Control randomisation approach for policy gradient and application to reinforcement learning in optimal switching

Robert DENKERT[*] Huyên PHAM[†] Xavier WARIN [‡]

December 2, 2024

Abstract. We propose a comprehensive framework for policy gradient methods tailored to continuous time reinforcement learning. This is based on the connection between stochastic control problems and randomised problems, enabling applications across various classes of Markovian continuous time control problems, beyond diffusion models, including e.g. regular, impulse and optimal stopping/switching problems. By utilizing change of measure in the control randomisation technique, we derive a new policy gradient representation for these randomised problems, featuring parametrised intensity policies. We further develop actor-critic algorithms specifically designed to address general Markovian stochastic control issues. Our framework is demonstrated through its application to optimal switching problems, with two numerical case studies in the energy sector focusing on real options.

Key words: Reinforcement learning in continuous time, policy gradient, control randomization, actor-critic algorithms, optimal switching.

[*] Humboldt University of Berlin, Department of Mathematics, denkertr at hu-berlin.de; This author gratefully acknowledges financial support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 410208580 – IRTG2544 ("Stochastic Analysis in Interaction").

[†] Ecole Polytechnique, CMAP, huyen.pham at polytechnique.edu; This author is supported by the BNP-PAR Chair "Futures of Quantitative Finance", and by FiME, Laboratoire de Finance des Marchés de l'Energie, and the "Finance and Sustainable Development" EDF - CACIB Chair

[‡] EDF R&D and FiME, Laboratoire de Finance des Marchés de l'Energie.

1. Introduction

The theory of reinforcement learning for continuous time stochastic control has advanced significantly, beginning with the foundational work [23], and continuing with [16], [15], [11] and [20] who developed policy gradient methods and actor-critic algorithms, and [17] for q-learning. These studies primarily focus on regular controls within (jump-)diffusion processes, employing the Feynman-Kac formula and the partial differential equations (PDE) representation of the value function to derive gradients of the performance value function with respect to the parameters of the stochastic policy.

Our research aims to expand the application of these methods beyond diffusion models to a broader range of Markovian control problems, including singular, impulse, and optimal stopping and switching problems. To achieve this, we propose a unified framework with a general reformulation in terms of Markovian randomised problems. This approach to stochastic control problem is commonly referred to as *control randomisation*, initially introduced in [2] for optimal switching problems, and further developed in [18] for impulse control, in [19] for regular controls, and in [9] for general non-Markovian stochastic control problems. The basic idea is to replace the control process $(\alpha_t)_t$ valued in A by a random (uncontrolled) point process $(I_t)_t$ with marks in A, formulate an auxiliary control problem where the intensity distribution of I is controlled, called randomized problem, and show that the value functions of the two problems coincide. The key feature of the randomised problems is its formulation in terms of a family of dominated probability measures under which the optimization is performed.

Utilizing the change of measure in these randomised settings, we derive a gradient representation of the value function with respect to parametrised intensity policies directly, without reliance on PDEs. This framework not only incorporates Poisson discretisation as per the randomization method but also accommodates standard fixed discretisations for continuous-time problems. Using this policy gradient, we design an Actor-Critic algorithm to alternately learn the value function and the optimal intensity policy. Notably, the gradient structure relies solely on the state at action points, circumventing the need for further discretisation during implementation.

We demonstrate the applicability of our results in a model-free setting, learning optimal control and value functions through empirical observations and samples. This methodology is applied specifically to optimal switching problems but is adaptable to a wide variety of continuous stochastic control scenarios. We provide numerical examples from real options in the energy markets to illustrate these concepts.

Our work is related to some recent papers that solve optimal stopping problems in continuous time with reinforcement learning methods. In [7], the author uses randomized stopping times, and policy iterations for computing the American Put option. This approach has been extended in [6] to general optimal stopping problems reformulated as a singular control problems, and the authors have shown convergence of their policy iteration algorithm. The paper [5] uses penalization methods for the variational inequality associated to optimal stopping problem for designing actor-critic algorithms in the spirit of [16].

The remainder of the paper is structured as follows: In Section 2, we detail the Markovian randomised problem and develop a corresponding policy gradient method. Section 3 applies this methodology to a diverse array of continuous time control problems, and Section 4 presents and evaluates numerical experiments within the context of optimal switching problems. We give concluding remarks in Section 5. In Appendix A, we recall how the randomised control problem is constructed for both regular control and optimal stopping problems.

2. Policy gradient method for Markovian randomised problems

In this section, we will consider a general class of Markovian randomised control problems in continuous time. Control randomisation method can be seen as a unified approach to a large class of control problems in continuous time, including optimal stopping, switching and impulse control problems, as we will see in Section 3. We will derive first a general policy gradient representation and then from this, an Actor-Critic algorithm to tackle this class of problems.

2.1. Theory background

2.1.1. Randomised control problem setup

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space on which we consider a simple random counting measure ν on $(0,\infty) \times A$ with A some Polish space, such that $\mathbb{E}[\nu((0,T] \times A)] < \infty$, and associated to the marked point process $(\tau_n, \mathrm{a}_n)_n$ and the pure jump A-valued process I with dynamics

$$dI_s = \int_A (e - I_{s-})\nu(ds, de), \quad s \leq T. \tag{2.1}$$

Both ν and its intensity are chosen by the reinforcement learning (RL) agent. We denote by $I^{t,a}$ the jump process starting from $a \in A$, at time $t \in [0,T]$, that is $I^{t,a}_t = a$, and following the dynamics (2.1) for $t \leq s \leq T$. In equation (2.1), and in the sequel, the letter e denotes the post-jump state. We consider a state process X valued on $\mathbb{R}^d$ s.t. the pair (X, I) is càdlàg and Markov. We assume that the jumps of X occur either at the times specified by ν (which are endogenously driven by the agent), or at times independent of ν (representing exogenous jumps that are unknown to the agent). We further assume that these exogenous jumps have an intensity absolutely continuous w.r.t. the Lebesgue measure. An example includes the case where X is driven by a SDE in the form

$$\begin{aligned} dX_s = &\ b(s, X_s, I_s)ds + \sigma(s, X_s, I_s)dW_s + \int_U \delta(s, X_{s-}, I_{s-}, u)\mu(ds, du) \\ &+ \int_A \gamma(s, X_{s-}, I_{s-}, e)\nu(ds, de), \end{aligned} \tag{2.2}$$

with W a Brownian motion, and μ a Poisson random measure on $(0,\infty) \times U$ independent of ν with intensity $\rho(s, du)ds$, where U is another Polish space. This setup in particular includes controlled jump-diffusion processes, as studied in [1] and [11]. We denote by $X^{t,x,a}$ the state process X that starts from x at time t, that is $X^{t,x,a}_t = x$, and s.t. $(X^{t,x,a}, I^{t,a})$ is Markov, and we assume the estimate

$$\mathbb{E}\Big[\sup_{t \leq s \leq T} |X^{t,x,a}_s|^p\Big] \leq C(1 + |x|^p), \qquad \forall x \in \mathbb{R}^d,$$

for some positive constant C and $p \in [1,\infty)$. This estimate is satisfied for X as in (2.2) under standard Lipschitz and linear growth conditions on $b, \sigma, \gamma, \delta$.

By [14, Theorems 2.1, 2.3, 3.4], there exists a unique (up to a $\mathbb{P}$-null set) predictable random measure $\hat{\nu}$ with $\hat{\nu}(\{s\} \times A) \leq 1$ for all $s \in (0,T]$ such that for every $\mathcal{P}(\mathbb{F}^{X,\nu}) \otimes \mathcal{B}(A)$-measurable random field $H \geq 0$, where $\mathcal{P}(\mathbb{F}^{X,\nu})$ denotes the predictable σ-algebra of $\mathbb{F}^{X,\nu}$, it holds that

$$\mathbb{E}\left[\int_0^T \int_A H(s,e)\nu(ds,de)\right] = \mathbb{E}\left[\int_0^T \int_A H(s,e)\hat{\nu}(ds,de)\right],$$

called the *predictable projection or compensator of* ν, which is uniquely characterising ν.

Guided by the approach of the randomisation method, we will now optimise over the set of (in a suitable sense) "intensities" of the process I. In the context of regular control problems, this means that instead of directly controlling the control process I, the RL agent optimises the underlying intensity function λ, which governs the frequency and distribution of the jumps of I.

To this end, we note that for every $Pred(\mathbb{F}^{X,\nu})\otimes\mathcal{B}(A)$-measurable, essentially bounded process λ satisfying

(i) $\int_A \lambda_s(e)\hat{\nu}(\{s\},de) \leq 1$ for all $s \in (0,T]$,

(ii) for all $s \in (0,T]$ such that $\hat{\nu}(\{s\}\times A) = 1$, it also holds that $\int_A \lambda_s(e)\hat{\nu}(\{s\},de) = 1$,

we can construct a tilted probability measure $\mathbb{P}^\lambda \ll \mathbb{P}$ such that ν is a random point measure with the predictable projection $\lambda_s(e)\hat{\nu}(ds,de)$ under $\mathbb{P}^\lambda$, thus changing the distribution of the control process I.

This is achieved through Girsanov's theorem, as outlined in e.g. [14, Theorem 4.5], by defining $\mathbb{P}^\lambda$ via its density process

$$\begin{aligned} Z_s^\lambda := \frac{d\mathbb{P}^\lambda}{d\mathbb{P}}\Big|_{\mathcal{F}_s^{X,\nu}} = & \prod_{t\in(0,s],0<\hat{\nu}(\{t\}\times A)<1,\nu(\{t\}\times A)=0} \frac{1-\int_A \lambda_t(e)\hat{\nu}(\{t\},de)}{1-\hat{\nu}(\{t\}\times A)} \\ & \cdot \exp\Big(\int_{(0,s]}\int_A \log\lambda_t(e)\nu(dt,de) - \int_{(0,s]}\int_A (\lambda_t(e)-1)\hat{\nu}^c(dt,de)\Big), \quad (2.3) \end{aligned}$$

for $s \in (0,T]$, where $\hat{\nu}^c(ds,de) := \mathbb{1}_{\{\hat{\nu}(\{s\}\times A)=0\}}\hat{\nu}(ds,de)$.

Notice that we do not assume necessarily that the compensator is absolutely continuous w.r.t. the Lebesgue measure ds, in order to take into account the possibility of jumps at deterministic times, hence to embed the case of stochastic control on discrete time, i.e. Markov decision process.

By the Markovian structure of our problem, we now define the set of admissible control $\mathcal{V}$ as all such processes λ satisfying the above conditions while being of the form

$$\lambda_s(e) = \lambda(e|s, X_{s-}, I_{s-}), \qquad s \leq T,$$

for some bounded deterministic function λ on $A\times[0,T]\times\mathbb{R}^d\times A$. For the ease of arguments, we furthermore require that all $\lambda \in \mathcal{V}$ satisfy the following conditions which ensure that also $\mathbb{P} \ll \mathbb{P}^\lambda$ and thus $\mathbb{P}^\lambda \sim \mathbb{P}$, [1]

(i) λ is bounded away from 0, that is $\inf_{(s,x,a,e)} \lambda(e|s,x,a) > 0$,

(ii) there exists a constant $C < 1$ such that for all $s \in (0,T]$, when $\hat{\nu}(\{s\}\times A) < 1$, then it also holds that $\int_A \lambda(e|s,X_{s-},I_{s-})\hat{\nu}(\{s\},de) \leq C < 1$.

[1] Since every λ with $\mathbb{P}^\lambda \ll \mathbb{P}$ can be approximated by $(\lambda^n)_n \subseteq \mathcal{V}$, this additional assumption also does not change the value function. Similar arguments are standard for randomised control problems.

Note that for each such $\lambda \in \mathcal{V}$, the process (X, I) will still be Markovian under $\mathbb{P}^\lambda \ll \mathbb{P}$, and we have the estimate

$$\mathbb{E}^\lambda \Big[\sup_{t \le s \le T} |X_s^{t,x,a}|^p \Big] \le\ C_\lambda (1 + |x|^p), \qquad \forall x \in \mathbb{R}^d,$$

where $\mathbb{E}^\lambda$ denotes the expectation under $\mathbb{P}^\lambda$. In general, our objective is now to optimise the reward functional:

$$J(t, x, a, \lambda) \coloneqq\ \mathbb{E}^\lambda \Big[g(X_T^{t,x,a}, I_T^{t,a}) + \int_t^T f(s, X_s^{t,x,a}, I_s^{t,a}) ds - \int_{(t,T]} \int_A c(s, X_{s-}^{t,x,a}, I_{s-}^{t,a}, e) \nu(ds, de) \Big], \tag{2.4}$$

for $(t, x, a) \in [0, T] \times \mathbb{R}^d \times A$, where the reward functions f, g and the cost function c are assumed to satisfy the polynomial growth condition

$$|f(t, x, a)| + |g(x, a)| + |c(t, x, a, e)| \le C(1 + |x|^p),$$

for all $t \in [0, T]$, $x \in \mathbb{R}^d$, $a, e \in A$. Notice that the reward functional J then also satisfies the the polynomial growth condition

$$|J(t, x, a, \lambda)| \le C_\lambda (1 + |x|^p).$$

Remark 2.1. *From the definition of the reward functional J and the Markov property of (X, I), we have the martingale property under $\mathbb{P}^\lambda$, $\lambda \in \mathcal{V}$, of the process*

$$J(s, X_s, I_s, \lambda) + \int_0^s f(r, X_r, I_r) dr - \int_{(0,s]} \int_A c(r, X_{r-}, I_{r-}, e) \nu(dr, de), \quad 0 \le s \le T.$$

Remark 2.2. *The difference between $\mathbb{P}^\lambda$ and $\mathbb{P}$ solely lies in the intensity of process ν, which is the (known) part that agent can control. In particular, the (unknown) environment (in our example, the dynamics of the state process is given by* (2.2)*) remains the same under both probability measures. In fact, by combining* (2.3) *and* (2.4)*, we can express the reward functional entirely under the original probability measure $\mathbb{P}$.*

2.1.2. Policy gradient representation

The policy gradient method aims to optimize the expected reward J by exploring a parameterized family $(\lambda^\theta)_{\theta \in \Theta} \subseteq \mathcal{V}$. This family is chosen to be sufficiently dense in $\mathcal{V}$, meaning that

$$\sup_{\lambda \in \mathcal{V}} J(t, x, a, \lambda) = \sup_{\theta \in \Theta} J(t, x, a, \theta),$$

where we denote by $J(t, x, a, \theta) \coloneqq J(t, x, a, \lambda^\theta)$ with a slight abuse of notation. The optimization process then involves computing the gradient of J^θ with respect to the parameter θ, allowing for updates to the policy parameters – typically done through methods like gradient descent – to maximize the overall reward.

Our aim in this section is now to derive an explicit formula for the gradient $\nabla_\theta J^\theta(t, x, a, \theta)$. While the approach by [16] is based on the Feynman-Kac formula for J^θ, we will instead use the

Girsanov formula (2.3). The advantage is that we do not need to assume or impose conditions for ensuring regularity on the functional J for deriving the partial differential equations that it satisfies in the continuous-time framework. This is crucial since the function J may be discontinuous in time in the case where $\hat{\nu}$ admits atoms in time, and then PDE method cannot be applied.

We shall assume that for all $(t,x,a,e) \in [0,T] \times \mathbb{R}^d \times A \times A$, the map $\theta \in \Theta \mapsto \lambda^\theta(e|t,x,a)$ is differentiable with a derivative satisfying the growth condition: for each $\theta \in \Theta$, there exists some positive constant C_θ s.t.

$$\int_{(t,T]} \int_A |\nabla_\theta \lambda^\theta(e|s,x,a)| \hat{\nu}(ds,de) \leq \ C_\theta(1+|x|), \quad (t,x) \in [0,T] \times \mathbb{R}^d.$$

Theorem 2.3. *We have*

$$\begin{aligned}\nabla_\theta J(t,x,a,\theta) = \mathbb{E}^\theta \Big[\int_{(t,T]} \int_A & \nabla_\theta (\log \lambda^\theta)(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) \\ & \cdot \Big(J(s, X_s^{t,x,a}, e, \theta) - J(s, X_{s-}^{t,x,a}, I_{s-}^{t,a}, \theta) - c(s, X_{s-}^{t,x,a}, I_{s-}^{t,a}, e) \Big) \nu(ds,de) \Big],\end{aligned} \tag{2.5}$$

for $(t,x,a) \in [0,T] \times \mathbb{R}^d \times A$, *where* $\mathbb{E}^\theta$ *denotes the expectation under* $\mathbb{P}^\theta = \mathbb{P}^{\lambda^\theta}$.

Proof. From Bayes formula with (2.3), the reward functional is formulated in term of the reference probability measure $\mathbb{P}$ instead of $\mathbb{P}^\theta := \mathbb{P}^{\lambda^\theta}$ as follows for $\theta \in \Theta$,

$$\begin{aligned} & J(t,x,a,\theta) \\ = \ & \mathbb{E}^\theta \Big[g(X_T^{t,x,a}, I_T^{t,a}) + \int_t^T f(s, X_s^{t,x,a}, I_s^{t,a}) ds - \int_{(t,T]} \int_A c(s, X_{s-}^{t,x,a}, I_{s-}^{t,a}, e) \nu(ds,de) \Big] \\ = \ & \mathbb{E} \Big[Z_T^{t,x,a,\theta} \Big(g(X_T^{t,x,a}, I_T^{t,a}) + \int_t^T f(s, X_s^{t,x,a}, I_s^{t,a}) ds - \int_{(t,T]} \int_A c(s, X_{s-}^{t,x,a}, I_{s-}^{t,a}, e) \nu(ds,de) \Big) \Big], \end{aligned}$$

where

$$\begin{aligned} Z_T^{t,x,a,\theta} = \ \exp \Big(& \int_{(t,T]} \int_A \log \lambda^\theta(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) \nu(ds,de) - \int_{(t,T]} \int_A (\lambda^\theta(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) - 1) \hat{\nu}^c(ds,de) \\ & + \sum_{s \in (t,T], 0 < \hat{\nu}(\{s\} \times A) < 1} \mathbb{1}_{\{\nu(\{s\} \times A) = 0\}} \log \Big(\frac{1 - \int_A \lambda^\theta(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) \hat{\nu}(\{s\}, de)}{1 - \hat{\nu}(\{s\} \times A)} \Big) \Big). \end{aligned}$$

By differentiating this relation w.r.t. θ, and writing $\nabla_\theta Z_T^{t,x,a,\theta} = Z_T^{t,x,a,\theta} L_T^{t,x,a,\theta}$ with

$$\begin{aligned} L_T^{t,x,a,\theta} = \ & \int_{(t,T]} \int_A \nabla_\theta (\log \lambda^\theta)(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) \nu(ds,de) - \int_{(t,T]} \int_A \nabla_\theta \lambda^\theta(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) \hat{\nu}^c(ds,de) \\ & - \sum_{s \in (t,T], 0 < \hat{\nu}(\{s\} \times A) < 1} \mathbb{1}_{\{\nu(\{s\} \times A) = 0\}} \frac{\int_A \nabla_\theta \lambda^\theta(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) \hat{\nu}(\{s\}, de)}{1 - \int_A \lambda^\theta(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) \hat{\nu}(\{s\}, de)}, \end{aligned}$$

we get

$$
\begin{aligned}
&\nabla_\theta J(t,x,a,\theta)\\
=&\,\mathbb{E}^\theta\Big[L_T^{t,x,a,\theta}\Big(g(X_T^{t,x,a},I_T^{t,a})+\int_t^T f(s,X_s^{t,x,a},I_s^{t,a})ds-\int_{(t,T]}\int_A c(s,X_{s-}^{t,x,a},I_{s-}^{t,a},e)\nu(ds,de)\Big)\Big]
\end{aligned}
$$

To simplify this expression, we will use that due the Markovian structure of our problem, the process $M^{t,x,a,\theta}$ given by

$$
M_s^{t,x,a,\theta}:=J(s,X_s^{t,x,a},I_s^{t,a},\theta)+\int_t^s f(r,X_r^{t,x,a},I_r^{t,a})dr-\int_{(t,s]}\int_A c(r,X_{r-}^{t,x,a},I_{r-}^{t,a},e)\nu(dr,de),\quad s\in[t,T],
$$

is a $\mathbb{P}^\theta$-martingale, see also Remark 2.1. We start by noting that $J(T,X_T^{t,x,a},I_T^{t,a},\theta)=g(X_T^{t,x,a},I_T^{t,a})$, which allows us to write $\nabla_\theta J(t,x,a,\theta)$ using $M^{t,x,a,\theta}$ as follows

$$
\begin{aligned}
\nabla_\theta J(t,x,a,\theta)&=\mathbb{E}^\theta[L_T^{t,x,a,\theta}M_T^{t,x,a,\theta}]\\
&=\mathbb{E}^\theta\Big[\int_{(t,T]}\int_A\nabla_\theta(\log\lambda^\theta)(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})M_T^{t,x,a,\theta}\nu(ds,de)\\
&\quad-\int_{(t,T]}\int_A\nabla_\theta\lambda^\theta(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})M_T^{t,x,a,\theta}\hat{\nu}^c(ds,de)\\
&\quad-\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}\mathbb{1}_{\{\nu(\{s\}\times A)=0\}}\frac{\int_A\nabla_\theta\lambda^\theta(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})\hat{\nu}(\{s\},de)}{1-\int_A\lambda^\theta(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})\hat{\nu}(\{s\},de)}M_T^{t,x,a,\theta}\Big].
\end{aligned}
$$

Now using the $\mathbb{P}^\theta$-martingale property of $M^{t,x,a,\theta}$, we obtain

$$
\begin{aligned}
\nabla_\theta J(t,x,a,\theta)&=\mathbb{E}^\theta\Big[\int_{(t,T]}\int_A\nabla_\theta(\log\lambda^\theta)(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})M_s^{t,x,a,\theta}\nu(ds,de)\\
&\quad-\int_{(t,T]}\int_A\nabla_\theta\lambda^\theta(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})M_s^{t,x,a,\theta}\hat{\nu}^c(ds,de)\\
&\quad-\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}\mathbb{1}_{\{\nu(\{s\}\times A)=0\}}\frac{\int_A\nabla_\theta\lambda^\theta(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})\hat{\nu}(\{s\},de)}{1-\int_A\lambda^\theta(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})\hat{\nu}(\{s\},de)}M_s^{t,x,a,\theta}\Big].
\end{aligned}
\tag{2.6}
$$

To simplify the notation in the following arguments, let us introduce the predictable process

$$
N_s^{t,x,a,\theta}:=J(s,X_{s-}^{t,x,a},I_{s-}^{t,a},\theta)+\int_t^s f(r,X_r^{t,x,a},I_r^{t,a})dr-\int_{(t,s)}\int_A c(r,X_{r-}^{t,x,a},I_{r-}^{t,a},e)\nu(dr,de),\quad s\in[t,T].
$$

We note that since $X^{t,x,a}$ and $I^{t,a}$ are both càdlàg, they have $\mathbb{P}$-a.s. only countably many discontinuities on $[t,T]$. Since ν has $\mathbb{P}$-a.s. only finitely many events on $(t,T]\times A$, this implies that $\{s\in[t,T]|M_s^{t,x,a,\theta}\neq N_s^{t,x,a,\theta}\}$ is $\mathbb{P}$-a.s. countable and thus $M_s^{t,x,a,\theta}=N_s^{t,x,a,\theta}$, $\mathbb{P}\otimes\hat{\nu}^c(\cdot,A)$-a.s., which allows us to rewrite the second term in (2.6) as

$$
\begin{aligned}
&\mathbb{E}^\theta\Big[\int_{(t,T]}\int_A\nabla_\theta\lambda^\theta(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})M_s^{t,x,a,\theta}\hat{\nu}^c(ds,de)\Big]\\
&=\mathbb{E}^\theta\Big[\int_{(t,T]}\int_A\nabla_\theta\lambda^\theta(e|s,X_{s-}^{t,x,a},I_{s-}^{t,a})N_s^{t,x,a,\theta}\hat{\nu}^c(ds,de)\Big].
\end{aligned}
\tag{2.7}
$$

For the last term in (2.6), we focus on the not-almost-sure jumps of ν. We first note that $\nu(\{s\}\times A)=0$ implies that $I^{t,a}_{s-}=I^{t,a}_s$. Additionally, since by assumption the jumps of X occur either at the times specified by ν, or at times independent of ν (with an compensator absolutely continuous with respect to the Lebesgue measure), we have

$$\mathbb{E}^\theta\Big[\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}\mathbb{1}_{\{\nu(\{s\}\times A)=0\}}\mathbb{1}_{\{X^{t,x,a}_{s-}\neq X^{t,x,a}_s\}}\Big]=0.$$

Finally, since $I^{t,a}_{s-}=I^{t,a}_s$ and $X^{t,x,a}_{s-}=X^{t,x,a,}_s$ together imply that also $M^{t,x,a,\theta}_s=N^{t,x,a,\theta}_s$, we obtain

$$\begin{aligned}
&\mathbb{E}^\theta\Big[\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}\mathbb{1}_{\{\nu(\{s\}\times A)=0\}}\frac{\int_A\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}{1-\int_A\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}M^{t,x,a,\theta}_s\Big]\\
&=\mathbb{E}^\theta\Big[\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}(1-\mathbb{1}_{\{\nu(\{s\}\times A)>0\}})\frac{\int_A\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}{1-\int_A\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}N^{t,x,a,\theta}_s\Big].
\end{aligned}\tag{2.8}$$

To simply this term, we note that since ν is a simple random counting measure

$$\begin{aligned}
&\mathbb{E}^\theta\Big[\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}\mathbb{1}_{\{\nu(\{s\}\times A)>0\}}\frac{\int_A\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}{1-\int_A\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}N^{t,x,a,\theta}_s\Big]\\
&=\mathbb{E}^\theta\Big[\int_{(t,T]}\int_A\mathbb{1}_{\{0<\hat{\nu}(\{s\}\times A)<1\}}\frac{\int_A\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}{1-\int_A\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}N^{t,x,a,\theta}_s\nu(\{s\},du)\Big]\\
&=\mathbb{E}^\theta\Big[\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}\frac{\int_A\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}{1-\int_A\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}N^{t,x,a,\theta}_s\int_A\lambda^\theta(u|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},du)\Big],
\end{aligned}$$

using that $N^{t,x,a,\theta}$ is by construction predictable. Therefore, we can rewrite (2.8) as

$$\begin{aligned}
&\mathbb{E}^\theta\Big[\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}\mathbb{1}_{\{\nu(\{s\}\times A)=0\}}\frac{\int_A\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}{1-\int_A\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)}M^{t,x,a,\theta}_s\Big]\\
&=\mathbb{E}^\theta\Big[\sum_{s\in(t,T],0<\hat{\nu}(\{s\}\times A)<1}\int_A\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})N^{t,x,a,\theta}_s\hat{\nu}(\{s\},de)\Big].
\end{aligned}\tag{2.9}$$

To continue, we need an auxiliary result, for which we will take a closer look at the times where $\hat{\nu}(\{s\}\times A)=1$. Since λ^θ is an admissible control, this implies for such time points that $\int_A\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)=1$ for all $\theta\in\Theta$ and thus

$$\int_A\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)=\nabla_\theta\Big(\int_A\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)\Big)=0.$$

This leads us to

$$\mathbb{E}^\theta\Big[\sum_{s\in(t,T],\hat{\nu}(\{s\}\times A)=1}\int_A N^{t,x,a,\theta}_s\nabla_\theta\lambda^\theta(e|s,X^{t,x,a}_{s-},I^{t,a}_{s-})\hat{\nu}(\{s\},de)\Big]=0.\tag{2.10}$$

Thus, putting (2.7), (2.9) and (2.10) together, we obtain that

$$\begin{aligned}
&\mathbb{E}^\theta \Bigg[\int_{(t,T]} \int_A \nabla_\theta \lambda^\theta (e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) M^{t,x,a,\theta}_s \hat{\nu}^c(ds, de) \\
&\qquad + \sum_{s \in (t,T], 0 < \hat{\nu}(\{s\} \times A) < 1} \mathbb{1}_{\{\nu(\{s\}\times A)=0\}} \frac{\int_A \nabla_\theta \lambda^\theta (e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) \hat{\nu}(\{s\}, de)}{1 - \int_A \lambda^\theta (e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) \hat{\nu}(\{s\}, de)} M^{t,x,a,\theta}_s \Bigg] \\
&= \mathbb{E}^\theta \Bigg[\int_{(t,T]} \int_A \nabla_\theta \lambda^\theta (e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) N^{t,x,a,\theta}_s \hat{\nu}(ds, de) \Bigg].
\end{aligned}$$

Now using that the integrand $N^{t,x,a,\theta}$ is predictable, we can apply that $\lambda^\theta(e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) \hat{\nu}(ds, de)$ is the predictable projection of ν under $\mathbb{P}^\theta$, to obtain

$$\begin{aligned}
&\mathbb{E}^\theta \Bigg[\int_{(t,T]} \int_A \nabla_\theta \lambda^\theta (e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) N^{t,x,a,\theta}_s \hat{\nu}(ds, de) \Bigg] \\
&= \mathbb{E}^\theta \Bigg[\int_{(t,T]} \int_A \nabla_\theta (\log \lambda^\theta) (e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) N^{t,x,a,\theta}_s \nu(ds, de) \Bigg].
\end{aligned}$$

Finally, together with (2.6), we obtain

$$\begin{aligned}
\nabla_\theta J(t,x,a,\theta) &= \mathbb{E}^\theta \Bigg[\int_{(t,T]} \int_A \nabla_\theta (\log \lambda^\theta) (e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) \cdot (M^{t,x,a,\theta}_s - N^{t,x,a,\theta}_s) \nu(ds, de) \Bigg] \\
&= \mathbb{E}^\theta \Bigg[\int_{(t,T]} \int_A \nabla_\theta (\log \lambda^\theta) (e|s, X^{t,x,a}_{s-}, I^{t,a}_{s-}) \\
&\qquad \cdot \Big(J(s, X^{t,x,a}_s, e, \theta) - J(s, X^{t,x,a}_{s-}, I^{t,a}_{s-}, \theta) - c(s, X^{t,x,a}_{s-}, I^{t,a}_{s-}, e) \Big) \nu(ds, de) \Bigg].
\end{aligned}$$

□

We will see that we can use this to construct policy gradient (PG) steps for a diverse class of control problems in continuous time.

Remark 2.4. *The policy gradient formula* (2.5) *only relies on the state at the action points where the actor changes its control. This is due to randomised controls being by construction piece-wise constant, leading to a policy gradient that is naturally discretised along the jump times of the control. In contrast to the direct approaches in [16, 17, 11, 1] on the non-randomised control problem, our approach using the randomised control problem avoids the need for further approximation of the policy gradient.*

Remark 2.5. *The policy gradient formula in this paper differs from [16, 17, 11] even for the fixed grid discretisation $\nu(ds, A) = \sum_{k=0}^N \delta_{\frac{kT}{N}}(ds)$. In [16, Algorithm 1], the policy gradient algorithm involves a discrete-time setup based on the intervals $\tau_k - \tau_{k-1}$. In contrast, our formula is designed around the jump points of the randomised control, focusing on $\tau_k - \tau_{k-}$, so the times right before and after the actor changes its control. Thus the continuous running reward does not explicitly appear in our formula, and only the immediate jump costs c are considered. However, the continuous running reward is implicitly captured through the value functional J in our setup.*

2.2. Actor-critic algorithm

We now aim to design an actor-critic (AC) learning algorithm for our randomised problem. AC algorithms are useful to tackle problems in environments where explicit knowledge of system dynamics is unavailable (e.g. model-free settings) and they consist out of two steps which are executed in turns: the policy evaluation (PE) step updates our reward functional estimate J based on the current policy λ, and the policy gradient (PG) step updates our current policy λ using the current estimate of J. This enables simultaneous learning of the optimal parameters κ and θ for our parametrised families $(J^\kappa)_\kappa$ representing the reward functional and $(\lambda^\theta)_\theta$ for the optimal intensity control. In particular, we expect J^κ to approximate the true value function for our control problem.

We will base the policy gradient (PG) step, on the representation (2.5) of the gradient, which we developed in Section 2.1. To fit a model-free setting, we consider that in general we do not know the exact form of c, but instead that at any point in time $s \in [t,T]$, we are able to observe our cumulative reward up to the current time,

$$R_s^{t,x,a} := \int_t^s f(r, X_r^{t,x,a}, I_r^{t,a})dr - \int_{(t,s]}\int_A c(r, X_{r-}^{t,x,a}, I_{r-}^{t,a}, e)\nu(dr, de).$$

This enables us, by observing our accumulated reward right before and after we change our action, so $R_{\tau_n-}^{t,x,a}$ before the jump and $R_{\tau_n}^{t,x,a}$ after the jump, to compute the cost term appearing in (2.5) as follows

$$c(\tau_n, X_{\tau_n-}^{t,x,a}, I_{\tau_n-}^{t,a}, I_{\tau_n}^{t,a}) = R_{\tau_n-}^{t,x,a} - R_{\tau_n}^{t,x,a},$$

and thus obtain the following formula for the policy gradient,

$$\nabla_\theta J(t,x,a,\theta) = \mathbb{E}^\theta\Big[\int_{(t,T]}\int_A \nabla_\theta(\log\lambda^\theta)(e|s, X_{s-}^{t,x,a}, I_{s-}^{t,a}) \cdot \Big(J(s, X_s^{t,x,a}, e, \theta) - J(s, X_{s-}^{t,x,a}, I_{s-}^{t,a}, \theta) + R_s^{t,x,a} - R_{s-}^{t,x,a}\Big)\nu(ds, de)\Big].$$

For the policy evaluation (PE) step, we can for example utilise a martingale loss function based on approach introduced in [15]. Their approach is based on the observation that for the true value function $v^\theta := J(\cdot, \theta)$ associated with a fixed policy λ^θ, the process

$$(v^\theta(s, X_s^{t,x,a}, I_s^{t,a}) + R_s^{t,x,a})_{s\in[t,T]} \tag{2.11}$$

is a martingale under $\mathbb{P}^\theta$, where $(X^{t,x,a}, I^{t,a})$ follow the intensity policy λ^θ. Now using that at terminal time the value function v^θ is just the terminal reward g, so

$$v^\theta(T, X_T^{t,x,a}, I_T^{t,a}) + R_T^{t,x,a} = g(X_T^{t,x,a}, I_T^{t,a}) + R_T^{t,x,a},$$

we can conclude that for all $s \in [t,T]$, under $\mathbb{P}^\theta$,

$$v^\theta(s, X_s^{t,x,a}, I_s^{t,a}) + R_s^{t,x,a} = \underset{\xi \text{ is } \mathcal{F}_s^{X,\nu}\text{-measurable}}{\arg\min} \mathbb{E}^\theta\left[|g(X_T^{t,x,a}, I_T^{t,a}) + R_T^{t,x,a} - \xi|^2\right].$$

Since J^κ is intended to approximate v^θ, this motivates us to consider the following martingale loss for our learned reward functional J^κ,

$$\mathrm{ML}(J^\kappa) \coloneqq \frac{1}{2}\,\mathbb{E}^\theta\left[\int_{(t,T]}\int_A |J^\kappa(s, X_s^{t,x,a}, I_s^{t,a}) + R_s^{t,x,a} - g(X_T^{t,x,a}, I_T^{t,a}) - R_T^{t,x,a}|^2 \nu(ds, de)\right].$$

This loss, in essence, quantifies how much we deviate from the martingale characterisation above. Another possible choice for $\mathrm{ML}(J^\kappa)$ would e.g. be $\mathbb{E}^\theta\left[\int_t^T |J^\kappa(s, X_s^{t,x,a}, I_s^{t,a}) + R_s^{t,x,a} - g(X_T^{t,x,a}, I_T^{t,a}) - R_T^{t,x,a}|^2 ds\right]$, which has been considered by [15]. To be able to learn the reward functional for a policy θ, we will update our estimate κ using the martingale loss $\mathrm{ML}(J^\kappa)$ by computing

$$\begin{aligned}&\nabla_\kappa \mathrm{ML}(J^\kappa)\\ &= \mathbb{E}^\theta\left[\int_{(t,T]}\int_A \Big(J^\kappa(s, X_s^{t,x,a}, I_s^{t,a}) + R_s^{t,x,a} - g(X_T^{t,x,a}, I_T^{t,a}) - R_T^{t,x,a}\Big)\nabla_\kappa J^\kappa(s, X_s^{t,x,a}, I_s^{t,a})\nu(ds, de)\right].\end{aligned}$$

By combining both steps, we then obtain the following generic actor-critic algorithm for randomised control problem in a model-free setting.

Algorithm 1: Offline-episodic actor-critic algorithm

```
**Input:** initial state $x_0$, initial action $a_0$, parametrised family of reward functions
    $(J^\kappa)_\kappa$, parametrised family of randomised intensity actions $(\lambda^\theta)_\theta$, initial
    learning rates $\eta_\kappa, \eta_\theta$, learning schedule $l(\cdot)$
**Output:** learned value function $J^\kappa$, optimal randomised control $\lambda^\theta$
initialise $\kappa, \theta$
**for** *episode* $j = 1, \dots$ **do**
    simulate $(X_t, I_t)_{t\in[0,T]}$ starting from $(X_0, I_0) = (x_0, a_0)$ according to the policy $\lambda^\theta$
      and observe the accumulated running reward $(R_t)_{t\in[0,T]}$ and the terminal reward
      $G_T = g(X_T, I_T)$
    compute $\nabla_\kappa \mathrm{ML}(J^\kappa) \leftarrow \sum_{I_{t-}\neq I_t}(J^\kappa(t, X_t, I_t) + R_t - G_T - R_T)\nabla_\kappa J^\kappa(t, X_t, I_t)$
    compute
      $\nabla_\theta J^\kappa \leftarrow \sum_{I_{t-}\neq I_t}(J^\kappa(t, X_t, I_t) - J^\kappa(t, X_{t-}, I_{t-}) + R_t - R_{t-})\nabla_\theta(\log\lambda^\theta)(I_t|t, X_{t-}, I_{t-})$

    update $\kappa \leftarrow \kappa - \eta_\kappa l(j)\nabla_\kappa \mathrm{ML}(J^\kappa)$
    update $\theta \leftarrow \theta + \eta_\theta l(j)\nabla_\theta J^\kappa$
**end**
```

Remark 2.6. *The offline-episodic algorithm can be adapted to an online setting with the following modifications. For the policy gradient $\nabla_\theta J^\kappa$, we focus on the most recent term in the sum. So suppose we are in j-th step at time t_j, then we update the policy gradient as follows*

$$\nabla_\theta J^\kappa \leftarrow (J^\kappa(t_j, X_{t_j}, I_{t_j}) - J^\kappa(t_j, X_{t_j-}, I_{t_{j-1}}) + R_{t_j} - R_{t_j-})\nabla_\theta(\log\lambda^\theta)(I_{t_j}|t_j, X_{t_j-}, I_{t_{j-1}}).$$

For the martingale loss $\nabla_\kappa ML(J^\kappa)$, again using as motivation that the process in (2.11) *is a martingale, we can formulate an online version by updating the martingale loss at time t_j with*

$$\nabla_\kappa ML(J^\kappa) \leftarrow (J^\kappa(t_{j-1}, X_{t_{j-1}}, I_{t_{j-1}}) + R_{t_{j-1}} - J^\kappa(t_j, X_{t_j}, I_{t_j}) - R_{t_j})\nabla_\kappa J^\kappa(t_{j-1}, X_{t_{j-1}}, I_{t_{j-1}}),$$

where t_{j-1} denotes the previous time step.

Remark 2.7. *In this initial paper, we do not explicitly address the exploration-exploitation tradeoff. However we believe that this can be achieved by suitably modifying the reward functional to include a penalty term with an entropy regulariser, as in [23]. Introducing such a term in our framework would help balance exploiting the current knowledge while encouraging exploration of new actions.*

Finally, it is important to note that while this algorithm addresses the randomised problem, our primary interest lies in (non-randomised) stochastic control problems; their randomised counterparts serve as tools for handling these control problems. Specifically, our objective will in general not be to find the optimal intensity λ^θ, but rather to find the optimal (non-randomised) control α. Therefore, in the next Section 3, we will discuss in more detail how to utilise this algorithm to find the optimal control for the corresponding stochastic control problems.

3. Application to stochastic control problems

In this section, we consider general stochastic control problems for which a randomised formulation in the form of Section 2 exists. The classical problem is the case of controlled Markov processes X^α, e.g., driven by diffusion processes, with regular controls α valued in A, and where the objective is to maximise over α a criterion in the form

$$\mathrm{J}(\alpha) = \mathbb{E}\Big[g(X_T^\alpha) + \int_0^T f(t, X_t^\alpha, \alpha_t)dt\Big].$$

The corresponding randomised formulation is the one described in Section 2 with $g(x)$ depending only on x, $c \equiv 0$, and the key result, proved in [19], see also [9], is the statement that the two value functions coincide, namely:

$$\sup_\alpha \mathrm{J}(\alpha) = \sup_{\lambda\in\mathcal{V}} \mathbb{E}^\lambda \Big[g(X_T) + \int_0^T f(t, X_t, I_t)dt\Big].$$

Such randomised formulations have been developed for a large class of continuous time control problems, including, but not limited to, impulse control problems in [18], optimal stopping in [10], or optimal switching problems in [2, 8] as it will be illustrated in the following Section 4 and in Appendix A. The core idea behind the randomisation framework is to replace the control by a random point process, usually a Poisson point process, whose intensity becomes the new control, which results in a randomised problem in form of Section 2. The advantage of this randomised formulation is that it provides a unified framework for many different classes of control problems, and not only for continuous time problems but even includes stochastic control in discrete time on deterministic and/or random grids. Our goal is then to develop an actor-critic algorithm for the original stochastic control problem by utilising its randomised counterpart and its actor-critic algorithm derived in Section 2.2. However, the drawback is that the randomised problem is not directly equivalent to the original problem. While one can often show that the value functions of both problems coincide, the optimal controls typically will not. In fact, in general, the randomised formulation does not have an optimal (randomised) control.

Despite this limitation, the randomised formulation is still a valuable tool. In the remainder of this section, we provide a heuristic explanation of how solving the randomised problem can help in recovering the optimal (non-randomised) control. Additionally, we describe how to construct an actor-critic algorithm to solve the original (non-randomised) stochastic control problem, based on the randomised framework presented in Section 2.

Let us for simplicity of presentation assume that $\hat{\nu}(ds, de) = \bar{\nu}(ds)\mu(de|s, X_{s-}, I_{s-})$, where $\bar{\nu}$ and μ are both non-random. Then $\bar{\nu}$ describes the distribution of points in ν in time, and μ is the kernel transition probability describing the mark distributions of such points. Similarly, we split λ^θ into an intensity Λ^θ for new points and a probability density $\bar{\lambda}^\theta$ for their marks as follows

$$\Lambda^\theta(s, x, a) := \int_A \lambda^\theta(e|s, x, a)\mu(de|s, x, a), \quad \bar{\lambda}^\theta(e|s, x, a) := \frac{\lambda^\theta(e|s, x, a)}{\Lambda^\theta(s, x, a)}, \quad (s, x, a) \in [0, T] \times \mathbb{R}^d \times A.$$

Let us further assume that $\mu(\{a\}|s, x, a) > 0$ for all $(s, x, a) \in [0, T] \times \mathbb{R}^d \times A$, so that at any point there is a non-negative probability that the jump of $\bar{\nu}$ does not induce a real jump in I. Given a function Λ, we now denote by $\Theta_{\leq\Lambda}$ (resp. $\Theta_{=\Lambda}$) the set of all $\theta \in \Theta$ such that $\Lambda^\theta \leq \Lambda$ (resp. $\Lambda^\theta = \Lambda$). Then we note that for every $\theta \in \Theta_{\leq\Lambda}$, the process $\lambda(e|s, x, a) := \lambda^\theta(e|s, x, a) + (\Lambda(s, x, a) - \Lambda^\theta(s, x, a))\frac{1}{\mu(\{a\}|s,x,a)} \in \mathcal{V}$ emulates the control λ^θ in the sense that $\mathbb{P}^\lambda_{(X,I)} = \mathbb{P}^\theta_{(X,I)}$. Therefore, supposing that $(\lambda_\theta)_{\theta\in\Theta}$ is sufficiently dense in $\mathcal{V}$, we see that λ can be approximated by $(\lambda^{\theta_n})_n$ such that $\theta_n \in \Theta_{=\Lambda}$, which shows that

$$\sup_{\theta\in\Theta_{\leq\Lambda}} J(t, x, a, \theta) = \sup_{\theta\in\Theta_{=\Lambda}} J(t, x, a, \theta).$$

This motivates us to view Λ as some kind of inverse step size, since as $\Lambda \to \infty$ (resp. $\Lambda\bar{\nu}(\{s\}) \uparrow 1$ if $\bar{\nu}(\{s\}) > 0$), we see that

$$\sup_{\theta\in\Theta_{=\Lambda}} J(t, x, a, \theta) \to \sup_{\theta\in\Theta} J(t, x, a, \theta).$$

Thus, we introduce the following restricted parameter sets for θ, for $\Lambda_c \geq 0$ and $0 \leq \Lambda_d \leq 1$,

$$\Theta_{\Lambda_c,\Lambda_d} = \{\theta \in \Theta \mid \Lambda^\theta(s, x, a) = \Lambda_c \mathbb{1}_{\{\bar{\nu}(\{s\})=0\}} + \tfrac{\Lambda_d}{\bar{\nu}(\{s\})}\mathbb{1}_{\{\bar{\nu}(\{s\})>0\}}, \text{ for all } (s, x, a)\} \subseteq \Theta.$$

Then, $\sup_{\theta\in\Theta_{\Lambda_c,\Lambda_d}} J \to \sup_{\theta\in\Theta} J$ as long as $\Lambda_c \to \infty$ and $\Lambda_d \uparrow 1$. Further, since while optimising over $\Theta_{\Lambda_c,\Lambda_d}$, the intensity is fixed, it is equivalent to optimise instead over the probability densities $(\bar{\lambda}_\theta)_{\theta\in\Theta_{\Lambda_c,\Lambda_d}}$, which by construction satisfy $\int_A \bar{\lambda}^\theta(e|\cdot)\mu(de|\cdot) \equiv 1$ and $\bar{\lambda}^\theta \geq 0$. Note that this family, if Θ is sufficiently exhaustive, does not actually depend on Λ_c, Λ_d anymore. Thus, by imposing an intensity schedule Λ_c and Λ_d ensuring that $\Lambda_c \to \infty$ and $\Lambda_d \uparrow 1$, we obtain Algorithm 2.

At the same time, we recall that solving the randomised problem was however not our original goal. Instead, it serves as a tool for solving the original (non-randomised) problem. In particular, the $\bar{\lambda}^\theta$ we obtain from Algorithm 1 is not our desired control. We recall that instead $\bar{\lambda}^\theta$ represents the intensity for I, and the process I then actually plays the role of the sought-after control process α. Therefore, let us define for each θ also a control α^θ as e.g. the arg max of the distribution $\bar{\lambda}^\theta(e|s, x, a)\mu(de|s, x, a)$. The motivation is that at each jump s of I, we draw our new control I_s from the distribution $\bar{\lambda}^\theta(e|s, X_{s-}, I_{s-})\mu(de|s, X_{s-}, I_{s-})$, and if the intensity $\Lambda^\theta \to \infty$ (resp. $\Lambda^\theta\bar{\nu}(\{s\}) \uparrow 1$ if $\bar{\nu}(\{s\}) > 0$), then we are essentially able to draw a new control at every time point $s \in [t, T]$, just as in the original control problem. Consequently, letting in our case $\Lambda_c \to \infty$ and $\Lambda_d \uparrow 1$, this then leads to the convergence of $\alpha^\theta \to \alpha_*$.

Algorithm 2: Offline-episodic actor-critic algorithm with random grids and intensity schedule

```
Input: initial state $x_0$, initial action $a_0$, terminal time $T$, parametrised family of
       reward functions $(J^\kappa)_\kappa$, baseline random grid sampling distribution $\bar{\nu}$, baseline
       action distribution kernel $\mu$, parametrised family of action densities $(\bar{\lambda}^\theta)_\theta$,
       intensity schedule $\Lambda_c(\cdot)$, $\Lambda_d(\cdot)$, initial learning rates $\eta_\kappa, \eta_\theta$, learning schedule $l(\cdot)$
Output: learned value function $J^\kappa$, optimal randomised control $\lambda^\theta$, optimal
        (non-randomised) control $\alpha^\theta$
initialise $\kappa, \theta$
for episode $j = 1, \dots$ do
    initialise $\tau_0 \leftarrow 0$, $r_0 \leftarrow 0$
    simulate point process $U$ on $(0, T]$ with stochastic intensity
      $\Lambda_c(j)\mathbb{1}_{\{\bar{\nu}(\{s\})=0\}}\bar{\nu}(ds) + \Lambda_d(j)\mathbb{1}_{\{\bar{\nu}(\{s\})>0\}}\delta_s(ds) \to$ obtain grid points $(\tau_n)_{n=1,\dots,N}$
    for $n = 1, ..., N$ do
        simulate $X_{[\tau_{n-1}, \tau_n)}$ from $X_{\tau_{n-1}} = x_{n-1}$ with control $a_{n-1}$
        observe the new state $x_{n-} \leftarrow X_{\tau_n-}$ and accumulated running reward
          $r_{n-} \leftarrow R_{\tau_n-}$ at time $\tau_n-$
        simulate and update the new control $a_n \sim \bar{\lambda}^\theta(e|\tau_n, x_{n-}, a_{n-1})\mu(de|\tau_n, x_{n-}, a_{n-1})$
        observe the new state $x_n \leftarrow X_{\tau_n}$ and accumulated running reward $r_n \leftarrow R_{\tau_n}$
          after updating the control at time $\tau_n$
    end
    simulate $X_{[\tau_N, T]}$ with control $a_N$
    observe the final state $x_{N+1} \leftarrow X_T$ and set $a_{N+1} \leftarrow a_N$, $\tau_{N+1} \leftarrow T$
    observe the final accumulated running reward $r_{N+1} \leftarrow R_T$ and the terminal reward
      $G_T = g(X_T, I_T)$ at time $T$
    compute $\nabla_\kappa \mathrm{ML}(J^\kappa) \leftarrow \sum_{a_n \neq a_{n-1}} (r_n + J^\kappa(\tau_n, x_n, a_n) - G_T - r_{N+1}) \nabla_\kappa J^\kappa(\tau_n, x_n, a_n)$
    compute $\nabla_\theta J^\kappa \leftarrow$
      $\sum_{a_n \neq a_{n-1}} (J^\kappa(\tau_n, x_n, a_n) - J^\kappa(\tau_n, x_{n-}, a_{n-1}) + r_n - r_{n-}) \nabla_\theta (\log \bar{\lambda}^\theta)(a_n|\tau_n, x_{n-}, a_{n-1})$

    update $\kappa \leftarrow \kappa - \eta_\kappa l(j) \nabla_\kappa \mathrm{ML}(J^\kappa)$
    update $\theta \leftarrow \theta + \eta_\theta l(j) \nabla_\theta J^\kappa$
end
obtain $\alpha^\theta(t, x, a)$ as the arg max of the probability distribution $\bar{\lambda}^\theta(e|t, x, a)\mu(de|t, x, a)$
```

Finally, we want to conclude this section by giving a version of the actor-critic algorithm but with a “fixed step size“. This is the version we will also use later in Section 4. So we will fix the intensity Λ and thus the base process $\tilde{\nu} := \Lambda\bar{\nu}$, and now optimise again over the parametrised family of action densities $(\bar{\lambda}^\theta)_\theta$, which results in the following Algorithm 3.

Remark 3.1. *This leads to a flexible framework accommodating various types of sampling grids, such as*

- *deterministic discrete grids, by choosing $\tilde{\nu} = \sum_{k=0}^{N} \delta_{\frac{kT}{N}}(ds)$, for $N \geq 2$,*
- *random discrete grids, by setting $\tilde{\nu} = \sum_{k=0}^{N} p_{samp} \delta_{\frac{kT}{N}}(ds)$, for $N \geq 2$ and $p_{samp} \in (0,1]$,*
- *Poisson grids are possible with $\tilde{\nu} = \lambda ds$ for some intensity $\lambda > 0$.*

In Section 4, we will compare the choice of deterministic and random discrete grids through two numerical examples of optimal switching problems.

Algorithm 3: Offline-episodic actor-critic algorithm with random grids

```
Input: initial state $x_0$, initial action $a_0$, terminal time $T$, parametrised family of
       reward functions $(J^\kappa)_\kappa$, random grid sampling distribution $\tilde{\nu}$, baseline action
       distribution kernel $\mu$, parametrised family of action densities $(\bar{\lambda}^\theta)_\theta$, initial
       learning rates $\eta_\kappa, \eta_\theta$, learning schedule $l(\cdot)$
Output: learned value function $J^\kappa$, optimal randomised control $\lambda^\theta$, optimal
        (non-randomised) control $\alpha^\theta$
initialise $\kappa, \theta$
for episode $j = 1, \dots$ do
    initialise $\tau_0 \leftarrow 0$, $r_0 \leftarrow 0$
    simulate point process on $(0, T]$ with stochastic intensity $\tilde{\nu}$ $\rightarrow$ obtain grid points
      $(\tau_n)_{n=1,\dots,N}$
    for $n = 1, ..., N$ do
        simulate $X_{[\tau_{n-1}, \tau_n)}$ from $X_{\tau_{n-1}} = x_{n-1}$ with control $a_{n-1}$
        observe the new state $x_{n-} \leftarrow X_{\tau_n -}$ and accumulated running reward
          $r_{n-} \leftarrow R_{\tau_n -}$ at time $\tau_n -$
        simulate and update the new control $a_n \sim \bar{\lambda}^\theta(e|\tau_n, x_{n-}, a_{n-1})\mu(de|\tau_n, x_{n-}, a_{n-1})$
        observe the new state $x_n \leftarrow X_{\tau_n}$ and accumulated running reward $r_n \leftarrow R_{\tau_n}$
          after updating the control at time $\tau_n$
    end
    simulate $X_{[\tau_N, T]}$ with control $a_N$
    observe the final state $x_{N+1} \leftarrow X_T$ and set $a_{N+1} \leftarrow a_N$, $\tau_{N+1} \leftarrow T$
    observe the final accumulated running reward $r_{N+1} \leftarrow R_T$ and the terminal reward
      $G_T = g(X_T, I_T)$ at time $T$

    compute $\nabla_\kappa \mathrm{ML}(J^\kappa) \leftarrow \sum_{a_n \neq a_{n-1}} (r_n + J^\kappa(\tau_n, x_n, a_n) - G_T - r_{N+1}) \nabla_\kappa J^\kappa(\tau_n, x_n, a_n)$
    compute $\nabla_\theta J^\kappa \leftarrow$
      $\sum_{a_n \neq a_{n-1}} (J^\kappa(\tau_n, x_n, a_n) - J^\kappa(\tau_n, x_{n-}, a_{n-1}) + r_n - r_{n-}) \nabla_\theta (\log \bar{\lambda}^\theta)(a_n|\tau_n, x_{n-}, a_{n-1})$

    update $\kappa \leftarrow \kappa - \eta_\kappa l(j) \nabla_\kappa \mathrm{ML}(J^\kappa)$
    update $\theta \leftarrow \theta + \eta_\theta l(j) \nabla_\theta J^\kappa$
end
define $\alpha^\theta(t, x, a)$ as the arg max of the probability distribution $\bar{\lambda}^\theta(e|t, x, a)\mu(de|t, x, a)$
```

Remark 3.2. *Both Algorithms 2 and 3 can be transformed to online algorithms, similar as in Remark 2.6 for Algorithm 1, by replacing $\nabla_\theta J^\kappa$ at time τ_n with*

$$\nabla_\theta J^\kappa \leftarrow (J^\kappa(\tau_n, x_n, a_n) - J^\kappa(\tau_n, x_{n-}, a_{n-1}) + r_n - r_{n-}) \nabla_\theta (\log \bar{\lambda}^\theta)(a_n|\tau_n, x_{n-}, a_{n-1}).$$

and $\nabla_\kappa ML(J^\kappa)$ *with*

$$\nabla_\kappa ML(J^\kappa) \leftarrow (r_{n-1} + J^\kappa(\tau_{n-1}, x_{n-1}, a_{n-1}) - J^\kappa(\tau_n, x_n, a_n) - r_n)\nabla_\kappa J^\kappa(\tau_{n-1}, x_{n-1}, a_{n-1}),$$

4. Numerical experiments for switching problems using neural networks

4.1. Optimal switching problem

In this paragraph, we recall the connection between optimal switching problems with their randomised formulation following [2]. Note that this randomisation method has been extended to large class of further problems including impulse control, optimal stopping and regular control problems, and thus such problems also fit into the unified setting for our policy gradient method.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space carrying an m-dimensional Brownian motion W and let $\mathbb{F}^W$ be its generated filtration. Let $A = \{1, \dots, N\}$ for some fixed $N \in \mathbb{N}$ denote the finite action set. A switching control is an A-valued piece-wise constant process of the form

$$\alpha = a\mathbb{1}_{[t,\tau_0)} + \sum_{n\in\mathbb{N}} \xi_n \mathbb{1}_{[\tau_n,\tau_{n+1})},$$

where $(\tau_n)_n$ is an increasing sequence of stopping times such that $\tau_n \to \infty$ $\mathbb{P}$-a.s., $(\xi_n)_n$ is a sequence of A-valued random variables such that ξ_n is $\mathcal{F}^W_{\tau_n}$-measurable and $a \in A$ is a fixed initial control. Given an initial value $x \in \mathbb{R}^d$, we further consider the controlled state dynamics as the solution to the stochastic differential equation

$$X_s^{t,x,\alpha} = x + \int_t^s b(r, X_r^{t,x,\alpha}, \alpha_r)dr + \int_t^s \sigma(r, X_r^{t,x,\alpha}, \alpha_r)dW_r + \sum_{t<\tau_n\le s} \gamma(\tau_n, X_{\tau_n-}^{t,x,\alpha}, \alpha_{\tau_n-}, \alpha_{\tau_n}). \tag{4.1}$$

We denote by $\mathcal{A}$ as the set of as all switching controls α. Note that together with the following Assumption A, this ensures that the above state dynamics (4.1) are well-defined.

Assumption A. *The coefficients b, σ and γ are Lipschitz and of linear growth w.r.t. x: there exists a positive constant C s.t. for all* $t \in [0, T]$, $x, x' \in \mathbb{R}^d$, $a, a' \in A$,

$$|b(t,x,a) - b(t,x',a)| + |\sigma(t,x,a) - \sigma(t,x',a)| + |\gamma(t,x,a,a') - \gamma(t,x',a,a')| \le C|x - x'|,$$
$$|b(t,x,a)| + |\sigma(t,x,a)| + |\gamma(t,x,a,a')| \le C(1 + |x|).$$

Our goal is now to maximise the following reward functional

$$\mathrm{J}(t,x,a,\alpha) := \mathbb{E}\left[g(X_T^{t,x,\alpha}, \alpha_T) + \int_t^T f(s, X_s^{t,x,\alpha}, \alpha_s)ds - \sum_{t<\tau_n\le T} c(\tau_n, X_{\tau_n-}^{t,x,\alpha}, \alpha_{\tau_n-}, \alpha_{\tau_n})\right],$$

and we define the value function as follows

$$V(t,x,a) := \sup_{\alpha\in\mathcal{A}} \mathrm{J}(t,x,a,\alpha).$$

We make the standard assumptions on the gain and cost functions:

Assumption B. *The reward functions f, g and the cost function c are continuous w.r.t. the x argument with quadratic growth condition: there exists some positive constant C s.t. for all $t \in [0,T]$, $x \in \mathbb{R}^d$, $a, a' \in A$,*

$$|f(t,x,a)| + |g(x,a)| + |c(t,x,a,a')| \leq C(1+|x|^2).$$

To formulate the randomised version of this problem as in Section 2, we introduce an independent random point process ν on $[0,T] \times A$ with predictable projection $\hat{\nu}$. Correspondingly, we define the A-valued process

$$I_s^{t,a} = a + \int_{(t,s]} \int_A (e - I_{r-}^{t,a}) \nu(dr, de), \quad s \in [t,T],$$

which will replace our control process. In particular, the state process will follow the following uncontrolled state dynamics

$$X_s^{t,x,a} = x + \int_t^s b(r, X_r^{t,x,a}, I_r^{t,a}) dr + \int_t^s \sigma(r, X_r^{t,x,a}, I_r^{t,a}) dW_r + \int_{(t,s]} \int_A \gamma(r, X_{r-}^{t,x,a}, I_{r-}^{t,a}, e) \nu(dr, de).$$

Our set of control will instead now be the set $\mathcal{V}$ of $Pred(\mathbb{F}^{W,\nu}) \otimes \mathcal{B}(A)$-measurable, essentially bounded processes λ such that there exists a with respect to $\mathbb{P}$ absolutely continuous probability measure $\mathbb{P}^\lambda \ll \mathbb{P}$ under which ν is a random point process with predictable projection $\lambda_s(e)\hat{\nu}(ds, de)$, see also (2.3) for a characterisation of such $\lambda \in \mathcal{V}$. Then the reward functional is defined by

$$J(t,x,a,\lambda) := \mathbb{E}^{\mathbb{P}^\lambda}\left[g(X_T^{t,x,a}, I_T^{t,a}) + \int_t^T f(s, X_s^{t,x,a}, I_s^{t,a}) ds - \int_{(t,T]} c(s, X_{s-}^{t,x,a}, I_{s-}^{t,a}, e) \nu(ds, de) \right],$$

and we introduce the following randomised value function

$$V^{\mathcal{R}}(t,x,a) := \sup_{\lambda \in \mathcal{V}} J(t,x,a,\lambda).$$

Bouchard [2] studied the case where ν is a Poisson point process with compensator $\hat{\nu}(ds, de) = \sum_{a \in A} \delta_a(de) ds$, for which the set of admissible randomised controls $\mathcal{V}$ then reduces to all $Pred(\mathbb{F}^{W,\nu}) \otimes \mathcal{B}(A)$-measurable, essentially bounded processes λ. Under some additional regularity and growth assumptions, it is proved the following equivalence result between the optimal switching and the randomised problem.

Theorem 4.1 ([2, Theorem 2.1])**.** *Let Assumptions A and B hold and ν have a predictable projection of the form $\hat{\nu}(ds, de) = \sum_{a \in A} \delta_a(de) ds$. Further assume that the regularity assumptions [2, Assumptions H1, H2] are satisfied. Then value functions of both problems coincide, that is $V = V^{\mathcal{R}}$.*

Remark 4.2. *While usually ν is chosen as Poisson point process, any* sufficiently dense *point measure, under suitable assumptions, would work for such an equivalence result. In particular, starting from any given point process, even with deterministic atoms as in the case of a deterministic grid, and by e.g. adding additional points sampled from a Poisson point process, one would obtain such a* sufficiently dense *point measure.*

In the sequel for our numerical experiments, we consider an optimal switching problem where part of the state which is not controlled is continuous, while the other part is controlled and takes discrete values. Therefore the randomised controlled state is modeled by a discrete Markov chain described by probability transitions which are functions of the global state. At convergence, we expect that these probabilities converge either to 1 or 0, are discontinuous in time for a fixed state, so that they cannot be represented as functions of time using neural networks. Consequently, we use a deterministic uniform grid of N dates on $[0, T]$. We note $t_n = n\Delta t$ where $\Delta t = \frac{T}{\bar{N}}$ with $\bar{N} = N - 1$ the number of time steps. At each time step and each possible state, a neural network is used to describe the transition probabilities from one state to the other on.

We then propose :

- either to sample time randomly on the deterministic grid : the number of points $\tilde{N}$ chosen on the grid is sampled using a binomial distribution with a probability p_{samp} and a number of trials equal to $N-2$. Then the points from the grid are chosen randomly with an uniform law,
- or to take the N points grid corresponding to $p_{samp} = 1$.

We denote by $(\tau_k)_{k\geq 0}$ the random lattice sampled from the deterministic lattice with $\tau_0 = 0$, and complete it with the convention that $\tau_p = T$ for $k < p \leq N$ if $\tau_k = T$. We set $[k] = \frac{\tau_k}{\Delta t}$ the random grid index associated with values in $[0, N]$.

Remark 4.3. *The fact that the control has to be modeled at each time step by different networks to get good results for degenerated controlled states with constraints was already shown in the case of reservoir optimization in [24].*

Next, depending on the problem, it may be interesting to take a representation of the reward functional J different form the one proposed by $\mathrm{ML}_{\mathcal{T}}(J^{\kappa})$ and in the different examples below we detail the representation taken.

In the two examples below, we model the energy curve using the classical HJM model as [24]:

$$\frac{dF(t,T)}{F(t,T)} = e^{-\beta(T-t)}\sigma dW_t \tag{4.2}$$

where W_t is a one-dimensional Brownian motion defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The spot price is then equal to $S_t = F(t,t)$. As numerical example, we take $T = 30$, $F(0,t) = 90 + 10\cos(2\pi\lfloor \frac{t}{30} \rfloor)$, $\beta = 0.15$, $\sigma = 0.5$.

4.2. Starting and stopping in physical assets

We consider the problem of a thermal asset generating power as in [13],[22]. The asset has a production cost of K per time unit, and has two states : either on (state 1) or off (state 0) and the switching control $\alpha = (\alpha_t)_t$ is

$$\alpha_t = \alpha_0 \mathbb{1}_{[\tau_0,\tau_1)} + \sum_{n>0} \xi_n \mathbb{1}_{[\tau_n,\tau_{n+1})}(t), \quad 0 < t \leq T,$$

with $\alpha_0 = 1$ (the asset is on at $t = 0$), and the random variables $(\xi_n)_n$ denote the sequence of operating regimes valued in $A = \{0, 1\}$, representing the decisions to stop or run the production. There is a fixed cost for switching from a mode to another one, namely $c_{0,1}$ (resp. $c_{1,0}$) for starting (resp. stopping) the production.

The manager of the power asset aims to maximize over α the expected global profit:

$$\mathrm{J}(\alpha) = \mathbb{E}\Big[\int_0^T f(S_t, \alpha_t)dt - \sum_n c_{\alpha_{\tau_n}, \alpha_{\tau_{n+1}}}\Big],$$

with running profit functions $f(s, 0) = 0$, and $f(s, 1) = s - K$.

- As for the control, we model the switching probability at each time step n by two neural networks depending on the price, and using a sigmoid function at the output layer, $\bar{\lambda}^{\theta_{n,i}}(S)$ takes values in $[0, 1]$ for $i = 0, 1$ with parameters $\theta_{n,i}$ and such that for $i \in \{0, 1\}$, $\bar{\lambda}^{\theta_{n,i}}$ is the probability that, given the state i in t_n^-, the asset will change to state $1 - i$ in t_n. Here, $\theta = ((\theta_{n,i})_{n=1,\bar{N}-1})_{i=0,1}$.
- As for the value function J we use similarly for each time step n two neural networks depending on the price S: $(J^{\kappa_{n,i}}(S))_{i=0,1}$ taking values in $\mathbb{R}$ with parameters $\kappa_{n,i}$ where $J^{\kappa_{n,i}}$ is the value function in state i for $i \in \{0, 1\}$. Here, $\kappa = ((\kappa_{n,i})_{n=0,\bar{N}-1})_{i=0,1}$. We also take the convention $J^{\kappa_{[N]},i} = 0$ for $i = 0, 1$.

For this example, to estimate the reward function J, we propose to minimize the loss function

$$\sum_{k=0}^{\bar{N}-1} \mathbb{E}\Big[\Big|J^{\kappa[k],\xi_k}(S_{\tau_k}) - \sum_{n=0}^{\bar{N}-1}(S_{t_n}^{\tau_k} - K)\Delta t \mathbb{1}_{\{\alpha_{t_n}^{\xi_k}=1\}}\mathbb{1}_{\{t_n \geq \tau_k\}} + \sum_{n \geq k} c_{\alpha_{\tau_n}^{\xi_k}, \alpha_{\tau_{n+1}}^{\xi_k}} \mathbb{1}_{\{\tau_{n+1} < T\}}\Big|^2\Big]$$

with the convention $c_{i,i} = 0$, $i = 0, 1$ and where for $k = 0, \dots, \bar{N} - 1$, ξ_k is the state regime with values in $\{0, 1\}$ which is sampled uniformly. $\alpha_t^{\xi_k}$ for $t \geq \tau_k$ denote the regimes sampled randomly using probabilities $(\bar{\lambda}^{\theta_{[l], \alpha_{\tau_{l-1}}^{\xi_k}}}(S_{\tau_l}))_{l>k}$ starting at date τ_k with a value ξ_k, while $S_{t_n}^{\tau_k}$ is the asset value in t_n obtained by sampling from its initial distribution in τ_k according to the asset law.

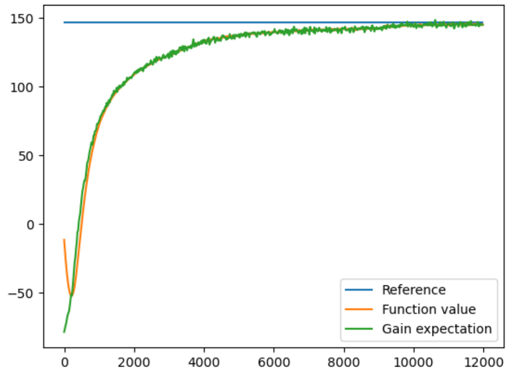
The gradient function $\nabla_\theta J^\kappa$ is estimated locally for each time step and the sum of the local gradients $DW(\theta)$ is used

$$DW(\theta) = \sum_{n=0}^{\bar{N}-2} \mathbb{E}[\nabla_\theta \log(\bar{\lambda}^{\theta_{n+1,\xi_n}}(S_{t_{n+1}}))(J^{\kappa_{n+1,\alpha_{t_{n+1}}^{\xi_n}}}(S_{t_{n+1}}) - J^{\kappa_{n+1,\xi_n}}(S_{t_{n+1}}) - c_{\xi_n, \alpha_{t_{n+1}}^{\xi_n}})]$$

where once again for each n in the loop $S_{t_{n+1}}$ is sampled according the asset law at date t_{n+1}, ξ_k is sampled uniformly in $\{0, 1\}$, and $\alpha_{t_{n+1}}^{\xi_n}$ is sampled according the probability $\bar{\lambda}^{\theta_{n+1,\xi_n}}(S_{t_{n+1}})$.

Remark 4.4. *Instead of simulating the process X in forward direction as in Algorithm 3 and evaluating $\nabla_\theta J^\kappa$, we use a local version of the gradient which randomly samples the state at every possible time step. By randomly choosing the states at each time step, we explore all possible states and thus get better results. This kind of approach is generally used in classical actor critic methods, where the control is randomized and taken as a normal law with decreasing variance with iterations (see for example [21]).*

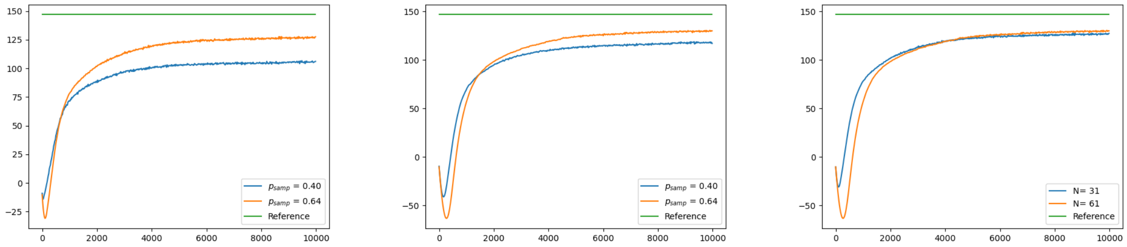
We use the ADAM algorithm with $\eta_\theta = 0.00015$, $\eta_\kappa = 0.03$. As noticed in [21], it is crucial to have $\eta_\theta << \eta_\kappa$ to have good convergence. The tanh activation function is taken for the activation functions in the hidden layers, while the sigmoid activation function is used for the output layer for the probabilities. The batch size is equal to 10000. The references are calculated using dynamic programming in the StOpt library [12] where regression are calculated using adapted linear regression per mesh [3] and 30 time steps so $N = 31$ are used. In the deterministic case, we get a value of 86, while using $\sigma = 0.15$, the value function is equal to 146.9. On Figure 1, we plot the convergence of the actor critic algorithm in the stochastic case using $p_{samp} = 1$. The convergence to the correct value is achieved after more than 10000 gradient iterations. In the graph, the "Function value" is obtained using the J^κ approximation of J, while the "Gain expectation" is obtained using the gain estimate in the simulation (the controls and grids are sampled).



$p_{samp} = 1.$, $N = 31$

Figure 1: Convergence of the actor critic algorithm for the thermal switching problem depending on the gradient iteration.

On Figure 2, we explore the effect of extra temporal randomization. Not surprisingly, using a fixed lattice to sample from degrades the results as the sampling ratio p_{samp} is lower. Similarly, by fixing the sampling ratio, the results improve as we increase the number of time steps of the lattice.

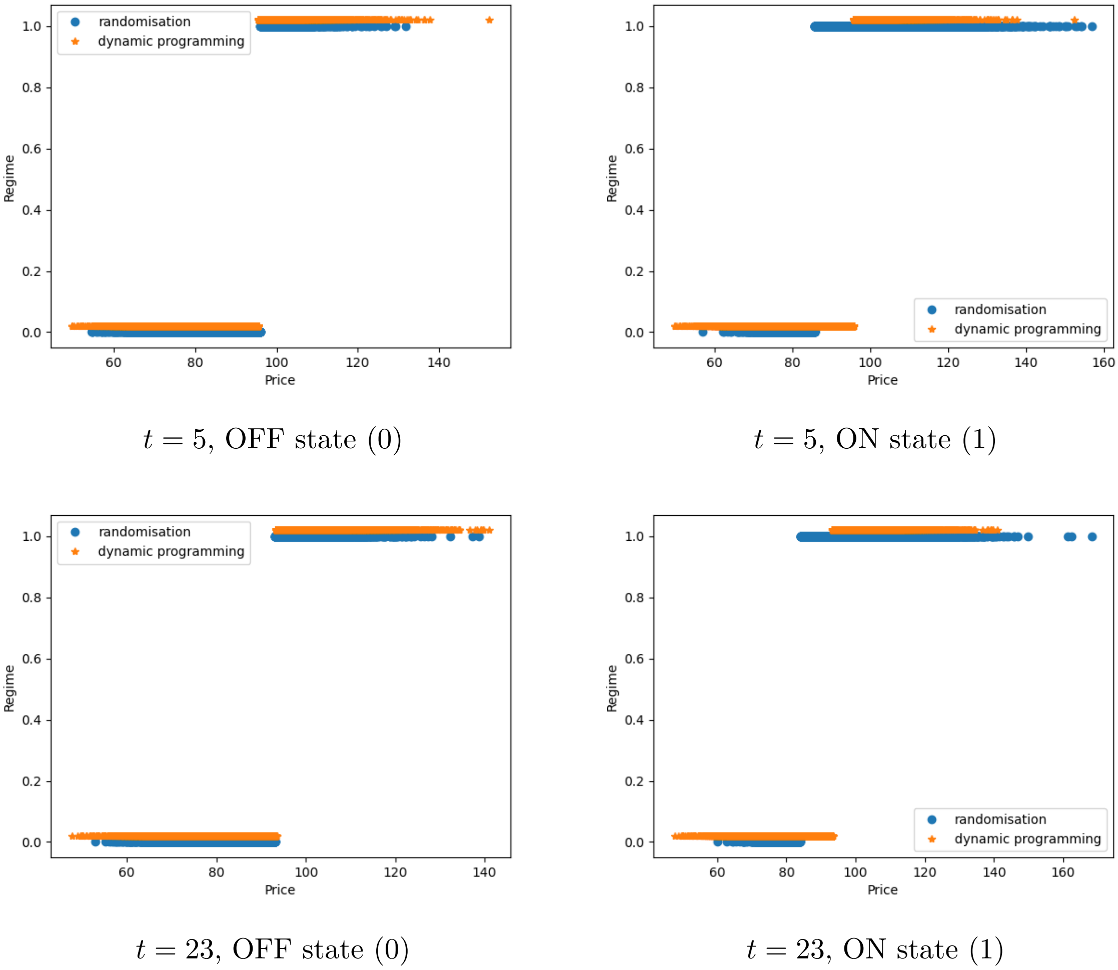


$N = 31$ $N = 61$ $p_{samp} = 0.64$

Figure 2: Convergence of the actor critic algorithm (function value) for the thermal switching problem depending on the gradient iteration letting p_{samp} or N vary.

It is also possible to use the estimated probability and in the simulation, using an Eulerian scheme, the control is selected as the most probable. In the stochastic case, we get a value equal to 146.0 using $p_{samp} = 1$ at the end of the iterations, while the value obtained with $N = 61$, $p_{samp} = 0.64$ is 139.1.

We give the control obtained in simulation (taken as the one with the highest probability) using $p_{samp} = 1$ on Figure 3 and the control calculated by dynamic programming in the StOpt library [12]. The control is globally well computed, although we observe that it is not fully convergent in the ON state.



$t = 5$, OFF state (0)

$t = 5$, ON state (1)

$t = 23$, OFF state (0)

$t = 23$, ON state (1)

Figure 3: Control obtained in the stochastic thermal test case depending the state for two date.

4.3. A storage model

We consider the example of a battery storage valuation formulated as an optimal switching problem, see [4], [24]: the manager of the battery aims to price its real options value by optimizing over a finite horizon the dynamic decisions to inject or withdraw power. The inventory process of the battery is denoted by $(K_t)_t$, and is controlled by a switching control $\alpha = (\alpha_t)_t$:

$$\alpha_t = \alpha_0 \mathbb{1}_{[\tau_0,\tau_1)} + \sum_{n>0} \xi_n \mathbb{1}_{[\tau_n,\tau_{n+1})}(t), \quad 0 < t \leq T,$$

where the random variables $(\xi_n)_n$ denote the sequence of operating regimes valued in $A = \{-1, 0, 1\}$, representing the decisions to withdraw, do nothing, or inject power. At $t = 0$, the battery is withdrawing power so that $\alpha_0 = -1$.

The effort of moving from regime $i \in A$ to another regime $j \in A$ incurs a cost $c_{i,j}$ with $c_{i,i} = 0$, $c_{i,j} > 0$ for $i \neq j$. The inventory is given by: $K_t = \int_0^t \alpha_s ds$, while satisfying the physical constraint: $K_t \in [0, K_{max}]$, for all $t \in [0, T]$. Therefore the number of discrete states is $k_{max} + 1$ where $k_{max} = \frac{K_{max}}{\Delta t}$. The exogenous price of the electricity is governed by equation (4.2). The objective of the manager is to maximize over switching control α the reward functional

$$J(\alpha) = \mathbb{E}\Big[\int_0^T f(S_t, \alpha_t)dt - \sum_n c_{\alpha_{\tau_n}, \alpha_{\tau_{n+1}}}\Big],$$

with a running profit function

$$f(s,a) = \begin{cases} -s, & \text{for } a = 1 \\ 0, & \text{for } a = 0 \\ s, & \text{for } a = -1. \end{cases}$$

Similarly as in the previous example:

- We model the switching probability at each time step n and at each inventory k by a neural network $\bar{\lambda}^{\theta_{n,k}}(S_{t_i})$ with parameters $\theta_{n,k}$ and with an output in $[0,1]^9$. When injection, do nothing and withdraw are allowed (so when $k \in \{1, \ldots, k_{max} - 1\}$), for $(l, m) \in \{-1, 0, 1\} \times \{-1, 0, 1\}$, $\bar{\lambda}^{\theta_{n,k}}_{l,m}(S_{t_n})$ represents the probability to go from state l at t_n^- to state m at t_n. When withdraw (respectively injection) is not allowed therefore when $k = 0$ (respectively $k = k_{max}$), $\bar{\lambda}^{\theta_{n,k}}_{l,-1}$ (respectively $\bar{\lambda}^{\theta_{n,k}}_{l,1}$) is set to 0. These neural networks use a softmax activation function at the output layer to satisfy that probabilities are positive with a sum equal to 1.

- As for the value function J we use similarly for each time step n and for a level k a neural network $J^{\kappa_{n,k}}(S_{t_n})$ with parameters $\kappa_{n,k}$ and an output in dimension 3 where $J_i^{\kappa_{n,k}}$ is at date t_n and level k, the value function in state i for $i \in \{-1, 0, 1\}$. As previously we take the convention $J_i^{\kappa[N],k} = 0$ for $i \in \{-1, 0, 1\}$ and each level k.

Comparing to the thermal switching asset,

- the state encompass the inventory level and we minimize the following loss function to estimate J:

$$\sum_{n=0}^{\bar{N}-1} \mathbb{E}\Big[\Big|J^{\kappa[n],K^n_{\tau_n}}_{\alpha^n_{\tau_n}}(S^n_{\tau_n}) - \sum_{p=0}^{\bar{N}-1} f(S^n_{t_p}, \alpha^n_{t_p})\mathbb{1}_{\{t_p \geq \tau_n\}}\Delta t + \sum_{p \geq n} c_{\alpha^n_{\tau_p}, \alpha^n_{\tau_{p+1}}} \mathbb{1}_{\{\tau_{p+1} < T\}}\Big|^2\Big]$$

  where at date τ_n in the outer summation $S^n_{\tau_n}$ is sampled according the asset law at date τ_n, while the inventory level $K^n_{\tau_n}$ and control applied $\alpha^n_{\tau_n}$ are sampled uniformly. $S^n_{t_p}$ is the asset value at date $t_p > \tau_n$ conditionally to its value at date τ_n, and $\alpha^n_{t_p}$ the applied control at date t_p starting from $\alpha^n_{\tau_n}$ at date τ_n and obtained sampling the switching probabilities as for the thermal asset. Therefore the flow equation for the inventory level is given for $t_p > \tau_n$ by

$$K^n_{t_{p+1}} = 0 \vee (K_{t_p} + \alpha^n_{t_p} \Delta t) \wedge K_{max} \tag{4.3}$$

- The gradient function DW is estimated with similar notations as

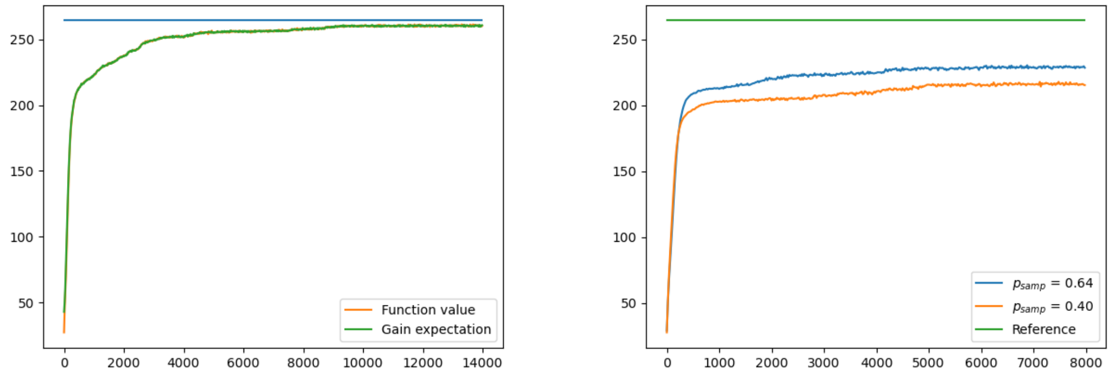
$$DW(\theta) = \sum_{n=1}^{\bar{N}-1} \mathbb{E}[\nabla_\theta \log(\bar{\lambda}^{\theta_n, K_{t_n}}_{\alpha^{n-1}_{t_{n-1}}, \alpha^{n-1}_{t_n}}(S_{t_n}))(J^{\kappa_n, K_{t_n}}_{\alpha^{n-1}_{t_n}}(S_{t_n}) - J^{\kappa_n, K_{t_n}}_{\alpha^{n-1}_{t_{n-1}}}(S_{t_n}) - c_{\alpha^{n-1}_{t_{n-1}}, \alpha^{n-1}_{t_n}})]$$

  where once again for each n in the loop, S_{t_n} is sampled according the asset law at date t_n, while the inventory level K_{t_n} at date t_n and the control $\alpha^{n-1}_{t_{n-1}}$ at t_n^- are sampled uniformly. The control $\alpha^{n-1}_{t_n}$ is sampled from the state $(t_n, K_{t_n}, \alpha^{n-1}_{t_{n-1}})$ using the probabilities $\bar{\lambda}^{\theta_n, K_{t_n}}_{\alpha^{n-1}_{t_{n-1}}, \alpha^{n-1}_{t_n}}(S_{t_n})$.

Remark 4.5. *We have to clip values in the flow equation* (4.3)*. A possible control for a single time step may be not admissible if $\tau_{n+1} - \tau_n > 1$. Then if we inject (control $\alpha = 1$), and if the control is admissible during one time step and not for two, the control is changed to* 0 *on the second time step and a switching cost is added. A similar adaptation is carried out in the withdrawal regime.*

As numerical example, we take the following switching costs: $c_{-1,1} = c_{1,-1} = 5$, and for (i, j) not in $\{(-1, 1), (1-1), (-1, -1), (0, 0), (1, 1)\}$, $c_{i,j} = 3$. We take $K_0 = 2$, $K_{max} = 2$ and the reference calculated with the StOpt library using 30 time steps is 264.3.

We use a batch size of 10000, $\eta_\theta = 0.00015$ and $\eta_\kappa = 0.05$ with ADAM optimizers.



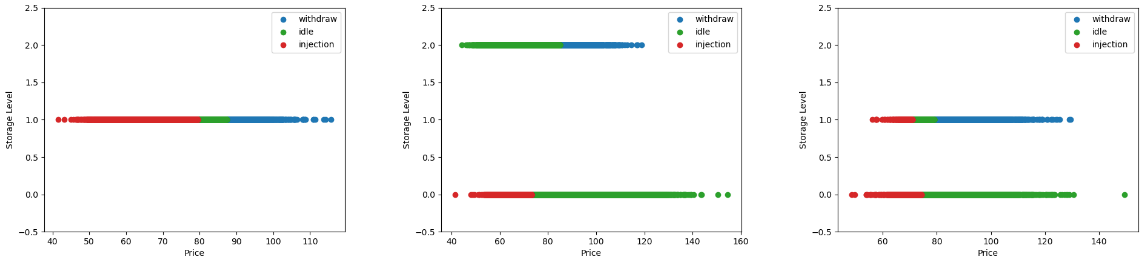
$p_{samp} = 1$, $N = 31$.

Results letting p_{samp} vary for $N = 31$.

Figure 4: Convergence of the actor critic algorithm for the battery storage case.

On Figure 4, we observe that taking $p_{samp} = 1$ allows to recover almost the exact solution and, as for the thermal asset, the results deteriorate as p_{samp} decreases.

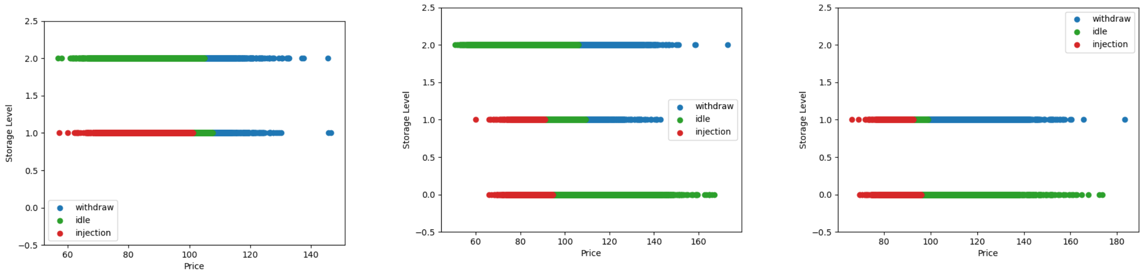
On Figures 5 and 6, we give an example of the control obtained in each regime in simulation. At each date the control with the highest probability is taken in simulation. On Figure 7, we provide the optimal controls obtained by dynamic programming at date 25. Comparing Figures 6 and 7, we see that the control is very well calculated by control randomization method.



Injection regime   Idle regime   Withdrawal regime

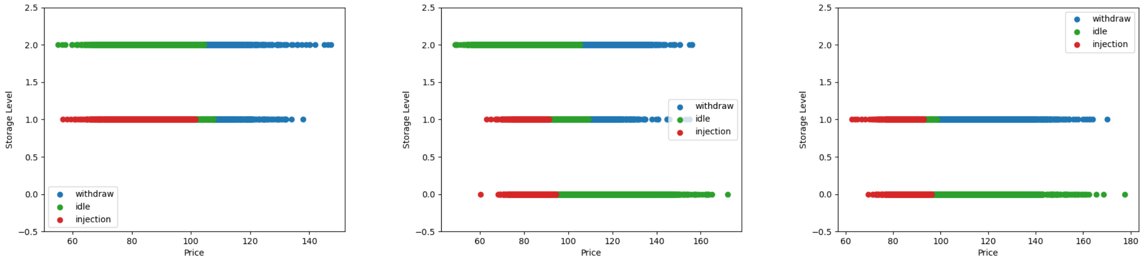
Figure 5: Control obtained at date 12 in the battery test case depending on the regime.

programminng



Injection regime   Idle regime   Withdrawal regime

Figure 6: Control obtained at date 25 in the battery test case depending on the regime.



Injection regime   Idle regime   Withdrawal regime

Figure 7: Control obtained by dynamic programming at date 25 in the battery test case depending on the regime.

5. Conclusion

We present a uniform framework for a broad class of stochastic control problems based on their corresponding randomised control problems. We first develop a policy gradient formula for randomised control problems, and, based on it, an actor-critic algorithm. The advantage of considering randomised control problems is the natural discretisation they provide, which avoids the need for subsequent approximations. We note that even for deterministic grids, our approach results in a different policy gradient formula compared to those in [16, 11, 17].

We demonstrate heuristically how to recover the optimal (non-randomised) control and develop an actor-critic algorithm for the original stochastic control problems based on their randomised formulation. By considering fixed sampling intensities – taking the role of the discretisation step size in our framework – we provide a flexible and implementable actor-critic algorithm that supports various sampling grid types, including deterministic grids, random discrete grids, and Poisson grids. We conduct numerical experiments to validate our framework, applying it to two optimal switching problems focusing on real options in the energy sector. We believe that while randomised grids pose additional challenges in terms of stability and implementation, they may be useful for off-policy learning for continuous-time problems with unevenly spaced (sampling) time series.

A. Control randomisation for regular control problems and optimal stopping problems

A.1. Regular control problems

In this subsection, we will recall the construction of the randomised problems for regular control problems based on the works of [19, 9]. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, W an m-dimensional Brownian motion and $\mathbb{F}^W$ the filtration generated by W. Further let A be a compact Polish space and admissible control set $\mathcal{A}$ the set of all $\mathbb{F}^W$-progressive A-valued processes. We consider a regular control problem in continuous time with the state dynamics

$$dX_s^{t,x,u} = b(s, X_s^{t,x,u}, u_s)ds + \sigma(s, X_s^{t,x,u}, u_s)dW_s, \quad X_t = x,$$

and the reward functional

$$\mathrm{J}(t, x, u) := \mathbb{E}\left[g(X_T^{t,x,u}) + \int_t^T f(s, X_s^{t,x,u}, u_s)ds\right],$$

and the value function

$$V(t, x) := \sup_{u \in \mathcal{A}} \mathrm{J}(t, x, u).$$

We will make the standard assumptions to ensure the optimisation problem is well-defined:

Assumption C. *(C1) The coefficients b and σ are continuous and there exists a positive constant C s.t. for all $t \in [0, T]$, $x, x' \in \mathbb{R}^d$, $a, a' \in A$,*

$$|b(t, x, a) - b(t, x', a)| + |\sigma(t, x, a) - \sigma(t, x', a)| \le C|x - x'|,$$
$$|b(t, 0, a)| + |\sigma(t, 0, a)| \le C.$$

(C2) The reward functions f and g are continuous and polynomially bounded in x, that is there exists some positive constants C and p such that for all $t \in [0, T]$, $x \in \mathbb{R}^d$, $a \in A$,

$$|f(t, x, a)| + |g(x)| \leq C(1 + |x|^p).$$

To bring this into the form of Section 2, we follow the randomisation method and introduce an independent Poisson point process ν on $(0, T] \times A$ with predictable projection $\hat{\nu}(ds, de)$ and let $I^{t,a}$ be the the corresponding A-valued process given by

$$I_s^{t,a} = a + \int_{(t,s]} \int_A (e - I_{r-}^{t,a}) \nu(dr, de), \qquad s \in [t, T],$$

for some fixed $a \in A$. The state process $X^{t,x,a}$ is given by the following uncontrolled dynamics where we replace our control u by the process $I^{t,a}$, again in the dynamics formulation,

$$dX_s^{t,x,a} = b(s, X_s^{t,x,a}, I_s^{t,a})ds + \sigma(s, X_s^{t,x,a}, I_s^{t,a})dW_s, \quad X_t = x, I_{t-}^{t,a} = a.$$

We now take the set $\mathcal{V}$ of all $Pred(\mathbb{F}^{W,\mu}) \otimes \mathcal{B}(A)$-measurable, essentially bounded process λ, where with slight abuse of notation we denote by $\mathbb{P}^\lambda \ll \mathbb{P}$ the probability measure under which ν is a Poisson random measure with intensity $\lambda_s(e)\hat{\nu}(ds, de)$, as the new action set. The corresponding randomised reward functional has the form

$$J(t, x, a, \lambda) \coloneqq \mathbb{E}^\lambda \left[g(X_T^{t,x,u}) + \int_t^T f(s, X_s^{t,x,u}, I_s^{t,a}) ds \right],$$

and we define the randomised value function as

$$V^{\mathcal{R}}(t, x, a) \coloneqq \sup_{\lambda \in \mathcal{V}} J(t, x, a, \lambda).$$

We note that Fuhrman and Pham [9] studied a more general class of path-dependent regular control problems, though we focus here on the Markovian setting. [9] considered the case where $I^{t,a}$ is a homogeneous marked Poisson process and proved the following equivalence between the regular control problem and the randomised problem.

Theorem A.1 (see also [9, Theorem 3.2])**.** *Let Assumption C hold and let ν have a predictable projection of the form $\hat{\nu}(ds, de) = \rho(de)ds$, where $\rho(de)$ is a finite measure on A with full support. Then the value functions of both problems coincide, that is $V = V^{\mathcal{R}}$, and in particular, $V^{\mathcal{R}}$ does not depend on $a \in A$.*

A.2. Optimal stopping problems

In this section we will review how to construct randomised control problems for optimal stopping problems. While [10] studied these problems in a general, potentially non-Markovian setting, we will focus on the simpler Markovian case here. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space and W be an m-dimensional Brownian motion with $\mathbb{F}^W$ being its generated filtration. For optimal stopping problems we are given state dynamics of the form

$$dX_s^{t,x} = b(s, X_s^{t,x})ds + \sigma(s, X_s^{t,x})dW_s, \qquad X_t^{t,x} = x,$$

and want to maximise the reward functional

$$\mathrm{J}(t,x,\tau) := \mathbb{E}\left[1_{\{\tau<T\}} g(X_\tau^{t,x}) + \mathbb{1}_{\{\tau=T\}} h(X_T^{t,x}) + \int_0^\tau f(s, X_s^{t,x})ds\right]$$

over the set $\mathcal{T}$ of $\mathbb{F}^W$-stopping times τ with $t \le \tau \le T$. The value function is then given by

$$V(t,x) := \sup_{\tau\in\mathcal{T}} \mathrm{J}(t,x,\tau).$$

We make the following standard assumptions:

Assumption D. *(D1) The coefficients b and σ are continuous and there exists a positive constant C s.t. for all $t \in [0,T]$, $x, x' \in \mathbb{R}^d$,*

$$|b(t,x) - b(t,x')| + |\sigma(t,x) - \sigma(t,x')| \le C|x - x'|.$$

(D2) The reward functions f, g and h are continuous and satisfy a quadratic growth condition in x, that is there exists some positive constant C such that for all $t \in [0,T]$, $x \in \mathbb{R}^d$,

$$|f(t,x)| + |g(x)| + |h(x)| \le C(1 + |x|^2).$$

For the construction of the randomised control problem, we in principle follow [10], however modifying the presentation slightly to better fit the unified setting described in Section 2. Let I^t be a Poisson point process with intensity $\mathbb{1}_{\{I^t_{s-}=0\}}ds = (1 - I^t_{s-})ds$ and $I^t_{t-} = 0$. Note that I^t has at most one jump on $[t,T]$, which will take the role of our stopping time $\tau^{I^t} := \inf\{s \in (t,T] | I^t_{s-} \neq I^t_s\} \wedge T$ in the optimal stopping problem. Further, instead of $X^{t,x}$, we will consider the (stopped) dynamics

$$d\hat{X}_s^{t,x} = \mathbb{1}_{\{I^t_{s-}=0\}} b(s, \hat{X}_s^{t,x})ds + \mathbb{1}_{\{I^t_{s-}=0\}} \sigma(s, \hat{X}_s^{t,x})dW_s, \qquad \hat{X}_t^{t,x} = x.$$

We note that $\hat{X}_T^{t,x} = X^{t,x}_{\tau^{I^t}}$. The randomised problem involves taking the supremum over the set $\mathcal{V}$ of all $\mathbb{F}^{W,I^t}$-predictable, essentially bounded processes λ, where $\mathbb{P}^\lambda$ denotes the probability measure given by

$$\left.\frac{d\mathbb{P}^\lambda}{d\mathbb{P}}\right|_{\mathcal{F}_s^{W,I^t}} := \exp\left(\int_{(0,s]} \log \lambda_r dI^t_r - \int_{(0,s]} (\lambda_r - 1)\mathbb{1}_{\{I^t_{r-}=0\}}dr\right), \quad s \in [t,T],$$

under which I^t is a Poisson point process with intensity $\mathbb{1}_{\{I^t_{s-}=0\}}\lambda_s ds$. The corresponding reward functional is given by

$$J(t,x,\lambda) := \mathbb{E}^\lambda\left[\mathbb{1}_{\{I^t_{T-}=1\}} h(\hat{X}_T^{t,x}) + \mathbb{1}_{\{I^t_{T-}=0\}} g(\hat{X}_T^{t,x}) + \int_t^T \mathbb{1}_{\{I^t_{s-}=0\}} f(s, \hat{X}_s^{t,x})ds\right],$$

resulting in the following randomised value function

$$V^{\mathcal{R}}(t,x) = \sup_{\lambda\in\mathcal{V}} J(t,x,\lambda).$$

The following equivalence result between the randomised and non-randomised problem is proved in [10].

Theorem A.2 ([10, Corollary 2.4]). *Let Assumption D hold. Then the value functions of both problems coincide, that is $V = V^{\mathcal{R}}$.*

References

[1] Christian Bender and Tran Thuan Nguyen. Entropy-regularized mean-variance portfolio optimization with jumps. *arXiv:2312.13409*, 2023.

[2] Bruno Bouchard. A stochastic target formulation for optimal switching problems in finite horizon. *Stochastics*, 81(2):171–197, 2009.

[3] Bruno Bouchard and Xavier Warin. Monte-carlo valuation of american options: Facts and new algorithms to improve existing methods. In *Numerical Methods in Finance: Bordeaux, June 2010*, pages 215–255. Springer, 2012.

[4] René Carmona and Michael Ludkovski. Valuation of energy storage: an optimal switching approach. *Quantitative Finance*, 10(4):359–374, 2010.

[5] Min Dai, Yu Sun, Zuo Quan Xu, and Xun Yu Zhou. Learning to optimally stop diffusion processes, with financial applications. *arXiv:2408.09242*, 2024.

[6] Jodi Dianetti, Giorgio Ferrari, and Renyuan Xu. Exploratory optimal stopping: A singular control formulation. *arXiv:2408.09335*, 2024.

[7] Yuchao Dong. Randomized optimal stopping problem in continuous time and reinforcement learning algorithm. *SIAM Journal on Control and Optimization*, 62(3):1590–1614, 2024.

[8] Romuald Elie and Idris Kharroubi. Probabilistic representation and approximation for coupled systems of variational inequalities. *Statistics & Probability Letters*, 80(17):1388–1396, 2010.

[9] Marco Fuhrman and Huyên Pham. Randomized and backward SDE representation for optimal control of non-Markovian SDEs. *The Annals of Applied Probability*, 25(4):2134–2167, 2015.

[10] Marco Fuhrman, Huyên Pham, and Federica Zeni. Representation of non-Markovian optimal stopping problems by constrained BSDEs with a single jump. *Electronic Communications in Probability*, 21:1–7, 2016.

[11] Xuefeng Gao, Lingfei Li, and Xun Yu Zhou. Reinforcement learning for jump-diffusions, with financial applications. *arXiv:2405.16449*, 2024.

[12] Hugo Gevret, Nicolas Langrené, Jerome Lelong, Xavier Warin, and Aditya Maheshwari. *STochastic OPTimization library in C++*. PhD thesis, EDF Lab, 2018.

[13] Said Hamadène and Monique Jeanblanc. On the starting and stopping problem: Application in reversible investments. *Mathematics of Operations Research*, 32(1):182–192, 2007.

[14] Jean Jacod. Multivariate point processes: predictable projection, Radon-Nikodym derivatives, representation of martingales. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 31(3):235–253, September 1975.

[15] Yanwei Jia and Xun Yu Zhou. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *Journal of Machine Learning Research*, 23(154):1–55, 2022.

[16] Yanwei Jia and Xun Yu Zhou. Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *Journal of Machine Learning Research*, 23(275):1–50, 2022.

[17] Yanwei Jia and Xun Yu Zhou. q-learning in continuous time. *Journal of Machine Learning Research*, 24(161):1–61, 2023.

[18] Idris Kharroubi, Jin Ma, Huyên Pham, and Jianfeng Zhang. Backward SDEs with constrained jumps and quasi-variational inequalities. *The Annals of Probability*, 38(2):794–840, 2010.

[19] Idris Kharroubi and Huyên Pham. Feynman-Kac representation for Hamilton-Jacobi–Bellman IPDE. *The Annals of Probability*, 43(4):1823–1865, July 2015.

[20] Huiling Meng, Ningyuan Chen, and Xuefeng Gao. Reinforcement learning for intensity control: An application to choice-based network revenue management. *arXiv:2406.05358*, 2024.

[21] Huyên Pham and Xavier Warin. Actor critic learning algorithms for mean-field control with moment neural networks. *arXiv:2309.04317*, 2023.

[22] Arnaud Porchet, Nizar Touzi, and Xavier Warin. Valuation of power plants by utility indifference and numerical computation. *Mathematical Methods of Operations Research*, 70:47–75, 2009.

[23] Haoran Wang, Xun Yu Zhou, and Thaleia Zariphopoulou. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21(198):1–34, 2020.

[24] Xavier Warin. Reservoir optimization and machine learning methods. *EURO Journal on Computational Optimization*, 11:100068, 2023.

Funding

The authors declare that no funds, grants, or other supports were received during the preparation of this manuscript.

Conflict of interest

The authors have no relevant or non-financial interests to disclose.