. This can be improved by adding a level of discretisation and implementing IS independently at the new mesh level. This procedure has two advantages: (1) at the very initial steps, when nothing is known about a good sampling control and the ESS is too small, a very accurate improvement of the sampling control may not be necessary; thus, a rough approximation with a coarse discretisation is enough to determine the right direction to take in the gradient descent procedure (Equation (13)); (2) if $\beta < 1$, the lowest levels of MLMC are the cheapest in computational cost; and if $\beta \geq 1$, since the initial discretisation is poor, at the first steps of Algorithm 2, less samples can be done, reducing also the computational cost. For these reasons, Algorithm 2 is the fastest. Neither of these advantages are present in Algorithm 1, since the final discretisation is considered for all steps and it is not possible to reduce the number of samples. However, it estimates better the involved Radon–Nikodym derivatives and, therefore, it gives

a better asymptotic ESS. Thus, a combined algorithm that starts using the advantages of Algorithm 2 and then implement a few steps of Algorithm 1 is the best option to obtain a good ESS with less computational time.

Nevertheless, both Algorithms 1 and 2 give better results than MC with IS (see Figure 2). It is worth to note that the ESS also depends on the discretisation, and thus the comparison between MLMC and MC for levels $l < L$ is not fair. Indeed, if when performing Algorithm 2, the ESS were computed with the sampling control of level $\mathcal{L}(\leq L)$, but considering the discretisation of level L, this ESS will be always better than that for MC (results are not shown).

In summary, the implementation of IS independently at each mesh level in the MLMC algorithm gives a speed-up and performs much better than MC. Even more, a combination of both algorithms not only gives a better ESS than MC at the same computational time, it also reaches a larger asymptotic ESS.

Acknowledgments

S. A. Menchón would like to thank H-C Ruiz for helpful discussions. This work has been done with the support of Radboud Excellence Initiative of the Radboud University Nijmegen [grant number U557171-1], SeCyT-UNC and CONICET [grant number 11220150100644CO].

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Ra [grant number U557171-1]; SeCyT-UNC and CONICET [grant number 11220150100644CO].

ORCID

S. A. Menchón http://orcid.org/0000-0002-2142-2515

References

de Boer, P. T., Kroese, D. P., Mannor, S., & Rubinstein. R. Y. (2005). A tutorial on the Cross-Entropy Method. *Annals of Operations Research, 134*, 19–67.

Giles, M. B. (2008). Multi-level Monte Carlo path simulation. *Operations Research, 56*, 607–617.

Kappen, H. J. (2005). Linear theory for control of nonlinear stochastic systems. *Physical Review Letters, 95*, 200201.

Kappen, H. J., Gómez, V., & Opper, M. (2012). Optimal control as a graphical model inference problem. *Machine Learning, 87*, 159–182.

Kappen, H. J., & Ruiz, H. C. (2016). Adaptive importance sampling for control and inference. *Journal of Statistical Physics, 162*, 1244–1266.

Karimi, A., & Paul, M. R. (2010). Extensive chaos in the Lorenz-96 model. *Chaos, 20*, 043105.

Kong, A. (1992). *A Note on importance sampling using standardized weights* (Technical Report 348). Chicago: Department of Statistics, University of Chicago.

Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing, 6*, 113–119.

Lorenz, E. N. (1996). Predictability: A problem partly solved. In T. Palmer (Ed.), *Proceedings of the seminar on predictability* (Vol. I). ECWF Seminar (Reading: ECMWF).

Ruiz, H. C., & Kappen, H. J. (2017). Particle smoothing for hidden diffusion processes: Adaptive path integral smoother. *IEEE Transactions on Signal Processing, 65*, 3191–3203.

Stengel, R. (1993). *Optimal control and estimation*. New York, NY: Dover Publications.

Šukys, J. (2014). *Adaptive load balancing for massively parallel multi-level Monte Carlo solvers* (pp. 47–56). PPAM 2013 Part I, LNCS 8384. Berlin: Springer.

Thijssen, S. A., & Kappen, H. J. (2015). Path integral control and state-dependent feedback. *Physical Review E, 91*, 032104, (2015).