ESAIM: PROCEEDINGS AND SURVEYS, February 2019, Vol. 65, p. 114-144
B. Bouchard, J.-F. Chassagneux, F. Delarue, E. Gobet and J. Lelong, Editors

A CLASS OF FINITE-DIMENSIONAL NUMERICALLY SOLVABLE MCKEAN-VLASOV CONTROL PROBLEMS

Alessandro Balata[1], Côme Huré[2], Mathieu Laurière[3], Huyên Pham[4] and Isaque Pimentel[5]

Abstract. We address a class of McKean-Vlasov (MKV) control problems with common noise, called polynomial conditional MKV, and extending the known class of linear quadratic stochastic MKV control problems. We show how this polynomial class can be reduced by suitable Markov embedding to finite-dimensional stochastic control problems, and provide a discussion and comparison of three probabilistic numerical methods for solving the reduced control problem: quantization, regression by control randomization, and regress-later methods. Our numerical results are illustrated on various examples from portfolio selection and liquidation under drift uncertainty, and a model of interbank systemic risk with partial observation.

Keywords: McKean-Vlasov control, polynomial class, quantization, regress-later, control randomization.

1. Introduction

The optimal control of McKean-Vlasov (also called mean-field) dynamics is a rather new topic in the area of stochastic control and applied probability, which has been knowing a surge of interest with the emergence of the mean-field game theory. It is motivated on the one hand by the asymptotic formulation of cooperative equilibrium for a large population of particles (players) in mean-field interaction, and on the other hand from control problems with cost functional involving nonlinear functional of the law of the state process (e.g., the mean-variance portfolio selection problem or risk measure in finance).

In this paper, we are interested in McKean-Vlasov (MKV) control problems under partial observation and common noise, whose formulation is described as follows. On a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with two

[1] University of Leeds, Leeds, United Kingdom;
e-mail: A.Balata@leeds.ac.uk
[2] Univ. Paris Diderot - LPSM, Paris, France;
e-mail: hure@lpsm.paris
[3] Operations Research and Financial Engineering, Princeton University, Princeton, USA;
e-mail: lauriere@princeton.edu
[4] Univ. Paris Diderot - LPSM and FiME Lab, Paris, France;
e-mail: pham@math.univ-paris-diderot.fr
[5] EDF and FiME Lab, Palaiseau, France;
e-mail: isaque.santa-brigida-pimentel@edf.fr

Article published online by EDP Sciences and available at https://www.esaim-proc.org or https://doi.org/10.1051/proc/201965114

independent Brownian motions B and W^0, let us consider the controlled stochastic MKV dynamics in $\mathbb{R}^n$:

$$dX_s = b(X_s, \mathbb{P}^{W^0}_{X_s}, \alpha_s)ds + \sigma(X_s, \mathbb{P}^{W^0}_{X_s}, \alpha_s)dB_s + \sigma_0(X_s, \mathbb{P}^{W^0}_{X_s}, \alpha_s)dW^0_s, \quad X_0 = x_0 \in \mathbb{R}^n, \tag{1.1}$$

where $\mathbb{P}^{W^0}_{X_s}$ denotes the conditional distribution of X_s given W^0 (or equivalently given $\mathcal{F}^0_s$ where $\mathbb{F}^0 = (\mathcal{F}^0_t)_t$ is the natural filtration generated by W^0), and the control α is $\mathbb{F}^0$-progressive valued in some Polish space A. This measurability condition on the control means that the controller has a partial observation of the state, in the sense that she can only observe the common noise. We make the standard Lipschitz condition on the coefficients $b(x,\mu,a)$, $\sigma(x,\mu,a)$, $\sigma_0(x,\mu,a)$ with respect to (x,μ) in $\mathbb{R}^n \times \mathcal{P}_2(\mathbb{R}^n)$, uniformly in $a \in A$, where $\mathcal{P}_2(\mathbb{R}^n)$ is the set of all probability measures on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ with a finite second-order moment, endowed with the 2-Wasserstein metric $\mathcal{W}_2$. This ensures the well-posedness of the controlled MKV stochastic differential equation (SDE) (1.1). The cost functional over a finite horizon T associated to the stochastic MKV equation (1.1) (sometimes called conditional MKV equation) for a control process α, is

$$J(\alpha) \quad = \quad \mathbb{E}\Big[\int_0^T f(X_t, \mathbb{P}^{W^0}_{X_t}, \alpha_t)dt + g(X_T, \mathbb{P}^{W^0}_{X_T})\Big],$$

and the objective is to maximize over an admissible set $\mathcal{A}$ of control processes the cost functional:

$$V_0 \quad = \quad \sup_{\alpha \in \mathcal{A}} J(\alpha). \tag{1.2}$$

The set $\mathcal{A}$ of admissible controls usually requires some integrability conditions depending on the growth conditions on f, g, in order to ensure that $J(\alpha)$ is well-defined for $\alpha \in \mathcal{A}$ (more details will be given in the examples, see Section 4). Notice that classical partial observation control problem (without MKV dependence on the coefficients) arises as a particular case of (1.1)-(1.2). We refer to the introduction in [22] for the details.

Let us recall from [22] the dynamic programming equation associated to the conditional MKV control problem (1.2). We start by defining a suitable dynamic version of this problem. Let us consider $\mathcal{F}_0$ a sub σ-algebra of $\mathcal{F}$ independent of B, W^0. It is assumed w.l.o.g. that $\mathcal{F}_0$ is rich enough in the sense that $\mathcal{P}_2(\mathbb{R}^n) = \{\mathcal{L}(\xi) : \xi \in L^2(\mathcal{F}_0; \mathbb{R}^n)\}$, where $\mathcal{L}(\xi)$ denotes the law of ξ. Given a control $\alpha \in \mathcal{A}$, we consider the dynamic version of (1.1) starting from $\xi \in L^2(\mathcal{F}_0; \mathbb{R}^n)$ at time $t \in [0,T]$, and written as:

$$\begin{aligned} X^{t,\xi,\alpha}_s \quad = \quad & \xi + \int_t^s b(X^{t,\xi,\alpha}_u, \mathbb{P}^{W^0}_{X^{t,\xi,\alpha}_u}, \alpha_u)du + \int_t^s \sigma(X^{t,\xi,\alpha}_u, \mathbb{P}^{W^0}_{X^{t,\xi,\alpha}_u}, \alpha_u)dB_u \\ & + \int_t^s \sigma_0(X^{t,\xi,\alpha}_u, \mathbb{P}^{W^0}_{X^{t,\xi,\alpha}_u}, \alpha_u)dW^0_u\ ,\ t \leqslant s \leqslant T. \end{aligned}$$

Let us then define the dynamic cost functional:

$$J(t,\xi,\alpha) \quad = \quad \mathbb{E}\Big[\int_t^T f(X^{t,\xi,\alpha}_s, \mathbb{P}^{W^0}_{X^{t,\xi,\alpha}_s}, \alpha_s)ds + g(X^{t,\xi,\alpha}_T, \mathbb{P}^{W^0}_{X^{t,\xi,\alpha}_T})\Big],$$

for $(t,\xi) \in [0,T] \times L^2(\mathcal{F}_0; \mathbb{R}^n)$, $\alpha \in \mathcal{A}$, and notice by the law of conditional expectations, and as α is $\mathbb{F}^0$-progressive that

$$J(t,\xi,\alpha) \quad = \quad \mathbb{E}\Big[\int_t^T \hat{f}(\mathbb{P}^{W^0}_{X^{t,\xi,\alpha}_s}, \alpha_s)ds + \hat{g}(\mathbb{P}^{W^0}_{X^{t,\xi,\alpha}_s})\Big],$$

where $\hat{f} : \mathcal{P}_2(\mathbb{R}^n) \times A \to \mathbb{R}$, $\hat{g} : \mathcal{P}_2(\mathbb{R}^n) \to \mathbb{R}$ are defined by

$$\hat{f}(\mu, a) \ = \ \mu(f(\cdot, \mu, a)) \quad = \quad \int_{\mathbb{R}^n} f(x, \mu, a)\mu(dx), \tag{1.3}$$

$$\hat{g}(\mu) \;=\; \mu(g(\cdot,\mu)) \;=\; \int_{\mathbb{R}^n} g(x,\mu)\mu(dx). \tag{1.4}$$

Moreover, notice that the conditional law of $X_s^{t,\xi,\alpha}$ given W^0 depends on ξ only through its law $\mathcal{L}(\xi)$, and we can then define for $\alpha \in \mathcal{A}$:

$$\rho_s^{t,\mu,\alpha} \;=\; \mathbb{P}^{W^0}_{X_s^{t,\xi,\alpha}}, \quad \text{for } \; t \leqslant s, \; \mu = \mathcal{L}(\xi) \in \mathcal{P}_2(\mathbb{R}^n).$$

Therefore, the dynamic cost functional $J(t,\xi,\alpha)$ depends on $\xi \in L^2(\mathcal{F}_0;\mathbb{R}^n)$ only through its law $\mathcal{L}(\xi)$, and by an abuse of notation, we write $J(t,\mu,\alpha) = J(t,\xi,\alpha)$ when $\mu = \mathcal{L}(\xi)$. We then consider the value function for the conditional MKV control problem (1.2), defined on $[0,T]\times\mathcal{P}_2(\mathbb{R}^n)$ by

$$v(t,\mu) \;=\; \sup_{\alpha\in\mathcal{A}} J(t,\mu,\alpha) \;=\; \sup_{\alpha\in\mathcal{A}} \mathbb{E}\Big[\int_t^T \hat{f}(\rho_s^{t,\mu,\alpha},\alpha_s)ds + \hat{g}(\rho_T^{t,\mu,\alpha})\Big], \tag{1.5}$$

and notice that at time $t=0$, when $\xi = x_0$ is a constant, then $V_0 = v(0,\delta_{x_0})$.
It is shown in [22] that dynamic programming principle (DPP) for the conditional MKV control problem (1.5) holds: for $(t,\mu)\in[0,T]\times\mathcal{P}_2(\mathbb{R}^n)$,

$$v(t,\mu) \;=\; \sup_{\alpha\in\mathcal{A}} \mathbb{E}\Big[\int_t^\theta \hat{f}(\rho_s^{t,\mu,\alpha},\alpha_s)ds + v(\theta,\rho_\theta^{t,\mu,\alpha})\Big],$$

for any $\mathbb{F}^0$-stopping time θ valued in $[t,T]$. Next, by relying on the notion of differentiability with respect to probability measures introduced by P. L. Lions [17] (see also the lecture notes [7]) and the chain rule (Itô's formula) along flow of probability measures (see [6], [9]), we derive the HJB equation for v:

$$\begin{cases} \partial_t v + \sup\limits_{a\in A}\Big[\hat{f}(\mu,a) + \mu\big(\mathbb{L}^a v(t,\mu)\big) + \mu\otimes\mu\big(\mathbb{M}^a v(t,\mu)\big)\Big] \;=\; 0, \quad (t,\mu)\in[0,T)\times\mathcal{P}_2(\mathbb{R}^n), \\ \qquad\qquad v(T,\mu) \;=\; \hat{g}(\mu), \quad \mu\in\mathcal{P}_2(\mathbb{R}^n), \end{cases} \tag{1.6}$$

where for $\phi\in\mathcal{C}_b^2(\mathcal{P}_2(\mathbb{R}^n))$, $a\in A$, and $\mu\in\mathcal{P}_2(\mathbb{R}^n)$, $\mathbb{L}^a\phi(\mu)$ is the function $\mathbb{R}^n\to\mathbb{R}$ defined by

$$\mathbb{L}^a\phi(\mu)(x) \;=\; \partial_\mu\phi(\mu)(x).b(x,\mu,a) + \frac{1}{2}\mathrm{tr}\big(\partial_x\partial_\mu\phi(\mu)(x)(\sigma\sigma^\intercal + \sigma_0\sigma_0^\intercal)(x,\mu,a)\big), \tag{1.7}$$

and $\mathbb{M}^a\phi(\mu)$ is the function $\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ defined by

$$\mathbb{M}^a\phi(\mu)(x,x') \;=\; \frac{1}{2}\mathrm{tr}\big(\partial_\mu^2\phi(\mu)(x,x')\sigma_0(x,\mu,a)\sigma_0^\intercal(x',\mu,a)\big). \tag{1.8}$$

The HJB equation (1.6) is a fully nonlinear partial differential equation (PDE) in the infinite-dimensional Wasserstein space. In general, this PDE does not have an explicit solution except in the notable important class of linear-quadratic MKV control problem. Numerical resolution for MKV control problem or equivalently for the associated HJB equation is a challenging problem due to the nonlinearity of the optimization problem and the infinite-dimensional feature of the Wasserstein space. In this work, our purpose is to investigate a class of MKV control problems which can be reduced to finite-dimensional problems in view of numerical resolution.

2. Polynomial McKean-Vlasov control problem

2.1. Main assumptions

We make two kinds of assumptions on the coefficients of the model: one on the dependence on x and the other on the dependence on μ.

Assumptions: dependence on x: we consider a class of models where the coefficients of the MKV equation are linear w.r.t. the state variable X, i.e., they are in the form

$$\left\{\begin{array}{rcl} b(x,\mu,a) &=& b_0(\mu,a)+b_1(\mu,a)x, \\ \sigma(x,\mu,a) &=& \vartheta_0(\mu,a)+\vartheta_1(\mu,a)x, \\ \sigma_0(x,\mu,a) &=& \gamma_0(\mu,a)+\gamma_1(\mu,a)x, \end{array}\right. \tag{2.1}$$

while the running and terminal cost functions are polynomial in the state variable in the sense that they are in the form (for simplicity we present here the one-dimensional case $n=1$)

$$\begin{array}{rcl} f(x,\mu,a) &=& f_0(\mu,a)+\sum_{k=1}^{p} f_k(\mu,a)x^k, \\ g(x,\mu) &=& g_0(\mu)+\sum_{k=1}^{p} g_k(\mu)x^k, \end{array}$$

for some integer $p \geqslant 1$.

Assumptions: dependence on μ: we assume that all the coefficients depend on μ through its first p moments, i.e., they are in the form

$$\left\{\begin{array}{rclrcl} b_0(\mu,a) &=& \bar{b}_0(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a), & b_1(\mu,a) &=& \bar{b}_1(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a) \\ \vartheta_0(\mu,a) &=& \bar{\vartheta}_0(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a), & \vartheta_1(\mu,a) &=& \bar{\vartheta}_1(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a) \\ \gamma_0(\mu,a) &=& \bar{\gamma}_0(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a), & \gamma_1(\mu,a) &=& \bar{\gamma}_1(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a) \\ f_k(\mu,a) &=& \bar{f}_k(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a), & g_k(\mu) &=& \bar{g}_k(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p), \quad k=0,\ldots,p, \end{array}\right. \tag{2.2}$$

where, given $\mu \in \mathcal{P}_p(\mathbb{R})$, we denote by

$$\bar{\mu}_k \;=\; \int x^k \mu(dx), \quad k=1,\ldots,p.$$

We assume that the coefficients $\bar{b}_0, \bar{b}_1, \bar{\vartheta}_0, \bar{\vartheta}_1, \bar{\gamma}_0, \bar{\gamma}_1$ are Lipschitz w.r.t. the p first arguments uniformly w.r.t. the control argument $a \in A$. This condition will ensure existence and uniqueness of a solution to the finite-dimensional MKV SDE defined later in (2.3).

Notice that in this case, the functions $\hat{f}$ and $\hat{g}$ defined in (1.3)-(1.4) are given by

$$\begin{array}{rcl} \hat{f}(\mu,a) &=& \bar{f}_0(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a)+\sum_{k=1}^{p}\bar{f}_k(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a)\bar{\mu}_k \\ &=:& \bar{f}(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p,a), \\ \hat{g}(\mu) &=& \bar{g}_0(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p)+\sum_{k=1}^{p}\bar{g}_k(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p)\bar{\mu}_k \\ &=:& \bar{g}(\bar{\mu}_1,\bar{\mu}_2,\ldots,\bar{\mu}_p). \end{array}$$

Remark 2.1. *In the multidimensional case, we should consider a class of multi-polynomial functions f and g of degree p in the form*

$$f(x,\mu,a) \;=\; \sum_{|\mathbf{k}|=0}^{p} f_{\mathbf{k}}\left((\mu^{\mathbf{k}'})_{|\mathbf{k}'|\leqslant p},a\right)x^{\mathbf{k}}, \qquad g(x,\mu) \;=\; \sum_{|\mathbf{k}|=0}^{p} g_{\mathbf{k}}\left((\mu^{\mathbf{k}'})_{|\mathbf{k}'|\leqslant p}\right)x^{\mathbf{k}},$$

where we use multi-index notations $\mathbf{k} = (k_1, \ldots, k_n) \in \mathbb{N}^n$, $|\mathbf{k}| = k_1 + \ldots + k_n$, $x^{\mathbf{k}} = x_1^{k_1} \ldots x_n^{k_n}$ *for* $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ *and*

$$\mu^{\mathbf{k}} \quad = \quad \int_{\mathbb{R}^n} x^{\mathbf{k}} \mu(dx).$$

2.2. **Markovian embedding**

Given the controlled process $X = X^\alpha$ solution to the stochastic MKV dynamics (1.1), denote by

$$Y_t^{(k)} \quad = \quad \mathbb{E}[X_t^k | W^0], \quad k = 1, \ldots, p.$$

To alleviate the notations, we assume that $n = 1$ (otherwise multi-indices should be used). From the linear/polynomial assumptions (2.1)-(2.2), by Itô's formula and taking conditional expectations, we can derive the dynamics of $(Y^{(1)}, Y^{(2)}, \ldots, Y^{(p)})$ as

$$dY_t^{(k)} \; = \; B_k(Y_t^{(1)}, Y_t^{(2)}, \ldots, Y_t^{(p)}, \alpha_t)dt + \Sigma_k(Y_t^{(1)}, Y_t^{(2)}, \ldots, Y_t^{(p)}, \alpha_t)dW_t^0, \quad Y_0^{(k)} = x_0^k, \quad k = 1, \ldots, p, \quad (2.3)$$

where, by convention $y_0 = 1$, $y_{-1} = 0$,

$$\begin{aligned}
B_k(y_1, y_2, \ldots, y_p, a) \quad &= \quad k\bar{b}_0(y_1, \ldots, y_p, a)y_{k-1} + k\bar{b}_1(y_1, \ldots, y_p, a)y_k \\
&\quad + \frac{k(k-1)}{2}(\bar{\vartheta}_0(y_1, \ldots, y_p, a))^2 y_{k-2} + \frac{k(k-1)}{2}(\bar{\vartheta}_1(y_1, \ldots, y_p, a))^2 y_k \\
&\quad + k(k-1)\bar{\vartheta}_0(y_1, \ldots, y_p, a)\bar{\vartheta}_1(y_1, \ldots, y_p, a)y_{k-1} \\
&\quad + \frac{k(k-1)}{2}(\bar{\gamma}_0(y_1, \ldots, y_p, a))^2 y_{k-2} + \frac{k(k-1)}{2}(\bar{\gamma}_1(y_1, \ldots, y_p, a))^2 y_k \\
&\quad + k(k-1)\bar{\gamma}_0(y_1, \ldots, y_p, a)\bar{\gamma}_1(y_1, \ldots, y_p, a)y_{k-1}, \quad k = 1, \ldots, p, \\
\Sigma_k(y_1, y_2, \ldots, y_p, a) \quad &= \quad k\left(\bar{\gamma}_0(y_1, \ldots, y_p, a)y_{k-1} + \bar{\gamma}_1(y_1, \ldots, y_p, a)y_k\right), \quad k = 1, \ldots, p,
\end{aligned}$$

while the cost functional is written as

$$J(\alpha) \quad = \quad \mathbb{E}\Big[\int_0^T \bar{f}(Y_t^{(1)}, Y_t^{(2)}, \ldots, Y_t^{(p)}, \alpha_t)dt + \bar{g}(Y_T^{(1)}, Y_T^{(2)}, \ldots, Y_T^{(p)})\Big]. \quad (2.4)$$

The MKV control problem is then reduced in this polynomial framework into a finite-dimensional control problem with $\mathbb{F}^0$-adapted controlled variables $(Y^{(1)}, Y^{(2)}, \ldots, Y^{(p)})$. In the next section, we describe three probabilistic numerical methods for solving finite-dimensional stochastic control problems and will apply in section 4 each of these methods to three examples arising from polynomial MKV control problems under partial observation and common noise.

3. Numerical methods

In this section, we introduce our numerical methods for the resolution of the reduced problem (2.3)-(2.4).
Let us introduce the process Z^α, valued in $\mathbb{R}^d$, controlled by an adapted process α taking values in A, solution to

$$dZ_t^\alpha = b(Z_t^\alpha, \alpha_t)dt + \sigma_0(Z_t^\alpha, \alpha_t)dW_t^0, \quad Z_0^\alpha = z_0 \in \mathbb{R}^d, \quad (3.1)$$

and the performance measure

$$J(t, z, \alpha) = \mathbb{E}\left[\int_t^T f(Z_t^\alpha, \alpha_t)dt + g(Z_T^\alpha)\Big| Z_t^\alpha = z\right], \quad (3.2)$$

which assesses the average performance of the control.

Introduce now a time discretization $t_n = n\Delta t$, $n = 0, \dots, N$, $\Delta t = T/N$, and denote by $\mathcal{A}_{\Delta t}$ the space of discrete processes $(\alpha_{t_n})_{n=0}^{N-1}$ such that for all n, $n = 0, \dots, N-1$, α_{t_n} is $\mathcal{F}_{t_n}^0$-measurable.

We can write the Euler approximation of the SDE governing the process $Z = Z^\alpha$, with $\alpha \in \mathcal{A}_{\Delta t}$ (to alleviate notations, we sometimes omit the dependence on α when there is no ambiguity, and keep the same notation Z for the discrete and continuous process)

$$Z_{t_{n+1}} = Z_{t_n} + b(Z_{t_n}, \alpha_{t_n})\Delta t + \sigma_0(Z_{t_n}, \alpha_{t_n})\Delta W_{t_n}^0, \tag{3.3}$$

where $\Delta W_{t_n}^0 \sim \mathcal{N}(0, \Delta t)$ is an increment of W^0.

The discrete time approximation of $J(t_n, z, \alpha)$ is defined as:

$$J_{\Delta t}(t_n, z, \alpha) = \mathbb{E}\left[\sum_{k=n}^{N-1} f(Z_{t_k}, \alpha_{t_k})\Delta t + g(Z_{t_N})\Big| Z_{t_n} = z\right], \tag{3.4}$$

where $\alpha \in \mathcal{A}_{\Delta t}$.

3.1. **Value and Performance iteration**

For $n = 0, \dots, N$, consider $V_{\Delta t}(t_n, z) = \sup_{\alpha \in \mathcal{A}_{\Delta t}} J_{\Delta t}(t_n, z, \alpha)$, the discrete time approximation of the value function at time t_n: $V\big(t_n, z\big) = \sup_{\alpha \in \mathcal{A}} J(t_n, z, \alpha)$. The dynamic programming principle states that $\big(V_{\Delta t}(t_n, \cdot)\big)_{0 \leqslant n \leqslant N}$ is solution to the Bellman equation:

$$\begin{cases} V_{\Delta t}(t_N, z) = g(z) \\ V_{\Delta t}(t_n, z) = \sup_{a \in A} \Big\{ f(z, a)\Delta t + \mathbb{E}_{n,z}^a\big[V_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\big]\Big\}, \quad n = N-1, \dots, 0, \end{cases} \tag{3.5}$$

where $\mathbb{E}_{n,z}^a[\cdot]$ denotes the expectation conditioned on the event $\{Z_{t_n} = z\}$ and when using the control $\alpha_{t_n} = a$ at time t_n. Observe that for $n = 0, \dots, N-1$, the equation (3.5) provides a backward procedure to recursively compute the $V_{\Delta t}(t_n, \cdot)$ if we know how to analytically compute the conditional expectations $\mathbb{E}_{n,z}^a[V_{\Delta t}(t_{n+1}, Z_{t_{n+1}})]$ for all $z \in \mathbb{R}^d$ and all control $a \in A$. We refer to the procedure in (3.5) as value iteration.

An alternative approach to compute $V_{\Delta t}(t_n, \cdot)$, for $n = 0, \dots, N-1$, is to notice that once again by the dynamic programming principle, it holds that $\big(V_{\Delta t}(t_n, \cdot)\big)_{0 \leqslant n \leqslant N}$ is solution to the backward equation

$$\begin{cases} V_{\Delta t}(t_N, z) = g(z) \\ V_{\Delta t}(t_n, z) = \sup_{a \in A} \left\{ f(z, a)\Delta t + \mathbb{E}_{n,z}^a\left[\sum_{k=n+1}^{N-1} f\big(Z_{t_k}, \alpha_{t_k}^*(Z_{t_k})\big)\Delta t + g(Z_{t_N})\right]\right\}, \quad n = N-1, \dots, 0, \end{cases} \tag{3.6}$$

where for $k = n+1, \dots, N-1$, the control $\alpha_{t_k}^*$ is the optimal control at time t_k defined as follows:

$$\alpha_{t_k}^*(z) = \operatorname*{argmax}_{a \in A} \left\{ f(z, a)\Delta t + \mathbb{E}_{k,z}^a\left[\sum_{\ell=k+1}^{N-1} f\big(Z_{t_\ell}^*, \alpha_{t_\ell}^*(Z_{t_\ell}^*)\big)\Delta t + g(Z_{t_N}^*)\right]\right\}, \tag{3.7}$$

and where $\big(Z_{t_k}^*\big)_{n \leqslant k \leqslant N}$ is the process Z controlled by the following control α from time t_n to t_N:

$$\begin{cases} \alpha_{t_n} = a, \\ \alpha_{t_k} = \alpha_{t_k}^* \text{ for } n+1 \leqslant k \leqslant N-1. \end{cases} \tag{3.8}$$

For $n = 0, \dots, N-1$, the scheme (3.6) provides once again a backward procedure to compute $V_{\Delta t}(t_n, \cdot)$, assuming that we know how to analytically compute the conditional expectations

$$\mathbb{E}^a_{n,z}\left[\sum_{k=n+1}^{N-1} f\big(Z_{t_k}, \alpha^*_{t_k}(Z_{t_k})\big)\Delta t + g(Z_{t_N})\right],$$

for all $z \in \mathbb{R}^d$ and all control $a \in A$. We refer to the procedure in (3.6) as the performance iteration[1].

Except for trivial cases, closed-form formulas for the conditional expectations appearing in the value and policy iteration procedures are not available, and they have to be approximated, which is the main difficulty when implementing both approaches to compute the value functions. In the next section, we discuss different ways to approximate conditional expectations and derive the corresponding estimations of the value functions $V_{\Delta t}(t_n, \cdot)$ for $n = 0, \dots, N-1$.

3.2. **Approximation of conditional expectations**

In this subsection, we present three numerical methods that we apply later to conditional MKV problems. Two of these methods belong to the class of Regression Monte Carlo techniques, a family of algorithms whose effectiveness highly relies on the choice of the basis functions used to approximate conditional expectations; the third algorithm, Quantization, approximate the controlled process $Z^\alpha_{t_n}$ with a particular finite state Markov chain for which expectations can be approximated quickly.

3.2.1. *Regression Monte Carlo*

In the simpler uncontrolled case, the family of Regression Monte Carlo algorithms is based on the idea of approximating the conditional expectation $\mathbb{E}\big[V_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\big|Z_{t_n}\big]$, for $n = 0, \dots, N-1$, by the orthogonal projection of $V_{\Delta t}(t_{n+1}, Z_{t_{n+1}})$ onto the space generated by a finite family of $\big\{\phi_k(Z_{t_n})\big\}_{k\geqslant 1}$ where $(\phi_k)_{k\geqslant 1}$ is a family of *basis functions*, i.e., a family of measurable real-valued functions defined on $\mathbb{R}^d$ such that $\big(\phi_k(Z_{t_n})\big)_{k\geqslant 1}$ is total in $L^2(\sigma(Z_{t_n}))$[2] and such that for all scalars β_k and all $K \geqslant 1$, if $\sum_{k=1}^K \beta_k\phi_k(Z_{t_n}) = 0$ a.s. then $\beta_k = 0$, for $k = 1, \dots, K$.

The expectation $\mathbb{E}\big[V_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\big|Z_{t_n}\big]$ should then be approximated as follows:

$$\mathbb{E}\big[V_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\big|Z_{t_n}\big] \approx \sum_{k=1}^K \beta^n_k\phi_k(Z_{t_n}), \tag{3.9}$$

where $K \geqslant 1$ is fixed and $\beta^n = (\beta^n_1, \dots, \beta^n_K)^\top$ is defined as:

$$\beta^n = \operatorname*{argmin}_{\beta\in\mathbb{R}^K}\left\{\mathbb{E}\left[\left|V_{\Delta t}(t_{n+1}, Z_{t_{n+1}}) - \sum_{k=1}^K \beta_k\phi_k(Z_{t_n})\right|^2\right]\right\}. \tag{3.10}$$

Notice that β^n is defined in (3.10) as the minimizer of a quadratic function, and can then be rewritten by straightforward calculations as:

$$\beta^n = \mathbb{E}\left[\phi(Z_{t_n})\phi(Z_{t_n})^\top\right]^{-1}\mathbb{E}\left[V_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\phi(Z_{t_n})\right], \tag{3.11}$$

[1] This procedure is also referred to as the "policy iteration" in the literature.

[2] $L^2(\sigma(Z_{t_n}))$ is the space of the square-integrable $\sigma(Z_{t_n})$-measurable r.v.

where we use the notation $\phi = (\phi_1, \ldots, \phi_K)^\top$, and where we assumed that $\mathbb{E}\left[\phi(Z_{t_n})\phi(Z_{t_n})^\top\right]$ is invertible[3]. In order to estimate a solution to (3.11) we rely on Monte Carlo simulations to approximate expectations with finite sums. Consider the training set $\{(Z^m_{t_n}, Z^m_{t_{n+1}})\}_{m=1}^M$ at time t_n obtained by running $M \geqslant 1$ forward simulations of the process Z from time $t_0 = 0$ to t_{n+1}. β^n defined in (3.11) can then be estimated by

$$\hat{\beta}^n = \left(\hat{\mathcal{A}}^M_n\right)^{-1} \frac{1}{M} \sum_{m=1}^M V_{\Delta t}(t_{n+1}, Z^m_{t_{n+1}})\phi(Z^m_{t_n}), \tag{3.12}$$

where we denote by $\hat{\mathcal{A}}^M_n$ the estimator $\frac{1}{M}\sum_{m=1}^M \phi(Z^m_{t_n})\phi(Z^m_{t_n})^\top$ of the covariance matrix

$$\mathcal{A}_n = \mathbb{E}\left[\phi(Z_{t_n})\phi(Z_{t_n})^\top\right].$$

The procedure presented above offers a convenient mean to approximate conditional expectations when the dynamics of the process Z are uncontrolled. When controlled, however, one has to account for the effect of the control on the conditional expectations either explicitly, via Control Randomization, or implicitly, via Regress-Later.

Control Randomization

In order to explicitly account for the effect of the control, one could directly introduce dependence on the control in the basis function. This basic idea of Control Randomization consists in replacing in the dynamics of Z the endogenous control by an exogenous control $(I_{t_n})_{0\leqslant n\leqslant N}$, as introduced in [15]. Trajectories of $(Z_{t_n}, I_{t_n})_{0\leqslant n\leqslant N}$ can then be simulated from time t_0 to time t_N. Consider the training set $\{Z^m_{t_n}, I^m_{t_n}\}^{N,M}_{n=0,m=1}$, with $M \geqslant 1$, where $I^m_{t_n}$ are i.i.d. samples from a "training distribution" μ_n with support in A. The training set will be used to estimate the optimal β^n coefficients for $n = 0, \ldots, N-1$. In the case of value iteration, $\{V_{\Delta t}(t_{n+1}, Z^m_{t_{n+1}})\}_{m=1}^M$ is regressed against basis functions (which are, in this context, functions of the state and the control) evaluated at the points $\{Z^m_{t_n}, I^m_{t_n}\}_{m=1}^M$, as follows:

$$\mathbb{E}^a_{n,z}\left[V_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\right] \approx \sum_{k=1}^K \hat{\beta}^n_k \phi_k(z, a),$$

where $\hat{\beta}^n$ is an estimator of

$$\beta^n := \underset{\beta\in\mathbb{R}^K}{\operatorname{argmin}} \left\{ \mathbb{E}\left[\left(V_{\Delta t}\left(t_{n+1}, Z_{t_{n+1}}\right) - \sum_{k=1}^K \beta_k \phi_k\left(Z_{t_n}, I_{t_n}\right) \right)^2 \right] \right\},$$

defined as

$$\hat{\beta}^n = \left(\hat{\mathcal{A}}^M_n\right)^{-1} \frac{1}{M} \sum_{m=1}^M \left[V_{\Delta t}\left(t_{n+1}, Z^m_{t_{n+1}}\right)\phi(Z^m_{t_n}, I^m_{t_n}) \right], \tag{3.13}$$

where $\phi = (\phi_1, \ldots, \phi_K)^\top$ and

$$\hat{\mathcal{A}}^M_n = \frac{1}{M} \sum_{m=1}^M \phi(Z^m_{t_n}, I^m_{t_n})\phi(Z^m_{t_n}, I^m_{t_n})^\top \tag{3.14}$$

estimates the covariance matrix $\mathcal{A}_n = \mathbb{E}\left[\phi(Z_{t_n}, I_{t_n})\phi(Z_{t_n}, I_{t_n})^\top\right]$.

[3] If the assumption does not hold, the least squares problem can still be solved via SVD approach, which is consistent with regression techniques. See e.g. chapter 8 in [13].

Notice that the basis functions take state and action variables as input in the case of Control Randomization-based method, i.e., their domain is $\mathbb{R}^d \times A$. Also, observe that the estimated conditional expectation highly depends on the choice of the randomization for the control[4].

An optimal feedback control at time t_n given $Z_{t_n} = z$ is approximated by the expression (see Subsection 3.4 for more practical details on the computation of the argmax):

$$\hat{\alpha}_{t_n}(z) = \underset{a \in A}{\operatorname{argmax}} \left\{ f(z,a)\Delta t + \sum_{k=1}^{K} \hat{\beta}_k^n \phi_k(z,a) \right\}. \tag{3.15}$$

The value function at time t_n is then estimated using Control Randomization method and value iteration procedure as

$$\widehat{V}_{\Delta t}^{\mathrm{CR}}(t_n, z) = f\big(z, \hat{\alpha}_{t_n}(z)\big)\Delta t + \sum_{k=1}^{K} \hat{\beta}_k^n \phi_k\big(z, \hat{\alpha}_{t_n}(z)\big), \quad z \in \mathbb{R}^d.$$

Notice that Control Randomization can be easily employed in a performance iteration procedure by computing controls (3.15), keeping in mind that at each time t_n we need to re-simulate new trajectories $\{\tilde{Z}_{t_k}^m\}_{k=n,m=1}^{N,M}$ iteratively from the initial condition $\tilde{Z}_{t_n}^m = z$, using the estimated optimal strategies $(\hat{\alpha}_{t_k})_{k=n+1}^{N-1}$ to compute the quantities $\sum_{k=n}^{N-1} f\big(t_k, \tilde{Z}_{t_k}^m, \hat{\alpha}_{t_k}(\tilde{Z}_{t_k}^m)\big) + g(\tilde{Z}_{t_N}^m)$, for $1 \leqslant m \leqslant M$.

Regress-Later

We present now a regress-later idea in which conditional expectation with respect to Z_{t_n} is computed in two stages. First, a conditional expectation with respect to $Z_{t_{n+1}}$ is approximated in a regression step by a linear combination of basis functions of $Z_{t_{n+1}}$. Then, analytical formulas are applied to condition this linear combination of functions of future values on the present value Z_{t_n}. For further details, see [12], [4], [19] or [2]. With this approach, the effect of the control is factored in implicitly, through its effect on the (conditional) distribution of $Z_{t_{n+1}}$ conditioned on Z_{t_n}.

Unlike the traditional Regress-Now method for approximating conditional expectations (which we discussed so far in the uncontrolled case and in Control Randomization), the Regress-Later approach, as studied in [2], imposes conditions on basis functions:

Assumption 3.1. *For each basis function ϕ_k, $k = 1, \dots, K$, the conditional expectation*

$$\hat{\phi}_k^n(z,a) = \mathbb{E}_{n,z}^a[\phi_k(Z_{t_{n+1}})]$$

can be computed analytically.

Using the Regress-Later approximation of the conditional expectation and recalling Assumption 3.1 we obtain the optimal control $\alpha_{t_n}^m$ corresponding to the point $Z_{t_n}^m$, sampled independently from a "training distribution" μ_n (see Subsection 3.3 for further details):

$$\alpha_{t_n}^m = \underset{a \in A}{\operatorname{argmax}} \Big\{ f(Z_{t_n}^m, a)\Delta t + \sum_{k=1}^{K} \hat{\beta}_k^{n+1} \hat{\phi}_k^n\big(Z_{t_n}^m, a\big) \Big\}.$$

Notice that we are able to exploit the linearity of conditional expectations because

$$\hat{\beta}^{n+1} = \underset{\beta \in \mathbb{R}^K}{\operatorname{argmin}} \left\{ \sum_{m=1}^{M} \Big[V_{\Delta t}(t_{n+1}, Z_{t_{n+1}}^m) - \sum_{k=1}^{K} \beta_k \phi_k(Z_{t_{n+1}}^m) \Big]^2 \right\} \tag{3.16}$$

[4]Basically, different randomized controls may drive the process Z to very different locations, and the estimations will suffer from inaccuracy on the states that have been rarely visited.

is a constant once the training sets at times t_k, $k = n+1, \dots, N$, are fixed.

The value function at time t_n, is then estimated using Regress-Later method and value iteration procedure as

$$\widehat{V}^{\mathrm{RL}}_{\Delta t}(t_n, Z^m_{t_n}) = f(Z^m_{t_n}, \alpha^m_{t_n})\Delta t + \sum_{k=1}^{K} \hat{\beta}^{n+1}_k \hat{\phi}^n_k\big(Z^m_{t_n}, \alpha^m_{t_n}\big).$$

Notice that contrary to Control Randomization, Regress-Later does not require the training points to be distributed as $Z_{t_{n+1}}$ conditioned on Z_{t_n} because the projection (3.16) is an approximation to an expectation conditional to the measure μ_n which can be chosen freely to optimize the precision of the sample estimation. On the other hand Regress-Later, similarly to Control Randomization, can be easily employed in a performance iteration procedure by generating forward trajectories at each time step.

Remark 3.1. *Recall that the Regress-Later method uses training points that are i.i.d at each time step and independent across time steps. Contrary to other Regression Monte Carlo approaches, Regress-Later does not require to use the information about the conditional distribution during the regression step as that is accounted for in the second step of the method, when conditional expectations are computed analytically.*

3.2.2. *Quantization*

We propose in this section a quantization-based algorithm to numerically solve control problems. We may also refer to the latter as the Q-algorithm or Q in all the numerical examples considered in Section 4, where Q stands for Quantization. Let us first introduce some ingredients of Quantization, and then propose different ways of using them to approximate conditional expectations.

Let $(E, |.|)$ be a Euclidean space. We denote by $\hat{\varepsilon}$ a L-quantizer of an E-valued random variable ε, that is a discrete random variable on a grid $\Gamma = \{e_1, \dots, e_L\} \subset E^L$ defined by

$$\hat{\varepsilon} \;=\; \mathrm{Proj}_\Gamma(\varepsilon) \;=\; \sum_{\ell=1}^{L} e_\ell 1_{\varepsilon \in C_\ell(\Gamma)},$$

where $C_1(\Gamma), \dots, C_L(\Gamma)$ are the Voronoi cells corresponding to Γ, i.e., they form a Borel partition of E satisfying

$$C_\ell(\Gamma) \;\subset\; \Big\{e \in E : |e - e_\ell| \;=\; \min_{j=1,\dots,L} |e - e_j|\Big\},$$

and where Proj_Γ stands for the Euclidean projection on Γ.

The discrete law of $\hat{\varepsilon}$ is then characterized by

$$p_\ell \;=\; \mathbb{P}[\hat{\varepsilon} = e_\ell] \;=\; \mathbb{P}[\varepsilon \in C_\ell(\Gamma)], \quad \ell = 1, \dots, L.$$

The grid of points $(e_\ell)_{\ell=1}^L$ which minimizes the L^2-quantization error $\|\varepsilon - \hat{\varepsilon}\|_2$ leads to the so-called optimal L-quantizer, and can be obtained by a stochastic gradient descent method, known as Kohonen algorithm or competitive learning vector quantization (CLVQ) algorithm, which also provides as a byproduct an estimation of the discrete law $(p_\ell)_{\ell=1}^L$. We refer to [20] for a description of the algorithm, and mention that for the normal distribution, the optimal grids and the weights of the Voronoi tesselations are precomputed for dimension up to 10 and are available on the website http://www.quantize.maths-fi.com.

In practice, optimal grids of the Gaussian random variable $\mathcal{N}_1(0,1)$ of dimension 1 with 25 to 50 points, have been used to solve the control problems considered in Section 4.

We now propose two ways to approximate conditional expectations. The first approximation belongs to the family of the constant piecewise approximation, and the other one is an improvement on the first one, where the continuity of the approximation w.r.t. the control variable is preserved.

In the sequel, assume that for $n = 0, \dots, N-1$ we have a set Γ_n of points in $\mathbb{R}^d$ that should be thought of as a training set used for estimating $V(t_n, \cdot)$. See Subsection 3.3 for more details on how to build Γ_n.

Piecewise constant interpolation

We assume here that we already have an estimate of $V_{\Delta t}(t_{n+1}, \cdot)$, the value function at time t_{n+1}, for $n \in \{0, \dots, N-1\}$, and we denote by $\widehat{V}^Q_{\Delta t}(t_{n+1}, \cdot)$ the estimate.
The conditional expectation is then approximated as

$$\mathbb{E}^a_{n,z}\big[\widehat{V}^Q_{\Delta t}\big(t_{n+1}, Z_{t_{n+1}}\big)\big] \approx \sum_{\ell=1}^{L} p_\ell \widehat{V}^Q_{\Delta t}\Big(t_{n+1}, \mathrm{Proj}_{\Gamma_{n+1}}\big(G_{\Delta t}(z, a, e_\ell)\big)\Big), \quad \text{for } z \in \Gamma_n, \tag{3.17}$$

where:

- $G_{\Delta t}$ is defined, using the notations introduced in (3.3), as

$$G_{\Delta t}(z, a, \varepsilon) = z + b(z, a)\Delta t + \sigma_0(z, a)\sqrt{\Delta t}\, \varepsilon. \tag{3.18}$$

- $\mathrm{Proj}_{\Gamma_n}(.)$ stands for the Euclidean projection on Γ_n.
- $\Gamma = \{e_1, \dots, e_L\}$ and $\{p_\ell\}_{1 \leqslant \ell \leqslant L}$ are respectively the optimal L-quantizer and its associated sequence of weights of the exogenous noise ε. See above for more details.

An optimal feedback control at time t_n and point $z \in \Gamma_n$ is approximated by the expression (see Subsection 3.4 for more practical details on the computation of the argmax):

$$\hat{\alpha}^Q_{t_n}(z) = \underset{a \in A}{\mathrm{argmax}} \left\{ f(z, a)\Delta t + \sum_{\ell=1}^{L} p_\ell \widehat{V}^Q_{\Delta t}\Big(t_{n+1}, \mathrm{Proj}_{\Gamma_{n+1}}\big(G_{\Delta t}(z, a, e_\ell)\big)\Big) \right\}. \tag{3.19}$$

The value function at time t_n, is then estimated using the piecewise constant approximation and value iteration procedure as

$$\widehat{V}^{\mathrm{Q}}_{\Delta t}(t_n, z) = f\big(Z^m_{t_n}, \hat{\alpha}^Q_{t_n}(z)\big)\Delta t + \sum_{\ell=1}^{L} p_\ell \widehat{V}^Q_{\Delta t}\left(t_{n+1}, \mathrm{Proj}_{\Gamma_{n+1}}\Big(G_{\Delta t}\big(z, \hat{\alpha}^Q_{t_n}(z), e_\ell\big)\Big)\right).$$

Remark 3.2. *Clearly, the constant piecewise approximation can be easily extended to control problems of all dimensions $d \geqslant 1$. However the latter is, in most cases, not continuous w.r.t. the control variable since it remains constant on each Voronoi cells (see, e.g., Figure 1 p.135). As a result, the optimization process over the control space suffers from high instability and inaccuracy, which implies a poor estimation of the value function $V(t_n, \cdot)$.*

Semi-linear interpolation Once again, we assume here that we already have $\widehat{V}^{\mathrm{Q}}_{\Delta t}(t_{n+1}, \cdot)$, an estimate of the value function at time t_{n+1}, with $n \in \{0, \dots, N-1\}$, and wish to provide an estimation of the conditional expectation in the particular case where the controlled process lies in dimension d=1. Consider the following piecewise linear approximation of the conditional expectation, which is continuous w.r.t. the control variable a:

$$\mathbb{E}^a_{n,z}\big[\widehat{V}^{\mathrm{Q}}_{\Delta t}\big(t_{n+1}, Z_{t_{n+1}}\big)\big] \approx \sum_{\ell=1}^{L} p_\ell \Big[\lambda_a^{e_\ell, z} \widehat{V}^{\mathrm{Q}}_{\Delta t}\big(t_{n+1}, z_+\big) + (1 - \lambda_a^{e_\ell, z})\widehat{V}^{\mathrm{Q}}_{\Delta t}\big(t_{n+1}, z_-\big)\Big], \quad \text{for } z \in \Gamma_n, \tag{3.20}$$

where for all $\ell = 1, \dots, L$, z_- and z_+ are defined as follows:

- z_- and z_+ are the two closest states in Γ_{n+1} from $G_{\Delta t}(z,a,e_\ell)$, such that $z_- < G_{\Delta t}(z,a,e_\ell) < z_+$, if such states exist; and, in this case, we define $\lambda_a^{e_\ell,z} = \frac{G_{\Delta t}(z,a,e_\ell)-z_-}{z_+-z_-}$.
- Otherwise, z_- and z_+ are equal and defined as the closest state in Γ_{n+1}^Z from $G_{\Delta t}(z,a,e_\ell)$ and we define $\lambda_a^{e_\ell,z} = 1$.

Remark 3.3. *This second approximation is continuous w.r.t. the control variable, which brings stability and accuracy to the optimal control task (see Subsection 3.4), and also ensures an accurate estimate of the value function at time t_n. We will mainly use this approximation in the numerical tests (see Section 4).*

Remark 3.4. *Although the dimension $d = 1$ plays a central role to define clearly the states z_- and z_+ in (3.20), the semi-linear approximation can actually be generalized to a certain class of control problems of dimension greater than 1, using multi-dimensional Quantization (see, e.g., the comments on the Q-algorithm designed to solve the Portfolio Optimization example, in Subsection 4.1.2). However, it is not well-suited to solve numerically general control problems in dimension greater than 1. For these cases, other interpolating methods such as the use of Gaussian processes are more appropriated (see, e.g., [18] for an introduction on the use of Gaussian processes in Regression Monte Carlo).*

The optimal feedback control at time t_n and point $z \in \Gamma_n$ is approximated as (see Subsection 3.4 for more practical details on the computation of the argmax):

$$\hat{\alpha}_{t_n}^{\mathrm{Q}}(z) = \underset{a\in A}{\operatorname{argmax}} \left\{ f(z,a)\Delta t + \sum_{\ell=1}^{L} p_\ell \Big[\lambda_a^{e_\ell,z} \widehat{V}_{\Delta t}^{\mathrm{Q}}\big(t_{n+1}, z_+\big) + (1-\lambda_a^{e_\ell,z}) \widehat{V}_{\Delta t}^{\mathrm{Q}}\big(t_{n+1}, z_-\big) \Big] \right\}. \tag{3.21}$$

The value function at time t_n is then estimated using the semi-linear approximation and value iteration procedure as

$$\widehat{V}_{\Delta t}^{\mathrm{Q}}(t_n,z) = f\big(z, \hat{\alpha}_{t_n}^{\mathrm{Q}}(z)\big)\Delta t + \sum_{\ell=1}^{L} p_\ell \Big[\lambda_{\hat{\alpha}_{t_n}^{\mathrm{Q}}(z)}^{e_\ell,z} \widehat{V}_{\Delta t}^{\mathrm{Q}}\big(t_{n+1}, z_+\big) + (1-\lambda_{\hat{\alpha}_{t_n}^{\mathrm{Q}}(z)}^{e_\ell,z}) \widehat{V}_{\Delta t}^{\mathrm{Q}}\big(t_{n+1}, z_-\big) \Big],$$

where z_+ and z_- are defined using the control $\hat{\alpha}_{t_n}^{\mathrm{Q}}(z)$. See (3.20) for their definitions.

3.3. **Training points design**

We discuss here the choice of the training measure μ and the sets $(\Gamma_n)_{n=0,\dots,N-1}$ used to compute the numerical approximations in Regression Monte Carlo and Quantization. Two cases are considered in this section. The first one is a knowledge-based selection, relevant when the controller knows with a certain degree of confidence where the process has to be driven in order to optimize her reward functional. The second case, on the other hand, is when the controller has no idea where or how to drive the process to optimize the reward functional.

3.3.1. *Exploitation only strategy*

In the knowledge-based setting there is no need for exhaustive and expensive (in time mainly) exploration of the state space, and the controller can directly choose training sets Γ constructed from distributions μ that assign more points to the parts of the state space where the optimal process is likely to be driven.

In practice, at time t_n, assuming we know that the optimal process is likely to stay in the ball centered around the point m_n and with radius r_n, we chose a training measure μ_n centered around m_n as, for example $\mathcal{N}(m_n, r_n^2)$, and build the training set as sample of the latter. In the Regress-Later setting this can be done straightforwardly, while Control Randomization requires one to select a measure for the random control such that the controlled process Z is driven in such area of the state space.

Taking samples according to μ to build grids makes them random. Another choice, which we used in the Quantization-based algorithm, is to use the (deterministic) optimal grid of $\mathcal{N}(m_n, \sigma_n^2)$ with reduced size (typically take 50 points for a problem in dimension 1, 250 for one of dimension 2 when $\sigma_n^2 = 1$,...), which can be

found at www.quantize.maths-fi.com, to reduce the size of the training set and alleviate the complexity of the algorithms.

Remark 3.5. *As the reader will see, we chose the training sets based on the "exploitation only strategy" procedure, i.e. by guessing where to drive optimally the process, when solving the Liquidation Problem introduced in Subsection 4.1.*

3.3.2. *Explore first, exploit later*

Explore first: If the agent has no idea of where to drive the process to receive large rewards, she can always proceed to an exploration step to discover favorable subsets of the state space. To do so, the Γ_n, for $n = 0, \dots, N-1$, can be built as uniform grids that cover a large part of the state space, or μ can be chosen uniform on such domain. It is essential to explore far enough to have a well understanding of where to drive and where not to drive the process.

Exploit later: The estimates for the optimal controls at time t_n, $n = 0, \dots, N-1$, that come up from the *Explore first* step, are relatively good in the way that they manage to avoid the wrong areas of state space when driving the process. However, the training sets that have been used to compute the estimated optimal control are too sparse to ensure accuracy on the estimation. In order to improve the accuracy, the natural idea is to build new training sets by simulating M times the process using the estimates on the optimal strategy computed from the *Explore first* step, and then proceed to another estimation of the optimal strategies using the new training sets. This trick can be seen as a two steps algorithm that improves the estimate of the optimal control.

Remark 3.6. *In Control Randomization, multiple runs of the method are often needed to obtain precise estimates, because the initial choice of the dummy control could drive the training points far from where the optimal control would have driven them. In practice, after having computed an approximate policy backward in time, such policy is used to drive M simulations of the process forward in time, which in turn produce control paths that can be fed as a random controls in a new backward procedure, leading to more accurate results.*

Remark 3.7. *We applied the "explore first, exploit later" idea to solve the Portfolio Optimization problem introduced in Subsection 4.1.*

3.4. Optimal control searching

Assume in this section that we already have the estimates $\widehat{V}_{\Delta t}(t_k, \cdot)$ for the value function at time t_k, for $k = n+1, \dots, N$, and want to estimate $V(t_n, \cdot)$ the value function at time t_n.
The optimal control searching task consists in optimizing the function[5]

$$\widehat{Q}_n : (z, \cdot) \mapsto f(z, a)\Delta t + \hat{\mathbb{E}}^a_{n,z}\big[\widehat{V}_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\big]$$

over the control space A, for each $z \in \Gamma_n$, and where we denote by $\hat{\mathbb{E}}^a_{n,z}\big[\widehat{V}_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\big]$ an approximation of $\mathbb{E}^a_{n,z}\big[\widehat{V}_{\Delta t}(t_{n+1}, Z_{t_{n+1}})\big]$ using Regress-Later, or Control Randomization or Quantization-based methods (see Subsection 3.2). Once again, we remind that importance of this task is motivated by the dynamic programming principle stating that for all $n = 0, \dots, N-1$, we can approximate the value function at time n as follows

$$\widehat{V}_{\Delta t}(t_n, z) = \sup_{a \in A} \widehat{Q}_n(z, a), \tag{3.22}$$

where $\widehat{V}_{\Delta t}(t_n, \cdot)$ is our desired estimate of the value function at time n.

[5] often referred to as the Q-function, or action-value function, in the reinforcement learning literature. Be aware that Q stands here for the "Quality" of an action taken in a given state, and in particular does not refer to Quantization.

3.4.1. *Low cardinality control set*

In the case where the control space A is discrete (with a relatively small cardinality), one can solve the optimization problem by an exhaustive search over all the available controls without compromising the computational speed.

Remark 3.8. *Note that in the case where the control space is continuous, one can always discretize the latter in order to rely on the effectiveness of extensive search to solve the optimal control problem. However, the control space discretization brings an error. So the control might have to include a high number of points in the discretization in order to reduce the error thereby causing a considerable slow down of the computations.*

3.4.2. *High cardinality/continuous control space*

If we assume differentiability almost everywhere, as follows from the semi-linear approximation in Quantization, and most choices of basis functions in Regression Monte Carlo, we can carry on the optimization step by using some gradient-based algorithm for optimization of differentiable functions. Actually, many optimizing algorithms (Brent, Golden-section Search, Newton gradient-descent,...) are already implemented in standard libraries of most programming languages like Python (see, e.g., package `scipy.optimize`), Julia (see, e.g., package `Optim.jl`), C and C++ (see, e.g., package `NLopt`).

Remark 3.9. *When the control is of dimension 1, polynomials of order smaller than 5 are employed as basis functions in Regression Monte Carlo as well as for the running reward f. The optimal control can then be computed analytically as a function of the regression coefficients, since every polynomial equation of order smaller than 4 can be solved by radicals.*

Concretely, in all the examples considered in Section 4, we used the Golden-section Search or the Brent methods when testing Quantization-based algorithm to find the optimal controls at each point of the grids. These algorithms were very accurate to find the optimal controls, and we made use of Remark 3.9 to find the optimal controls using the Regress-Later-based algorithm.

3.5. Upper and lower bounds

After completing the backward procedure, we can compute an unbiased estimation of the value of the control policy by using Monte Carlo simulations and sample average. Assume already computed (or simply available) the matrix of regression coefficients, in the case of Regression Monte Carlo, and discrete probability law $\hat{p}$ for Quantization, we can use this information to implicitly compute the control and simulate forward many trajectories of the controlled process starting from a common initial condition. We can then evaluate the average performance measure by computing the sample average of the rewards collected on each trajectory. Denoting such approximation by $\hat{\hat{V}}_{\Delta t}(0,z)$, and recalling that by definition $J_{\Delta t}(0,z,\alpha) \leqslant J_{\Delta t}(0,z,\alpha^*)$, for all $\alpha \in \mathcal{A}_{\Delta t}$ and where α^* represents the optimal control process; it holds $\hat{\hat{V}}_{\Delta t}(0,z) \leqslant V_{\Delta t}(0,z)$, for $z \in \mathbb{R}^d$.

The argument above implies that, neglecting the time-discretization error, we obtain a lower bound for $V_{\Delta t}(0,\cdot)$ by evaluating the estimated policy. To get an upper bound of the value function via duality, see [14] based on [23], and [3].

3.6. Pseudo-codes

In this section, we present the pseudo-code for the three approaches presented in the previous sections. For simplicity, we will only show the algorithms designed using value iteration procedure. However, the performance iteration update rule can be substituted in the codes below provided that forward simulations are run to obtain a pathwise realization of the controlled process and associated rewards.

3.6.1. *Pseudo-code for a Regress-Later-based algorithm*

We present in Algorithm 1 a pseudo-code to estimate $V_{\Delta t}(t_n, \cdot)$, for $n = 0, \dots, N-1$, using Value Iteration and based on Regress-Later method. For $n = 0, \dots, N-1$, we denote by $\hat{V}^{\mathrm{RL}}_{\Delta t}(t_n, \cdot)$ the derived estimation of $V_{\Delta t}(t_n, \cdot)$, and will refer to it as the RLMC algorithm in the numerical tests presented in Section 4.

Note that we use the same training measure μ at each time step so that there is only one covariance matrix to estimate (since $\mathcal{A}_{t_n}$ is the same for all $n = 0, \dots, N-1$). Denote by $\hat{\mathcal{A}}^M$ the estimator, as defined in (3.14).

Algorithm 1 Regress-Later Monte Carlo algorithm (RLMC) - Value iteration

Inputs:

- M: number of training points,
- μ: distribution of training points,
- K: number of basis functions,
- $\{\phi_k\}_{k=1}^K$: family of basis functions.

1: Estimate the covariance matrix $\hat{\mathcal{A}}^M$.
2: Generate i.i.d. training points $\{Z^m_{t_N}\}_{m=1}^M$ accordingly to the distribution μ.
3: Initialize the value function $\hat{V}^{\mathrm{RL}}_{\Delta t}(t_N, Z^m_{t_N}) = g(Z^m_{t_N}),\ \forall m = 1, \dots, M$.
4: **for** $n = N-1$ to 0 **do**
5: $\quad \hat{\beta}^n = \hat{\mathcal{A}}_M^{-1} \frac{1}{M} \sum_{m=1}^M \left[\hat{V}^{\mathrm{RL}}_{\Delta t}(t_{n+1}, Z^m_{t_{n+1}}) \phi(Z^m_{t_{n+1}}) \right]$.
6: $\quad$ Generate a new layer of i.i.d. training points $\{Z^m_{t_n}\}_{m=1}^M$ accordingly to the distribution μ.
7: $\quad$ For all $m = 1, \dots, M$ do

$$\hat{V}^{\mathrm{RL}}_{\Delta t}(t_n, Z^m_{t_n}) = \sup_{a \in A} \Big\{ f(Z^m_{t_n}, a)\Delta t + \sum_{k=1}^K \hat{\beta}^n_k \hat{\phi}^n_k(Z^m_{t_n}, a) \Big\}.$$

8: **Evaluate the policy to obtain** $\hat{\hat{V}}^{\mathrm{RL}}_{\Delta t}$.

Outputs: $\{\hat{\beta}^n_k\}_{n,k=1}^{N,K}$, $\hat{\hat{V}}^{\mathrm{RL}}_{\Delta t}(0, z)$ for $z \in \mathbb{R}^d$.

3.6.2. *Pseudo-code for a Control Randomization-based algorithm*

We present in Algorithm 2 a pseudo-code to estimate $V_{\Delta t}(t_n, \cdot, \cdot)$, for $n = 0, \dots, N-1$, using Value Iteration and based on Control Randomization method. For $n = 0, \dots, N-1$, we denote by $\hat{V}^{\mathrm{CR}}_{\Delta t}(t_n, \cdot)$ the derived estimation of $V_{\Delta t}(t_n, \cdot)$, and will refer to it as the CR algorithm in the numerical tests presented in Section 4.

3.6.3. *Pseudo-code for a Quantization-based algorithm*

We present in Algorithm 3 a pseudo-code to estimate $V_{\Delta t}(t_n, \cdot, \cdot)$, for $n = 0, \dots, N-1$, using value iteration procedure and based on Quantization method. For $n = 0, \dots, N-1$, we denote by $\hat{V}^{\mathrm{Q}}_{\Delta t}(t_n, \cdot)$ the derived estimation of $V_{\Delta t}(t_n, \cdot)$, and will refer to it as the Q-algorithm in the numerical tests presented in Section 4.

Note that we made use of a piecewise constant approximation of conditional expectations to approximate $\hat{V}^{\mathrm{Q}}_{\Delta t}(t_n, \cdot)$ in order to keep the algorithm simple. Also, note that, as said previously, in most of the numerical tests run in Section 4, we will use optimal grids available at `www.quantize.maths-fi.com` and will take $L = 25$ to 50 points for the size of the optimal grid of the Gaussian noise ε.

Algorithm 2 Control Randomization algorithm (CR) - Value iteration

Inputs:

- M: number of training points,
- μ: initial distribution of dummy control,
- K: number of basis functions,
- $\{\phi_k\}_{k=1}^K$: family of basis functions.

1: Estimate the covariance matrix $\hat{\mathcal{A}}^M$.
2: Generate m trajectories, $\{Z_{t_n}^m, I_{t_n}^m\}_{n=0,m=1}^{N,M}$, where $Z_{t_n}^m$ is driven by $I_{t_n}^m$, and the $I_{t_n}^m$ are i.i.d with distribution μ.
3: Initialize the value function $\hat{V}_{\Delta t}^{\mathrm{CR}}(t_N, Z_{t_N}^m) = g(Z_{t_N}^m),\ m = 1, \dots, M$.
4: **for** $n = N-1$ to 0 **do**
5: $\quad \hat{\beta}^n = (\hat{\mathcal{A}}^M)^{-1} \frac{1}{M} \sum_{m=1}^M \Big[\hat{V}_{\Delta t}^{\mathrm{CR}}(t_{n+1}, Z_{t_{n+1}}^m)\phi(Z_{t_n}^m, I_{t_n}^m)\Big]$.
6: $\quad$ For all $m = 1, \dots, M$ do

$$\hat{V}_{\Delta t}^{\mathrm{CR}}(t_n, Z_{t_n}^m) = \sup_{a\in A}\Big\{f(Z_{t_n}^m, a)\Delta t + \sum_{k=1}^K \hat{\beta}_k^n \phi_k(Z_{t_n}^m, a)\Big\}.$$

7: **Evaluate the policy to obtain** $\hat{\hat{V}}_{\Delta t}^{\mathrm{CR}}$.

Outputs: $\{\hat{\beta}_k^n\}_{n=0,k=1}^{N,K}$, $\hat{\hat{V}}_{\Delta t}^{\mathrm{CR}}(0, z)$ for $z \in \mathbb{R}^d$.

Algorithm 3 Quantization algorithm (Q) - Value iteration

Inputs:

- Γ_k, $k = 0, \dots, N$: grids of training points in $\mathbb{R}^d$,
- $\Gamma = \{e_1, \dots, e_L\}$, $(p_\ell)_{1\leqslant\ell\leqslant L}$: the L-optimal grid of the exogenous noise ε, and its associated weights,

1: Initialize the estimated value function at time N: $\hat{V}_{\Delta t}^{\mathrm{Q}}(t_N, z) = g(z), \quad \forall z \in \Gamma_N$.
2: **for** $n = N-1$ to 0 **do**
3: $\quad$ Estimate the value function at time t_n as follows:

$$\widehat{V}_{\Delta t}^{\mathrm{Q}}(t_n, z) = \max_{a\in A}\left[f(z,a)\,\Delta t + \sum_{\ell=1}^L p_\ell \widehat{V}_{\Delta t}^{\mathrm{Q}}\Big(t_{n+1}, \mathrm{Proj}_{\Gamma_{n+1}}\big(G_{\Delta t}(z,a,e_\ell)\big)\Big)\right], \quad \forall z \in \Gamma_n. \tag{3.23}$$

4: **Evaluate the policy to obtain** $\hat{\hat{V}}_{\Delta t}^{\mathrm{Q}}$.

Outputs: $\big(\hat{\alpha}(t_n, z)\big)_{z\in\Gamma_n, 0\leqslant n\leqslant N-1}$, $\big(\hat{\hat{V}}_{\Delta t}^{\mathrm{Q}}(0, z)\big)_{z\in\Gamma_0}$.

4. Applications and numerical results

4.1. Portfolio Optimization under drift uncertainty

4.1.1. *The model*

We consider a financial market model with one risk-free asset, assumed to be equal to one, and d risky assets of price process $S = (S^1, \dots, S^d)$ governed

$$dS_t = \mathrm{diag}(S_t)(\beta_t dt + \sigma dB_t^0), \quad S_0 = s_0 \in \mathbb{R}^d,$$

where B^0 is a d-dimensional Brownian motion on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P}^0)$, σ is the $d \times d$ invertible matrix volatility coefficient, assumed to be known and constant. However, the drift (β_t) of the asset (which is typically a diffusion process governed by another independent Brownian motion B) is unknown and unobservable like the Brownian motion B^0. The agent can actually only observe the stock prices S, and we denote by $\mathbb{F}^S$ the filtration generated by the price process S, which should be view as the available information. In this context, we shall consider two important classes of optimization problems in finance:

(1) *Portfolio Liquidation.* We consider the problem of an agent (trader) who has to liquidate a large number y_0 of shares in some asset (we consider one stock, $d = 1$) within a finite time T, and faces execution costs and market price impact. In contrast with frictionless Merton problem, we do not consider mark-to-market value of the portfolio and instead consider separately the amount on the cash account and the inventory Y, i.e., the position or number of shares held at any time. The strategy of the agent is then described by a real-valued $\mathbb{F}^S$-adapted process α, representing the velocity at which she buys ($\alpha_t > 0$) or sells ($\alpha_t < 0$) the asset, and the inventory is thus given by

$$Y_t \;=\; y_0 + \int_0^t \alpha_u du, \quad 0 \leqslant t \leqslant T.$$

The objective of the trader is to minimize over α the total liquidation cost

$$J_1(\alpha) \;=\; \mathbb{E}^0\big[\int_0^T \alpha_t\big(S_t + f(\alpha_t)\big)dt + \ell(Y_T)\big]$$

where $f(.)$ is an increasing function with $f(0) = 0$, representing a temporary price impact, and $\ell(.)$ is a loss function, i.e., a convex function with $\ell(0) = 0$, penalizing the trader when she does not succeed to liquidate all her shares.

(2) *Portfolio Selection.* The set $\mathcal{A}$ of portfolio strategies, representing the amount invested in the assets, consists in all $\mathbb{F}^S$-adapted processes α valued in some set A of $\mathbb{R}^d$, and satisfying $\int_0^T |\alpha_t|^2 dt < \infty$. The dynamics of wealth process $X = X^\alpha$ associated to a portfolio strategy α is then governed by

$$\begin{aligned} dX_t &= \alpha_t S_t^{-1} dS_t \\ &= \alpha_t.\beta_t dt + \alpha_t^\intercal \sigma dB_t^0, \quad X_0 \;=\; x_0 \in \mathbb{R}, \end{aligned}$$

and as in Merton Portfolio Selection problem, the objective of the agent is to maximize over portfolio strategies the utility of terminal wealth

$$J_2(\alpha) \;=\; \mathbb{E}^0[U(X_T)],$$

where U is a utility function on $\mathbb{R}$, e.g., CARA function $U(x) = -\exp(-px)$, $p > 0$.

Let us show how one can reformulate the above problems into a McKean-Vlasov type problem under partial observation and common noise as described in Section 1. We first introduce the so-called probability reference $\mathbb{P}$, which makes the observation price process a martingale. Let us then define the process

$$Z_t \;=\; \exp\big(-\int_0^t \sigma^{-1}\beta_u dB_u^0 - \frac{1}{2}\int_0^t |\sigma^{-1}\beta_u|^2 du\big), \quad 0 \leqslant t \leqslant T,$$

which is a $(\mathbb{P}^0, \mathbb{F})$-martingale (under suitable integrability conditions on β), and defines a probability measure $\mathbb{P} \sim \mathbb{P}^0$ through its density: $\frac{d\mathbb{P}}{d\mathbb{P}^0}\Big|_{\mathcal{F}_t} = Z_t$, and under which the process

$$W_t^0 \;=\; B_t^0 + \int_0^t \sigma^{-1}\beta_u du, \quad 0 \leqslant t \leqslant T,$$

is a $(\mathbb{P},\mathbb{F})$-Brownian motion by Girsanov's theorem, and the dynamics of S is

$$dS_t \;=\; \mathrm{diag}(S_t)\sigma dW_t^0.$$

Notice that $\mathbb{F}^S=\mathbb{F}^0$ the filtration generated by W^0. We also denote by $L_t=1/Z_t$ the $(\mathbb{P},\mathbb{F})$-martingale governed by

$$dL_t \;=\; L_t\sigma^{-1}\beta_t.dW_t^0.$$

Next, we use Bayes formula and rewrite the gain (resp. cost) functionals of our two portfolio optimization problems as

$$\begin{aligned}
J_1(\alpha) &= \mathbb{E}\Big[\int_0 L_t\alpha_t(S_t+f(\alpha_t))dt + L_T\ell(Y_T)\Big]\\
&= \mathbb{E}\Big[\int_0^T \bar L_t^0\alpha_t(S_t+f(\alpha_t))dt + \bar L_T^0\ell(Y_T)\Big]\\
&= \mathbb{E}\Big[\int_0^T \bar L_t^0\alpha_t(\bar S_t^0+f(\alpha_t))dt + \bar L_T^0\ell(\bar Y_T^0)\Big],\\
J_2(\alpha) &= \mathbb{E}\big[L_TU(X_T)\big] \;=\; \mathbb{E}\big[\bar L_T^0U(X_T)\big] \;=\; \mathbb{E}\big[\bar L_T^0U(\bar X_T^0)\big]
\end{aligned}$$

where $\bar L_t^0=\mathbb{E}[L_t|W^0]=\int \ell\mathbb{P}_{L_t}^{W^0}(d\ell)$, $\bar X_t^0=\mathbb{E}[X_t|W^0]=\int x\mathbb{P}_{X_t}^{W^0}(dx)=X_t$, $\bar Y_t^0=\mathbb{E}[Y_t|W^0]=\int y\mathbb{P}_{Y_t}^{W^0}(dy)=Y_t$, $\bar S_t^0=\mathbb{E}[S_t|W^0]=\int s\mathbb{P}_{S_t}^{W^0}(ds)=S_t$, and we used the law of conditional expectations and the fact that S, X and Y are $\mathbb{F}^0$-adapted. This formulation of the functional J_1 (resp. J_2) fits into the MKV framework of Section 1 with state variables (X,L,β) (resp. (Y,S,L,β))

We now consider the particular case when β is an $\mathcal{F}_0$-measurable random variable distributed according to some probability distribution $\nu(db)$: this corresponds to a Bayesian point of view when the agent's belief about the drift is modeled by a prior distribution. In this case, let us show how our partial observation problem can be embedded into a finite-dimensional full observation Markov control problem. Indeed, by noting that β is independent of the Brownian motion W^0 under $\mathbb{P}$, we have

$$\bar L_t^0 \;=\; \mathbb{E}\Big[\exp\big(\sigma^{-1}\beta.W_t^0-\frac{1}{2}|\sigma^{-1}\beta|^2t\big)\big|W^0\Big] \;=\; F(t,W_t^0),$$

where

$$F(t,w) \;=\; \int \exp\big(\sigma^{-1}b.w-\frac{1}{2}|\sigma^{-1}b|^2t\big)\nu(db).$$

Hence, the functionals J_1 and J_2 can be written as

$$J_1(\alpha) \;=\; \mathbb{E}\Big[\int_0^T F(t,W_t^0)\alpha_t(S_t+f(\alpha_t))dt + F(T,W_T^0)\ell(Y_T)\Big], \tag{4.1}$$

$$J_2(\alpha) \;=\; \mathbb{E}\big[F(T,W_T^0)U(X_T)\big]. \tag{4.2}$$

We are then reduced to a $(\mathbb{P},\mathbb{F}^0)$-control problem with state variables (W^0,X) for problem (1) and (W^0,S,Y) for problem (2), with the following dynamics under $\mathbb{P}$:

$$dS_t \;=\; \mathrm{diag}(S_t)\sigma dW_t^0, \quad S_0=s_0\in(\mathbb{R}_+)^d, \tag{4.3}$$

$$dX_t \;=\; \alpha_t^\intercal\sigma dW_t^0, \quad X_0=0, \tag{4.4}$$

$$dY_t \;=\; \alpha_t dt, \quad Y_0=y_0\in\mathbb{R}_+. \tag{4.5}$$

Remark 4.1. The case where the drift β is modeled by a linear Gaussian process is another example of partial observation. This would lead to the well-known Kalman-Bucy filter, hence to a finite-dimensional control problem. However, for general unobserved drift process β, we fall into an infinite dimensional control problem involving the filter process.

4.1.2. *Numerical results*

Let us now illustrate numerically the impact of uncertain Bayesian drift on the Portfolio Liquidation problem and the Portfolio Selection problem in dimension $d = 1$, by considering a Gaussian prior distribution $\beta \sim \nu = \mathcal{N}(b_0, \gamma_0^2)$, with $b_0 \in \mathbb{R}$ and $\gamma_0 > 0$. In this case, F is explicitly given by:

$$F(t,w) = \frac{\sigma}{\sqrt{\sigma^2 + \gamma_0^2 t}} \exp\Big(\frac{1}{2(\sigma^2+\gamma_0^2 t)}(-b_0^2 t + 2b_0\sigma w + \gamma_0^2 w^2)\Big).$$

1. Portfolio Liquidation. Let us first consider the Portfolio Liquidation problem (1) with a linear price impact function $f(a) = \gamma a$, $\gamma > 0$, and a quadratic loss function $\ell(y) = \eta y^2$, $\eta > 0$. The optimal trading rate is given by (see [21])

$$\alpha_t^* \quad = \quad -\frac{Y_t^*}{T-t+\gamma/\eta} + \frac{1}{2\gamma}\Big(\frac{1}{T-t+\gamma/\eta}\int_t^T \mathbb{E}^0[S_u|\mathcal{F}_t^S]du - S_t\Big)$$

where Y^* is the associated inventory with feedback control α^*: $dY_t^* = \alpha_t^* dt$, $Y_0^* = y_0$. Since we consider a Gaussian prior $\mathcal{N}(b_0, \gamma_0^2)$ for β, the optimal trading rate is explicitly given by

$$\alpha_t^* = -\frac{1}{T-t+\gamma/\eta}\left\{Y_t^* + \frac{1}{2\gamma}\left[-\frac{1}{\gamma_0}\sqrt{\frac{\pi}{2}}e^{-\frac{b_0^2}{2\gamma_0^2}}\left(\mathrm{erfi}\left(\frac{b_0+\gamma_0^2(T-t)}{\sqrt{2}\gamma_0}\right) - \mathrm{erfi}\left(\frac{b_0}{\sqrt{2}\gamma_0}\right)\right) + (T-t+\frac{\gamma}{\eta})\right]S_t\right\},$$

where `erfi` is the imaginary error function, defined as:

$$\mathrm{erfi}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{t^2}dt.$$

Remark 4.2. *In the particular case where the price process is a martingale, i.e., $b_0 = 0$, and in the limiting case when the penalty parameter η goes to infinity, corresponding to the final constraint $Y_T = 0$, we see that α_t^* converges to $-Y_t^*/(T-t)$, hence it becomes independent of the price process, and this leads to an explicit optimal inventory: $Y_t^* = y_0\frac{T-t}{T}$ with constant trading rate $\alpha_t^* = -y_0/T$. We retrieve the well-known VWAP strategy obtained in [1].*

We solve the problem numerically, taking $N = 100$ for the time discretization, and fixing the other parameters as follows: γ=5, S_0=6, Y_0=1, η=100 and σ=0.4. We run two sets of forward Monte Carlo simulations for $b_0 = 0.1$, $T = 1$ and $b_0 = -0.1$, $T = 0.5$ changing the value of γ_0. We tested the Regress-Later Monte Carlo (RLMC), the Control Randomization (CR) and the Quantization (Q) algorithms. In particular, we wanted to compare the performance of these algorithms with $(\alpha_{t_n}^*)_{n=0}^{N-1}$, where α^*, defined above, is the optimal strategy associated to the continuous-time Portfolio Liquidation problem. We refer to this discrete-time strategy as α^* (i.e., re-using the same notation), and we use Opt, or *continuous-time* optimal strategy when we want to stress the fact that this strategy is optimal for the *continuous-time* control problem, and not for the discrete time one. We also tested a benchmark strategy (Bench) which consists in liquidating the inventory at a constant rate $-y_0/T$. The test consisted in computing the estimates $\hat{\hat{V}}_{\Delta t}(t_0 = 0, S_0 = 6, Y_0 = 1)$ associated to the different algorithms.

We display the results obtained by the different algorithms in Table 1 and plot them in Figure 2. One can observe in Figure 2 that for $\Delta t = \frac{1}{100}$ the estimations $\hat{\hat{V}}_{\Delta t}(t_0 = 0, S_0 = 6, Y_0 = 1)$ of the value function $V_{\Delta t}(t_0 = 0, S_0 = 6, Y_0 = 1)$, provided by RLMC, CR or Q-based methods, are sometimes such that

$$\hat{\hat{V}}_{\Delta t}(t_0 = 0, S_0 = 6, Y_0 = 1) \leqslant \hat{J}_{\Delta t}(t_0 = 0, S_0 = 6, Y_0 = 1, \alpha^*),$$

where $\hat{J}_{\Delta t}(\cdot, \cdot, \cdot, \alpha^*)$ is a Monte Carlo estimate of $J_{\Delta t}(\cdot, \cdot, \cdot, \alpha^*)$ applying strategy $(\alpha^*_{t_n})_{n=0}^{N-1}$ (see in Figure 2 the curve Opt). It means that RLMC, CR, or Q-based methods sometimes provide better estimations of the optimal strategy than α^* for the discrete time control problem. However, since under suitable conditions (see, e.g., [16]), the optimal strategy for the discrete time control problem $\alpha^*_{\Delta t}$ converges toward α^*, i.e. we have $\alpha^*_{\Delta t} \xrightarrow[\Delta t \to 0]{} \alpha^*$, then it holds:

$$\hat{J}_{\Delta t}(t_0 = 0, S_0 = 6, Y_0 = 1, \alpha^*) \xrightarrow[\Delta t \to 0]{} V(t_0 = 0, S_0 = 6, Y_0 = 1).$$

Figure 3 shows a sample of the inventory $(Y_t)_{t \in [0,T]}$ when the agent follows α^* and the Quantization algorithm. One can notice that given the chosen penalization parameters, it is optimal to short some stocks at terminal time. Finally, notice that the concavity of the curves comes from the fact that the running cost does not penalize the inventory. If so, we expect the curves of the inventory w.r.t. time to be convex, see, e.g., [11].

Details on the RL and CR algorithms implementation

The implementation of Regression Monte Carlo algorithms has required intense tuning and the use of the performance iteration technique introduced in Subsection 3.3.2 in order to obtain satisfactory results. Paramount is, in addition, the distribution chosen for the training points in Regress-Later and for the initial control in Control Randomization. The problem of finding the best set of data to provide to the backward procedure is similar in the two Regression Monte Carlo algorithms. However little study is available in the literature; for more details on this problem in the Regress-Later setting see [19] and [2]. In the case of RL algorithm a training measure μ_n has been chosen in order to sufficiently explore the state space in the Y dimension, in particular we considered $\mu_n = \mathcal{U}[-0.5, 0.5 + \frac{T - t_n}{t_n}]$. Similarly for CR we seek a distribution of the random control such that the controlled process Y results in having a distribution similar to μ_n. In order to achieve such goal we follow the "explore first, exploit later" approach presented in Subsection 3.3.2 and use a perturbed version of the empirical distribution of the control (to avoid concentration of the training points) obtained at previous iteration of the method to determine the random control at next iteration of the method.

In order to choose the basis functions, we used the fact that we expect the value function to be convex in the Y dimension with minimum around the optimal inventory level and monotone in the S dimension. For RL algorithm we choose therefore the following set of basis functions: $\{s, y, y^2, sy, sy^2\}$, where we take the square function y^2 as a general approximator for convex functions around their minima (where we expect the measure μ_n to be concentrated). On the other hand, CR requires that we guess what the functional form of the conditional expectation of the value function is with respect to the control process. Considering our argument on square function approximating general convex functions we choose to add the set $\{\alpha, \alpha^2, \alpha y, \alpha s\}$ to the set of basis functions used by RL.
Note that there is no need for time-consuming optimal control searching with such a choice of basis functions, as explained in Remark 3.9.

Finally note that we observed very high volatility in the quality of the policy estimated by control randomization. For this reason we estimated the policy 50 times, and report in Table 1 the results provided by the best performing one; increasing the number of training points further affects the variability only marginally.

Details on the Q algorithm implementation

To numerically solve this example, we used the optimal grid of the Gaussian random variable with $L = 50$ points, denoted by Γ_L^ε, to define the grid[6] $\Gamma_n^W = t_n\Gamma_L^\varepsilon$ that discretizes W_{t_n}, the Brownian motion at time t_n, and the grid $\Gamma_n^Y = Y_0 - \frac{t_n}{T} + t_n\Gamma_L^\varepsilon$ that discretizes Y_{t_n}, the inventory at time t_n, for $n = 0, ..., N$. Note that Γ_n^Y, for $n = 0, ..., N$, is centered at point $Y_0 - \frac{t_n}{T}$ because we guessed that the optimal liquidation rate was close to $\frac{Y_0}{T}$ (see Figure 3 to check that our guess is correct).
We then considered the grid $\Gamma_n = \Gamma_n^W \times \Gamma_n^Y$ to discretize $Z_{t_n} = (W_{t_n}, Y_{t_n})$, $n = 0, ..., N$.

We first tried to design a quantization algorithm using the following expression for the conditional expectation approximations:

$$\begin{aligned}\mathbb{E}^a_{n,(w,y)}\Big[\widehat{V}^{\mathrm{Q}}_{\Delta t}\Big(t_{n+1}, \mathrm{Proj}_{\Gamma^W_{n+1}}\big(W_{t_{n+1}}\big), \mathrm{Proj}_{\Gamma^Y_{n+1}}\big(Y_{t_{n+1}}\big)\Big)\Big] \qquad (4.6)\\ \approx \sum_{\ell=1}^{L} p_\ell \widehat{V}^{\mathrm{Q}}_{\Delta t}\Big(t_{n+1}, \mathrm{Proj}_{\Gamma^W_{n+1}}\big(G_{\Delta t}((w,y),a,e_\ell)\big), \mathrm{Proj}_{\Gamma^Y_{n+1}}\big(G_{\Delta t}((w,y),a,e_\ell)\big)\Big),\\ \text{for } (w,y,a) \in \Gamma_n^W \times \Gamma_n^Y \times A,\end{aligned}$$

where the first and second components of the process $Z = (W, Y)$ are projected respectively on the grids Γ_n^W and Γ_n^Y; and $\mathrm{Proj}_{\Gamma_n^W}$ (resp. $\mathrm{Proj}_{\Gamma_n^Y}$) stands for the Euclidean projection of the first (resp. second) component of $Z = (W, Y)$ on Γ_n^W (resp. Γ_n^Y).
This approximation belongs to the family of constant piecewise approximations, and is in the spirit of multidimensional component-wise-quantization methods already studied in the literature (see, e.g., [10]).
Unfortunately, as it can be seen in Figure 1, approximation (4.6) is discontinuous w.r.t. the control variable a in such a way that the optimal control searching task suffered from instability and inaccuracy, which implied bad value function estimations at time $n = 0, ..., N-1$. We thus had to use a better conditional expectation approximation.

We then decided to smooth the previous approximation of the conditional expectations w.r.t. the control variable by considering the following

$$\begin{aligned}&\mathbb{E}^a_{n,(w,y)}\Big[\widehat{V}^{\mathrm{Q}}_{\Delta t}\Big(t_{n+1}, \mathrm{Proj}_{\Gamma^W_{n+1}}\big(W_{t_{n+1}}\big), \mathrm{Proj}_{\Gamma^Y_{n+1}}\big(Y_{t_{n+1}}\big)\Big)\Big]\\ &\quad\approx \sum_{\ell=1}^{L} p_\ell \Big[\lambda_a^{e_\ell,(w,y)} \widehat{V}^{\mathrm{Q}}_{\Delta t}\big(t_{n+1}, \mathrm{Proj}_{\Gamma^W_{n+1}}\big[G^w_{\Delta t}((w,y),a,e_\ell)\big], y_+\big)\\ &\qquad + (1-\lambda_a^{e_\ell,(w,y)})\widehat{V}^{\mathrm{Q}}_{\Delta t}\Big(t_{n+1}, \mathrm{Proj}_{\Gamma^W_{n+1}}\big[G^w_{\Delta t}((w,y),a,e_\ell)\big], y_-\Big)\Big],\end{aligned}$$

where, in the spirit of the semi-linear approximation presented in Subsection 3.2, we have for all $\ell = 1, ..., L$:

- $G^w_{\Delta t}((w,y),a,e_\ell)$ and $G^y_{\Delta t}((w,y),a,e_\ell)$ respectively stand for the first and the second component of $G_{\Delta t}((w,y),a,e_\ell)$, i.e., $G_{\Delta t}((w,y),a,e_\ell) = \big(G^w_{\Delta t}((w,y),a,e_\ell), G^y_{\Delta t}((w,y),a,e_\ell)\big)$. See (3.18) for the definition of $G_{\Delta t}$.
- y_- and y_+ are the two closest states in Γ^Y_{n+1} from $G^y_{\Delta t}((w,y),a,e_\ell)$, such that $y_- < G^y_{\Delta t}((w,y),a,e_\ell) < y_+$ if such point exists; y_- and y_+ are equal to the closest state in Γ^Y_{n+1} from $G^y_{\Delta t}((w,y),a,e_\ell)$ otherwise.
- $\lambda_a^{e_\ell,(w,y)} = \frac{G^y_{\Delta t}((w,y),a,e_\ell)-y_-}{y_+-y_-}$ in the first case of the definition of y_- and y_+ above; $\lambda_a^{e_\ell,(w,y)} = 1$ otherwise.

[6] We use the notation $tB = \{tb, b \in B\}$, where $t \in \mathbb{R}$ and B is a set.

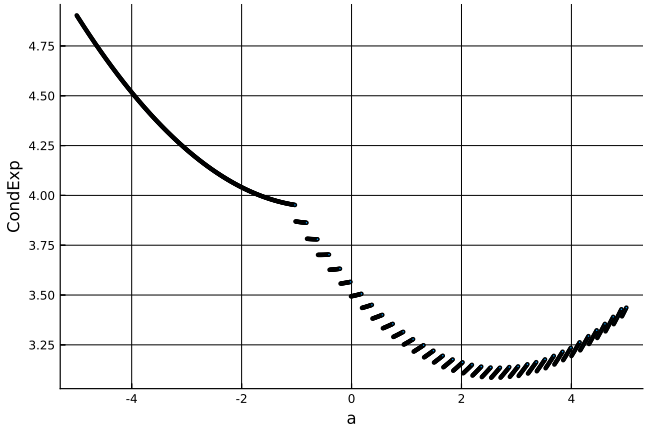


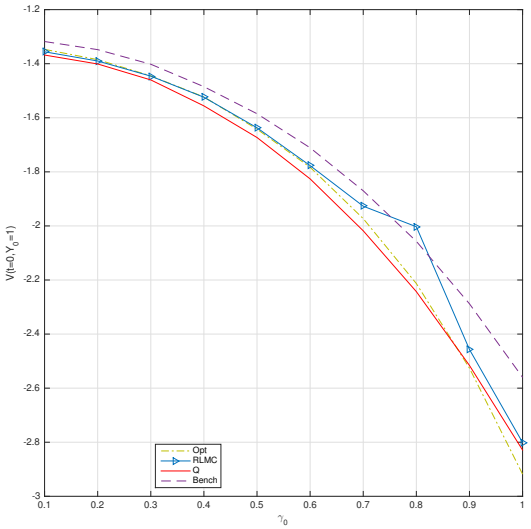
FIGURE 1. Plot of the quantized-based piecewise-constant approximation of the conditional expectation CondExp:
$a \mapsto \sum_{e\in\Gamma^\varepsilon} \mathbb{P}(\widehat{\varepsilon}=e)\widehat{V}^{\mathrm{Q}}_{\Delta t}\Big(t_{n+1}, \mathrm{Proj}_{\Gamma^W_{n+1}}\big(G_{\Delta t}((w,y),a,e)\big), \mathrm{Proj}_{\Gamma^Y_{n+1}}\big(G_{\Delta t}((w,y),a,e)\big)\Big)$.
We took $n = N-1$, $w = 0$, and $y = -0.18$ to plot the curve. Observe that the approximation is discontinuous w.r.t. the control variable a in such a way that it makes the search of the minimizer of this function very difficult by usual (gradient descent-based) algorithms. Also, observe that the minimum of the function, which is actually equal to the estimation of the value function at time $N-1$ at point $(w=0, y=-0.18)$, suffers from inaccuracy.

TABLE 1. Portfolio Liquidation results. Estimations of the value functions at point $(s_0 = 6, y_0 = 1)$ and time 0 provided by different algorithms.

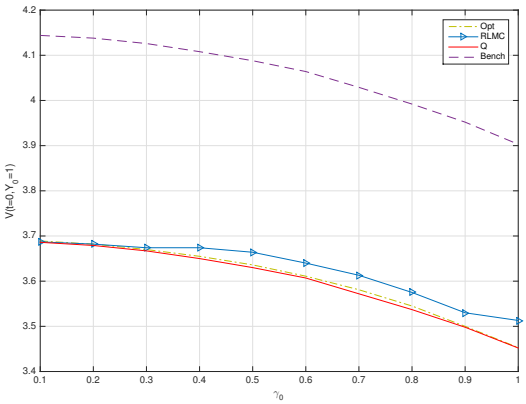
	$b_0 = 0.1$, $T = 1$					$b_0 = -0.1$, $T = 1/2$				
γ_0	Opt	RLMC	CR	Q	Bench	Opt	RLMC	CR	Q	Bench
0.1	-1.347	-1.356	-1.278	-1.368	-1.318	3.689	3.687	3.995	3.686	4.144
0.2	-1.385	-1.390	-1.283	-1.401	-1.348	3.682	3.682	3.847	3.679	4.138
0.3	-1.445	-1.446	-1.314	-1.460	-1.402	3.670	3.674	4.034	3.667	4.126
0.4	-1.523	-1.524	-1.323	-1.556	-1.485	3.655	3.674	4.128	3.650	4.108
0.5	-1.642	-1.637	-1.348	-1.673	-1.585	3.636	3.664	4.243	3.630	4.088
0.6	-1.783	-1.777	-1.425	-1.826	-1.711	3.611	3.640	4.386	3.607	4.064
0.7	-1.973	-1.927	-1.513	-2.018	-1.870	3.581	3.613	4.783	3.572	4.029
0.8	-2.213	-2.003	-1.637	-2.243	-2.057	3.545	3.575	5.142	3.537	3.992
0.9	-2.526	-2.457	-1.819	-2.516	-2.288	3.500	3.530	5.345	3.498	3.952
1	-2.918	-2.801	-1.806	-2.829	-2.560	3.453	3.513	6.765	3.452	3.903

This approximation is a slight generalization (to dimension d=2) of the semi-linear approximation developed in (3.20). Its main interest lies in the continuity of the approximation w.r.t. the control variable a, which provides stability and accuracy to the usual (gradient descent-based) algorithms for the optimal controls searching, as can be seen on the numerical results (see, e.g., Table 1).

2. Portfolio Selection. Consider the Portfolio Selection problem with one risky asset. We choose a CARA utility function $U(x) = -\exp(-px)$, with $p > 0$. It has been shown in [5, Corollary 1] that the optimal portfolio



$b_0 = 0.1$ and $T = 1$.



$b_0 = -0.1$ and $T = 0.5$.

FIGURE 2. Results for the Portfolio Liquidation problem. Estimation of the value function at point $(s_0 = 6, y_0 = 1)$ at time 0 provided by different strategies w.r.t. γ_0. We took γ=5, S_0=6, Y_0=1, η=100 and σ=0.4.

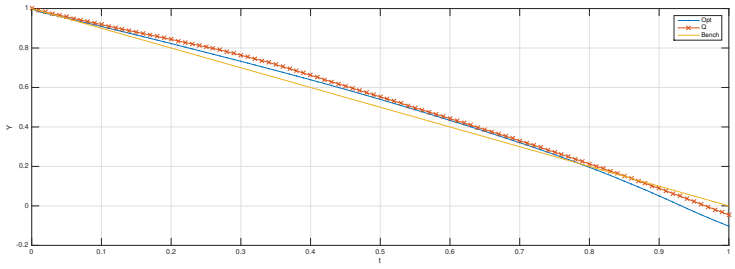


FIGURE 3. Simulation of $(Y_t)_{t\in[0,T]}$ using the (continuous-time) optimal strategy (Opt), the (Q) estimated one, and the Benchmark strategy (Bench) to solve the Portfolio Liquidation problem. We took $T = 1$, $\sigma = 0.4$, $\gamma_0 = 1$, $b_0 = 0.1$, $S_0 = 6$, $Y_0 = 1$, $N = 100$, $\gamma = 5$, $\eta = 100$.

strategy is explicitly given by

$$\alpha_t^* \quad = \quad \frac{\sigma^2 + \gamma_0^2 t}{\sigma^2 + \gamma_0^2 T} \frac{\hat{\beta}_t}{p\sigma^2}$$

where

$$\hat{\beta}_t \quad = \quad \mathbb{E}^0[\beta|\mathcal{F}_t^S] \quad = \quad \frac{\sigma^2}{\sigma^2 + \gamma_0^2 t} b_0 + \frac{\gamma_0^2}{\sigma^2 + \gamma_0^2 t} \Big(\ln \frac{S_t}{S_0} + \frac{1}{2}\sigma^2 t \Big),$$

is the posterior mean of the drift (Bayesian learning on the drift), and the optimal performance by

$$J_2(\alpha^*) \quad = \quad -\exp\Big[-p\Big(x_0+\frac{1}{2p}\big(\ln\big(\frac{\sigma^2+\gamma_0^2T}{\sigma^2}\big)-\frac{\gamma_0^2T}{\sigma^2+\gamma_0^2T}\big)+\frac{b_0^2}{2p\sigma^2}\frac{\sigma^2T}{\sigma^2+\gamma_0^2T}\Big)\Big].$$

The Portfolio Selection problem, even though in many aspects similar to the Portfolio Liquidation problem, is interesting in its own right because the control acts only on the variance of the controlled wealth process. We tested the Regress-Later Monte Carlo (RLMC), the Control Randomization (CR) and the Quantization (Q) algorithm on the Portfolio Selection problem. Similarly to what has been done for Portfolio Liquidation problem, we discretized time choosing $N = 100$ and solved the discrete time problem associated. We considered two set of experiments, $b_0 = 0.1$, $T = 1$ and $b_0 = -0.1$, $T = 0.5$, for different values of $\gamma_0 \in [0, 1]$, $p = 1$, $\sigma = 0.4$. Given all these different parameters, we compared the performance of these algorithms with the one of the optimal strategy for the continuous-time problem α^* (Opt). The general test consists in computing a forward Monte Carlo with 500000 samples, following optimal strategy estimated using different strategies, to provide estimates of $V(t_0 = 0, X_0 = 0, W_0 = 0)$ the value function at time 0.

We present the results of our numerical experiments in Table 2. One can see that the Quantization algorithm performs similarly to the theoretical optimal strategy (Opt) for the continuous time problem, which can be interpreted as stability and accuracy of the Q algorithm, and also shows that the time discretization error is almost zero here.

We also present in Figure 4 a sample of the wealth of the agent following the optimal strategy and the (Q) estimated one. One can see that the strategies slightly differ when the drift is high, and remain the same when the drift is low. The small difference can be explained by the fact that the optimal strategy (Opt) is not optimal for the discrete time version of the problem.

Details on the Q algorithm implementation
We designed the same Quantization algorithm as the one built to solve the Portfolio Liquidation problem. We nevertheless had to take a larger number of points in the grids to minimize the back-propagation of errors from the borders of the girds; and had to use the "explore first, exploit later" idea (see Subsection 3.3.2) to improve the results.

Details on the RL and CR algorithms implementation
When implementing Regression Monte Carlo algorithms, and choosing basis functions, the control on variance implies that low order polynomial can not be used alone, as they can easily cause the control to be bang-bang between the boundaries of its domain. Similarly, piecewise approximations are not very effective, as the dependence on the control is very weak, requiring a high number of local supports and making the computational complexity overwhelming. We tested both value and performance iteration and tried to employ different kinds of basis functions and training points. Unfortunately, both Regress-Later and Control Randomization do not cope well with controlling the dynamics of a process through the variance only. A tailor-made implementation of Regression Monte Carlo to deal with this kind of problems is outside the scope of this paper and further investigation will follow in future work. For now, we chose not to provide results based on RL and CR methods.

TABLE 2. Portfolio Selection results. Estimations of the value function at point $(x_0 = 0, S_0 = 6)$ time 0 using the *continuous-time* optimal strategy (Opt) and (Q) estimated optimal strategy.

	$b_0 = 0.1$, $T = 1$		$b_0 = -0.1$, $T = 0.5$	
γ_0	Opt	Q	Opt	Q
0.1	-0.985	-0.985	-0.992	-0.992
0.2	-0.982	-0.982	-0.991	-0.991
0.3	-0.973	-0.973	-0.988	-0.988
0.4	-0.954	-0.953	-0.981	-0.981
0.5	-0.927	-0.927	-0.969	-0.969
0.6	-0.896	-0.896	-0.952	-0.952
0.7	-0.863	-0.863	-0.932	-0.932
0.8	-0.830	-0.830	-0.910	-0.910
0.9	-0.797	-0.797	-0.886	-0.886
1	-0.767	-0.766	-0.863	-0.863

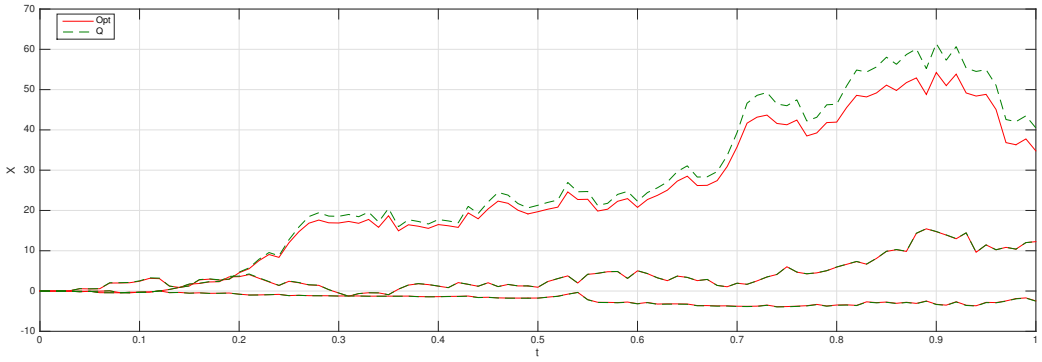


FIGURE 4. 3 simulations of the agent's wealth $(X_t)_{t\in[0,T]}$ when the latter follows the continuous-time optimal strategy (Opt) and the (Q) estimated optimal strategy to solve the Portfolio Selection problem. We took σ=0.4, T=1, P=0.1, γ_0=5, b_0=0,1. One can see that the two strategies are the same when the drift is low; but Q performs slightly better than Opt when the drift is high, which is a time-discretization effect.

4.2. **A model of interbank systemic risk with partial observation**

4.2.1. *The model*

We consider the following model of systemic risk inspired by the model in [8]. The log-monetary reserves of N banks lending to and borrowing from each other are governed by the system

$$dX_t^i = \frac{\kappa}{N}\sum_{j=1}^{N}(X_t^j - X_t^i)dt + \sigma X_t^i(\sqrt{1-\rho^2}dW_t^i + \rho dW_t^0), \quad i = 1,\dots,N$$

where W^i, $i = 1,\dots,N$, are independent Brownian motions, representing the idiosyncratic risk of each bank, W^0 is a common noise independent of W^i, $\sigma > 0$ is given real parameter, $\rho \in [-1, 1]$, and where X_0^i, $i = 1, ..., N$ are i.i.d.. The mean-reversion coefficient $\kappa > 0$ models the strength of interaction between the banks where bank i can lend to and borrow from bank j with an amount proportional to the difference between their reserves. In

the asymptotic regime when $N \to \infty$, the theory of propagation of chaos implies that the reserve state X^i of individual banks become independent and identically distributed conditionally on the common noise W^0, with a state governed by

$$dX_t = \kappa(\mathbb{E}[X_t|W^0] - X_t)dt + \sigma X_t(\sqrt{1-\rho^2}dB_t + \rho dW_t^0)$$

for some Brownian motion B independent of W^0.

Let us now consider a central bank, viewed as a social planner, who only observes the common noise and not the reserves of each bank, and can influence the strength of the interaction between the individual banks, through an $\mathbb{F}^0$-adapted control process α_t. The reserve of the representative bank in the asymptotic regime is then driven by

$$dX_t = (\kappa+\alpha_t)(\mathbb{E}[X_t|W^0] - X_t)dt + \sigma X_t(\sqrt{1-\rho^2}dB_t + \rho dW_t^0), \quad X_0 \sim X_0^1,$$

and we consider that the objective of the central bank is to minimize

$$J(\alpha) = \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\alpha_t^2 + \frac{\eta}{2}(X_t - \mathbb{E}[X_t|W^0])^2\Big)dt + \frac{c}{2}(X_T - \mathbb{E}[X_T|W^0])^2\Big],$$

where $\eta > 0$ and $c > 0$ penalize the departure of the reserve from the average. This is a MKV control problem under partial observation, but notice that it does not belong to the class of linear quadratic (LQ) MKV problems due to the control α which appears in a multiplicative form with the state. However, it fits into our class of polynomial MKV problem, and can be embedded into standard control problem as follows: We set $\bar{X}_t = \mathbb{E}[X_t|W^0]$ and $Y_t = \mathbb{E}[(X_t - \bar{X}_t)^2|W^0]$. The cost functional is then written as

$$J(\alpha) = \mathbb{E}\Big[\int_0^T \Big(\frac{1}{2}\alpha_t^2 + \frac{\eta}{2}Y_t\Big)dt + \frac{c}{2}Y_T\Big]$$

where the dynamics of $\bar{X}$ and Y are governed by

$$\begin{aligned} d\bar{X}_t &= \sigma\rho\bar{X}_t dW_t^0, \quad \bar{X}_0 = x_0 = \mathbb{E}[X_0], \\ dY_t &= \big[\big(\sigma^2 - 2(\kappa+\alpha_t)\big)Y_t + \sigma^2(1-\rho^2)\bar{X}_t^2\big]dt + 2\rho\sigma Y_t dW_t^0, \quad Y_0 = \mathrm{Var}(X_0). \end{aligned}$$

We have then reduced the problem to a $(\mathbb{P}, \mathbb{F}^0)$-control problem, with state variables $(\bar{X}, Y)$ of dimension two, which is not LQ but can be solved numerically.

4.2.2. *Numerical results*

For this problem, in the absence of analytical solution, we decided to compare the estimations of the value function at time 0 provided by our algorithms with a numerical approximation based on finite difference scheme provided by Mathematica, of the solution to the 2-dimensional HJB equation associated to the systemic risk problem:

$$\begin{cases} \partial_t V + \frac{\eta}{2}y + \Big(\big(\sigma^2 - 2\kappa\big)y + \sigma^2(1-\rho^2)x^2\Big)\partial_y V + \sup_{a\in A}\Big[\frac{1}{2}a^2 - 2ay\partial_y V\Big] \\ \qquad\qquad + \frac{\sigma^2\rho^2x^2}{2}\partial_{xx}^2 V + 2\sigma^2\rho^2 xy\partial_{xy}V + 2\sigma^2\rho^2y^2\partial_{yy}^2 V = 0, \quad \text{for } (t,x,y) \in [0,T)\times\mathbb{R}\times\mathbb{R}_+, \\ V(T,x,y) = \frac{c}{2}y, \quad \forall (x,y) \in \mathbb{R}\times\mathbb{R}_+. \end{cases} \tag{4.7}$$

We refer to the solution of this partial differential equation (obtained using Mathematica using finite differences as explained below) as the Benchmark (or simply Bench) in the sequel.

We computed $\hat{\hat{V}}_{\Delta t}(t_0 = 0, x_0 = 10, y_0 = 0)$ using RL, CR and Q methods by considering a sample of size 500 000, and using the following parameters $T = 1$, $\sigma = 0.1$, $\kappa = 0.5$ and $X_0 = 10$. We recall that $\hat{\hat{V}}_{\Delta t}(t_0 = 0, x_0, y_0)$ is an estimation of $V(0, x_0 = 10, y_0 = 0)$, the value function at (x_0, y_0) and time 0 (see its definition on the last step of each pseudo-code presented in Subsection 3.6).

In Table 3 we display the numerical results of experiments run for two situations: we took $\eta = 10$, $c = 100$ and $\eta = 100$, $\rho = 0.5$ and vary the value of ρ in the first case, and vary the value of c in the second one. Plots of the two tables are also available in Figure 5. One can observe that the algorithms performs well. Mainly, Bench and Q provide slightly better results than the Regression Monte Carlo-based algorithms (the curves of Bench and Q are below those of the other two).

Figure 6 shows two examples of paths $(X_t)_{t\in[0,T]}$ controlled by RLMC (curve "RLMC"), $(X_t)_{t\in[0,T]}$ naively controlled by $\alpha = 0$ (curve "uncontrolled"), and the conditional expectation of X $(\bar{X}_t)_{t\in[0,T]}$ (curve "$E(X|W)$"). One can see in these two examples that the (RLMC estimated) optimal control is as follows:

- do nothing when the terminal time is far, i.e., take $\alpha = 0$, not to pay any running cost.
- catch $\bar{X}$ when the terminal time is getting close, to minimize the terminal cost.

We finally present a sample of paths $(Y_t)_{t\in[0,T]}$ controlled by the decisions given by Q in Figure 7. One can see that the (Q estimated) optimal strategy minimizes the running cost first by letting Y grow; and deals with the terminal cost later by making Y small when the terminal time is approaching.

Details on the RL and CR algorithms implementation

For the implementation of the RL algorithm we decided to use polynomial basis functions up to degree 2. This choice allows us to compute the optimal control analytically as a function of the regression coefficients (see Remark 3.9). Compared to other optimization techniques, explicit expression allows for much faster and error-free computations (see Remark 3.9). For CR, we used basis functions up to degree 3 in all dimensions to obtain more stable results.

Regarding the choice of the training measure in RL, we employed marginal normal distributions on each dimension. As we know that the inventory dimension Y represents the conditional variance of the original process X, we centered the training distribution μ_n at zero but considered only training points $Y_n^m \geqslant 0$. In CR, on the other hand, we need to carefully choose the distribution of the random control so that the process Y does not become negative. Notice in fact that the Euler approximation, contrary to the original SDE describing Y, does not remain positive and we would therefore need to carefully choose a control to avoid driving Y negative. In order to achieve such goal, without having to worry too much about the control, we modified the Euler approximation of (4.7) to feature a reflexive boundary at zero. Such features allow to train the estimated control policy to not overshoot when trying to drive the process Y to zero, without having Y to become negative.

Details on the Q algorithm implementation

As stated above, it is straightforward that $Y > 0$ on $(0, T]$. However, the Euler scheme used to approximate the dynamics of Y does not prevent the associated process $(Y_{t_i})_{0<i\leqslant N}$ to be non-positive. When implementing the Q algorithm for the systemic risk problem, we forced $\left(\text{Proj}_{\Gamma_i^Y}\left(Y_{t_i}\right)\right)_{0<i\leqslant N}$ to remain positive by simply choosing positive points for the grids Γ_i^Y that quantize the states of Y_{t_i}, at time t_i for $i = 0, ..., N$.

Also, given the expression of the instantaneous and terminal reward, one can expect Y to stay close to 0, but we do not have any idea of how small Y should stay for the strategy to be optimal (cf. Figure 7 to see a posteriori where Y lies). To deal with this situation, we decided to adopt the "explore first, exploit later" procedure. First, we chose some random grids with a lot of points near 0 and computed the optimal strategy on these

ρ	RLMC	CR	Q	Bench
0.1	8.88	9.12	8.76	8.94
0.2	8.73	8.98	8.69	8.77
0.3	8.42	8.69	8.32	8.48
0.4	8.02	8.25	7.91	8.06
0.5	7.61	7.73	7.37	7.51
0.6	6.93	6.97	6.68	6.79
0.7	5.94	6.07	5.78	5.87
0.8	4.86	4.82	4.62	4.67
0.9	3.32	3.10	3.02	2.97

$c = 100$ and $\eta = 10$.

c	RLMC	CR	Q	Bench
0	7.79	7.78	7.77	7.79
1	7.88	7.87	7.86	7.88
5	8.22	8.23	8.21	8.23
10	8.63	8.64	8.61	8.62
25	9.69	9.76	9.61	9.62
50	11.08	11.27	10.94	10.97

$\rho = 0.5$ and $\eta = 100$.

TABLE 3. Results for the systemic risk problem. Estimations of the value function at point ($x_0 = 10$) at time 0 provided by different strategies. We took $T = 1$, $N = 100$, $\sigma = 0.1$, $\kappa = 0.5$, $X_0 = 10$.

grids. Then, we ran forward Monte Carlo simulations and generated an empirical distribution of the quantized Y. Second, we build new grids of Quantization for Y by generating new points according to the empirical distribution that we got from in the previous step. Finally, we computed new (hopefully better) estimations of the optimal strategy by running the Q algorithm using the new grids. The Q strategy performed better after applying this step, but not significantly since our first naive guess for the grids (i.e., before bootstrapping) was already good enough.

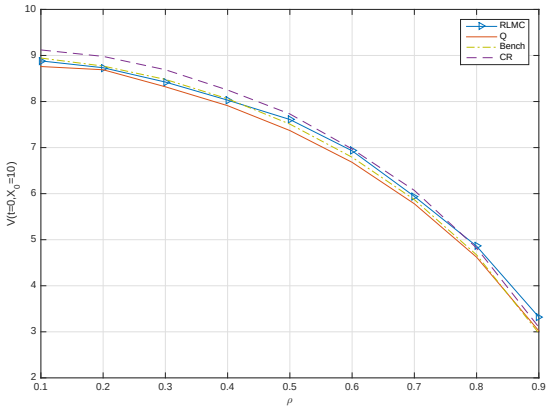
Details on the implementation of the deterministic algorithm for the resolution of the HJB

We use the `NDSolve` function in Mathematica based on finite difference method to solve (4.7). Note that usually terminal and boundary conditions are required to get numerical results. The final condition: $V(T, x, y) = \frac{c}{2}y$ is already given by (4.7). However, the boundary conditions on $V(t, 0, y)$ and $V(t, x, 0)$ are missing, except the trivial condition consisting of $V(t, 0, 0) = 0$. We then provided the HJB without boundary conditions to the Mathematica function `NDSolve`, and let the latter add artificial boundary conditions by itself to output results.

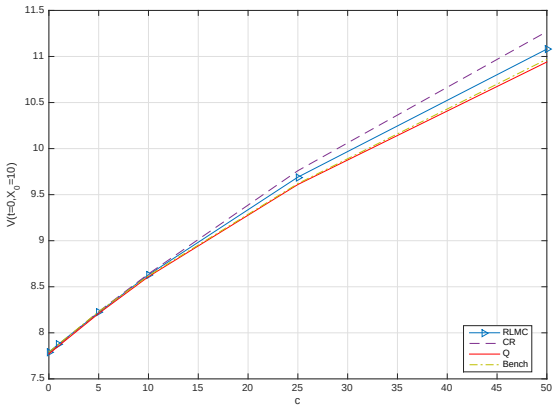
5. Conclusion

In this work, we have investigated how to use probabilistic numerical methods for some classes of mean field control problem via Markovian embedding. We focused on two types of Regression Monte Carlo methods (namely, Regress-Later and Control Randomization) and Quantization. We have then presented three different examples of applications.

We found that the Regression Monte Carlo algorithms perform well in problems of control of the drift. In such problems, they are much faster than Quantization for similar precision. In particular, we noticed that Regress-Later is usually more reliable than Control Randomization; often the choice of a uniform distribution of the training points on an appropriate interval is sufficient to obtain high-quality estimations. On the other hand Control Randomization is very sensitive to the choice of the distribution of the randomized control, and often a few iterations are necessary before finding a good control distribution. We have also tried to use the performance iteration or path recomputation method, but on the examples we considered, it was very time consuming and did not help much in terms of accuracy. Despite the success of Regression Monte Carlo methods in problems with control on the drift, the example of Portfolio Selection highlighted a possible weakness of these algorithms. When the control acts on the variance only, we found difficult to make the numerical scheme converge to sensible results within the computational resources available. We realized that the study of these problems and the solution via Regression Monte Carlo methods is outside the scope of this paper. This is



$c = 100$ and $\eta = 10$.



$\rho = 0.5$ and $\eta = 100$.

FIGURE 5. Results for the systemic risk problem. Estimations of the value function at time 0 using different algorithms w.r.t. ρ and c. We took T=1, N=100, σ=0.1, κ=0.5, x_0=10.

probably related to another limitation of this family of methods: the choice of the basis functions for the regression. Indeed, for some problems, a good basis might be very large or might require several steps of trials and errors.

Quantization-based method, on the other hand, provided very stable and accurate results. A first interesting and practical feature of the Q-algorithm is that regressing the value function using quantization-based methods is local. So, first, it can be easily parallelized to provide fast results, and, second, it is easy to check at which points of the grids the estimations suffer from instability and how to change the grid to fix the problem (basically, by adding more points where the estimations need to be improved). Another interesting feature of the quantization methods is that, one can choose the grids on which to learn the value function. It is possible to exploit this feature in the case where one has, a priori, a rough idea of where the controlled process should be driven by the optimal strategy (see, e.g., the liquidation problem). In this case, one should build grids with many points located where the process is supposed to go. In the case where one has no guess of where the optimal process goes, it is always possible to use bootstrapping methods to build better grids iteratively, starting from a random guess for the grid (see, e.g., the systemic risk problem). In both cases, one has to be particularly careful with the borders of the grids that have been built. Indeed, the decisions computed by quantization-based methods at the borders might easily be wrong if the grids do not have a "good shape" at the borders. Unfortunately, the shape

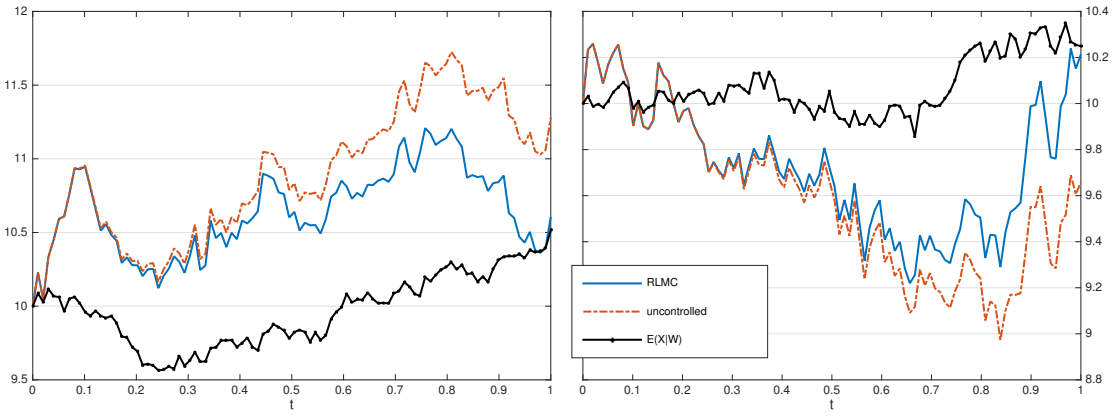


FIGURE 6. Two realizations of $(X_t)_{t\in[0,T]}$ controlled by RLMC (curve "RLMC"), $(X_t)_{t\in[0,T]}$ naively controlled taken $\alpha = 0$ (curve "uncontrolled"), and $\bar{X}$ (curve "$E(X|W)$"). The optimal control for the systemic risk problem (computed by RLMC) is to do nothing at first, and catch $\bar{X}$ when the terminal time is getting close.

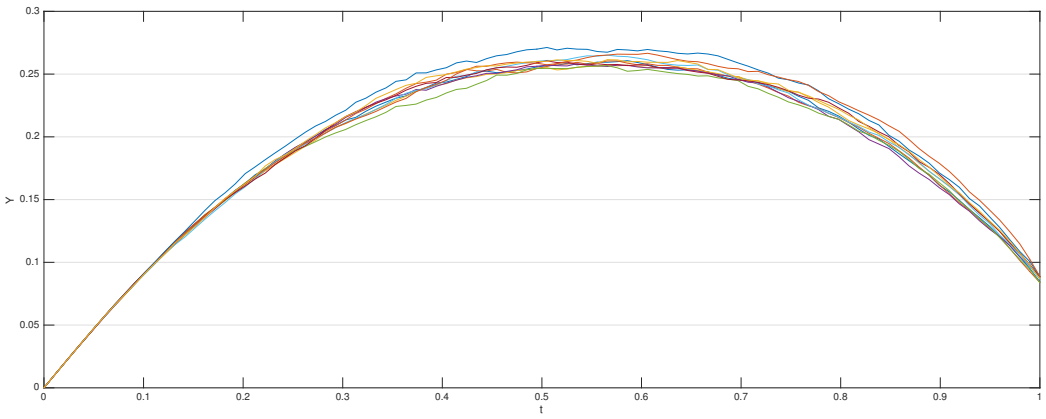


FIGURE 7. Sample of $(Y_t)_{t\in[0,T]}$ controlled by Q. The (Q) estimated optimal control for the systemic risk problem is to initially let Y become large, and then reduce its value when the approaching the terminal time.

of the grid that should be used depends heavily on the problem under consideration. Except in special cases, it seems not possible to avoid the use of deterministic algorithms (such as gradient descent methods or extensive search) to find the optimal action at each point of the grid. A smooth expression of the conditional expectations of the quantized processes is necessary for the deterministic algorithms find optimal strategy efficiently. Once again, the use of parallel computing can alleviate the time-consuming task of searching for the optimal control at each point of the grids.

Acknowledgments. Most of this work was realized while the third author was a postdoctoral fellow at NYU Shanghai; the support of a research discretionary fund from the NYU-ECNU Institute of Mathematical Sciences

and the support provided by the CEMRACS for his stay at CIRM are gratefully acknowledged. The authors's research is part of the ANR project CAESARS (ANR-15-CE05-0024).

References

[1] R. Almgren and N. Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–40, 2001.
[2] A. Balata and J. Palczewski. Regress-Later Monte Carlo for optimal control of Markov processes. *arXiv preprint arXiv:1712.09705*, 2017.
[3] C. Bender, C. Gärtner, and N. Schweizer. Pathwise Dynamic Programming. *Mathematics of Operations Research*, 2018.
[4] C. Bender and J. Steiner. Least-squares Monte Carlo for backward SDEs. In R. Carmona, P. Del Moral, P. Hu, and N. Oudjane, editors, *Numerical methods in finance*, pages 257–289. Springer, 2012.
[5] A. Bismuth, O. Guéant, and J. Pu. Portfolio choice, portfolio liquidation, and portfolio transition under drift uncertainty. *arXiv preprint arXiv:1611.07843*, 2016.
[6] R. Buckdahn, J. Li, S. Peng, and C. Rainer. Mean-field stochastic differential equations and associated PDEs. *The Annals of Probability*, 45(2):824–878, 2017.
[7] P. Cardaliaguet. Notes on mean field games. Technical report, from P.-L. Lions' lectures at Collège de France, 2010.
[8] R. Carmona, J.-P. Fouque, and L.-H. Sun. Mean field games and systemic risk. *Communications in Mathematical Sciences*, 13(4):911–933, 2015.
[9] J.-F. Chassagneux, D. Crisan, and F. Delarue. A probabilistic approach to classical solutions of the master equation for large population equilibria. *arXiv preprint arXiv:1411.3009*, 2014.
[10] L. Fiorin, G. Pagès, and A. Sagna. Product Markovian quantization of a diffusion process with applications to finance. *Methodology and Computing in Applied Probability*, pages 1–32, 2018.
[11] J. Gatheral and A. Schied. Dynamical models of market impact and algorithms for order execution. In *Handbook on Systemic Risk*, pages 579–602. Cambridge University Press, 2013.
[12] P. Glasserman and B. Yu. Simulation for American options: Regression Now or Regression Later? In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 213–226. Springer, 2004.
[13] E. Gobet. *Monte-Carlo methods and stochastic processes: from linear to non-linear*. Chapman and Hall/CRC, 2016.
[14] P. Henry-Labordere. Deep Primal-Dual Algorithm for BSDEs: Applications of Machine Learning to CVA and IM. *SSRN*, 2017.
[15] I. Kharroubi, N. Langrené, and H. Pham. A numerical algorithm for fully nonlinear HJB equations: an approach by control randomization. *Monte Carlo Methods and Applications*, 20(2):145–165, 2014.
[16] I. Kharroubi, N. Langrene, and H. Pham. Discrete time approximation of fully nonlinear HJB equations via BSDEs with nonpositive jumps. *Annals of Applied Probability*, 25:2301–2338, 2015.
[17] P.-L. Lions. Théorie des jeux de champ moyen et applications (mean field games). *Cours du College de France*, 2012.
[18] M. Ludkovski and A. Maheshwari. Simulation methods for stochastic storage problems: A statistical learning perspective. *arXiv:1803.11309*, 2018.
[19] S. Nadarajah, F. Margot, and N. Secomandi. Comparison of least squares Monte Carlo methods with applications to energy real options. *European Journal of Operational Research*, 256(1):196–204, 2017.
[20] G. Pagès, H. Pham, and J. Printems. Optimal quantization methods and applications to numerical problems in finance. In S. T. Rachev, editor, *Handbook of computational and numerical methods in finance*, pages 253–297. Birkhäuser, 2004.
[21] H. Pham. Linear quadratic optimal control of conditional McKean-Vlasov equation with random coefficients and applications. *Probability, Uncertainty and Quantitative Risk*, 1(1):7, 2016.
[22] H. Pham and X. Wei. Dynamic programming for optimal control of stochastic McKean-Vlasov dynamics. *SIAM Journal on Control and Optimization*, 55(2):1069–1101, 2017.
[23] L. C. G. Rogers. Monte Carlo Valuation of American Options. *Mathematical Finance*, 12:271–286, 2002.