

Forward-Backward Rapidly-Exploring Random Trees for Stochastic Optimal Control

Kelsey P. Hawkins, Ali Pakniyat, Evangelos Theodorou, Panagiotis Tsiotras

Abstract—We propose a numerical method for the computation of the forward-backward stochastic differential equations (FBSDE) appearing in the Feynman-Kac representation of the value function in stochastic optimal control problems. By the use of the Girsanov change of probability measures, it is demonstrated how a rapidly-exploring random tree (RRT) method can be utilized for the forward integration pass, as long as the controlled drift terms are appropriately compensated in the backward integration pass. Subsequently, a numerical approximation of the value function is proposed by solving a series of function approximation problems backwards in time along the edges of the constructed RRT. Moreover, a local entropy-weighted least squares Monte Carlo (LSMC) method is developed to concentrate function approximation accuracy in regions most likely to be visited by optimally controlled trajectories. The results of the proposed methodology are demonstrated on linear and nonlinear stochastic optimal control problems with non-quadratic running costs, which reveal significant convergence improvements over previous FBSDE-based numerical solution methods.

I. INTRODUCTION

The Feynman-Kac representation theory and its associated forward-backward stochastic differential equations (FBSDEs) has been gaining traction as a framework to solve nonlinear stochastic control problems, including optimal control problems with quadratic cost [1], minimum-fuel (L_1 -running cost) problems [2], [3], differential games [4], and reachability problems [1], [5]. FBSDE-based numerical methods have also received interest from the mathematical finance community [6], [7], [8]. Although initial results demonstrate promise in terms of flexibility and theoretical validity, numerical algorithms which leverage this theory have not yet matured. For even modest problems, state-of-the-art algorithms often have issues with slow and unstable convergence to the optimal policy. Producing more robust numerical methods is critical for the broader adoption of FBSDE methods for real-world tasks.

FBSDE numerical solution methods broadly consist of two steps, a forward pass, which generates Monte Carlo samples of the forward stochastic process, and a backward pass, which iteratively approximates the value function backwards in time. Typically, FBSDE methods perform this approximation using a least-squares Monte Carlo (LSMC) scheme, which implicitly solves the backward SDE with parametric function approximation [7]. The approximate value function fit in the backward pass is then often used to improve

sampling in an updated forward pass, leading to an iterative algorithm which, ideally, improves the approximation till convergence. Although FBSDE methods share a distinct similarity to differential dynamic programming (DDP) techniques [9], [10], [11], DDP is generally less flexible. For most DDP applications, a strictly positive definite running cost with respect to the control is required for convergence [12, Section 2.2.3]. Furthermore, in DDP, the computation of first and second order derivatives of both dynamics and costs is necessary for the backward pass, making it challenging to apply this approach to problems where these derivatives are not known analytically. In contrast, FBSDE techniques only require a good fit of the value function and the evaluation of the gradient of this value function to obtain the optimal control.

The flexibility of Feynman-Kac-based FBSDE algorithms stems from the intrinsic relationship between the solution of a broad class of second-order parabolic or elliptic PDEs to the solution of FBSDEs (see, e.g., [13, Chapter 7]), brought to prominence in [14], [15], [16]. Both Hamilton-Jacobi-Bellman (HJB) and Hamilton-Jacobi-Isaacs (HJI) second order PDEs, utilized for solving stochastic optimal control and stochastic differential game equations respectively, can thus be solved via FBSDE methods, even when the costs and dynamics are nonlinear. This provides an alternative to the direct solution of PDEs, typically solved using grid-based methods such as the Level Set Toolbox [17], known for poor scaling in high dimensional state spaces ($n \geq 4$).

The primary advantage of Feynman-Kac-based FBSDE methods is that they produce an unbiased estimator for the value function associated with the HJB equations. However, a naïve application of the theory leads to estimators with high variance by producing sample trajectories away from the optimal ones. Recent work has shown that Girsanov's theorem can be used to change the sampling measure of the forward pass without adding intrinsic bias to the estimator [1], [2], [3]. That is, a change over probability spaces corresponds to the introduction of a drift to the forward SDE that can be employed to modify the sampling in the forward pass; this, in turn, requires appropriate accommodation of the change of measures in the backward pass.

In this work we expand upon the above results, by showing that the forward sampling measure can be modified at will, which enables us to incorporate methods from other domains, namely, rapidly-exploring random trees (RRTs) (see, e.g., [18] and the recent survey in [19]), in order to more efficiently explore the state space in the forward pass. RRTs are frequently applied to reachability-type motion

All authors are with the Georgia Institute of Technology, Atlanta, Georgia 30332-0250. Contact at kphawkins, pakniyat, evangelos.theodorou, tsiotras@gatech.edu

Support for this work has been provided by NSF award IIS-2008686.

planning problems, biasing the samples towards regions of the state space that have low density. Using RRTs in the forward sampling allows us to spread samples evenly over the reachable state space, increasing the likelihood that near-optimal samples are well-represented in the forward pass sample distribution. By sampling more efficiently and relying less on incremental approximations of the value function to guide our search, we can achieve faster and more robust convergence than previous FBSDE methods. In the backward pass, we take advantage of the path-integrated running costs and estimates of the value function to produce a heuristic which weighs paths in the function approximation according to a local-entropy measure-theoretic optimization. Although local-entropy path integral theory and RRTs have been used together in [20], called PI-RRT, this method is more closely related to the path-integral approach to control [10]. Our method similarly performs forward passes to broadly sample the state space, but follows them with backward passes to obtain approximations for the value functions, and consequently obtain closed loop policies over the full horizon.

The primary contributions of this paper are as follows:

- Providing the theoretical basis for the use of McKean-Markov branched sampling in the forward pass of FBSDE techniques.
- Introducing an RRT-inspired algorithm for sampling the forward SDE.
- Presenting a technique for concentrating value function approximation accuracy in regions containing optimal trajectories.
- Proposing an iterative numerical method for the purpose of approximating the optimal value function and its policy.

We call the proposed method forward-backward rapidly exploring random trees (FBRRT). After we describe the approach in both theory and numerical implementation, we apply FBRRT to two problems, comparing it to [2], and demonstrating its ability to solve nonlinear stochastic optimal control problems with non-quadratic running costs.

II. THE HAMILTON-JACOBI EQUATION AND ON-POLICY VALUE FUNCTION

Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \in [0, T]}, \mathbb{Q})$, be a complete, filtered probability space, on which $W_s^{\mathbb{Q}}$ is an n -dimensional standard Brownian (Wiener) process with respect to the probability measure \mathbb{Q} and adapted to the filtration $\{\mathcal{F}_t\}_{t \in [0, T]}$. Consider a stochastic system whose dynamics are governed by

$$dX_s = f(s, X_s, u_s) ds + \sigma(s, X_s) dW_s^{\mathbb{Q}}, \quad X_0 = x_0, \quad (1)$$

where X_s is a \mathcal{F}_s -progressively measurable state process on the interval $s \in [0, T]$, taking values in \mathbb{R}^n , $u_{[0, T]}$ is a progressively measurable input process on the same interval, taking values in the compact set $U \subseteq \mathbb{R}^m$, and $f : [0, T] \times \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$, $\sigma : [0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ are the Markovian drift and diffusion functions respectively. The

cost associated with a given control signal is

$$S_t[u_{[t, T]}] := \int_t^T \ell(s, X_s, u_s) ds + g(X_T), \quad (2)$$

where $\ell : [0, T] \times \mathbb{R}^n \times U \rightarrow \mathbb{R}^+$ is the running cost, and $g : \mathbb{R}^n \rightarrow \mathbb{R}^+$ is the terminal cost. Let membership of a function in $C_b^{l, k}$ denote that the function and its partial derivatives in t of order $\leq l$ and in x of order $\leq k$ are continuous and bounded on the domain. The membership in C_b^k is defined similarly. We assume the functions f , σ , $a := \sigma\sigma^\top$ and ℓ belong to $C_b^{1, 2}$, that $g \in C_b^3$, and that σ^{-1} exists and is uniformly bounded on its domain.

The stochastic optimal control (SOC) problem is to determine the value function $V^* : [0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^+$ defined as

$$V^*(t, x) = \inf_{u_{[t, T]}} \mathbf{E}_{\mathbb{Q}}^{t, x}[S_t[u_{[t, T]}]], \quad (\text{SOC})$$

where $\mathbf{E}_{\mathbb{Q}}^{t, x}[\cdot] := \mathbf{E}_{\mathbb{Q}}[\cdot | X_t = x]$ denotes the conditional expectation given $X_t = x_t$ under the probability measure \mathbb{Q} .

Under mild regularity assumptions, in particular that $\sigma\sigma^\top$ is uniformly positive definite, there exists a unique classical solution $V^* \in C_b^{1, 2}$ to the Hamilton-Jacobi-Bellman PDE, as well as a (not necessarily unique) optimal Markov control policy π^* , which satisfies the inclusion

$$\pi^*(s, x) \in \arg \min_{u \in U} \{\ell(s, x, u) + f(s, x, u)^\top \partial_x V^*(s, x)\}, \quad (3)$$

with the property that $V^*(t, x) = \mathbf{E}_{\mathbb{Q}}^{t, x}[S_t[\pi^*]]$, where $\partial_x V^*$ is the partial derivative of V^* with respect to state x [21, Chapter 4, Theorems 4.2 and 4.4, and Chapter 6, Theorem 6.2].

In this paper, instead of a direct solution of the HJB PDE, we work with a class of generic Markov policies $\mu : [0, T] \times \mathbb{R}^n \rightarrow U$ and their associated value functions V^μ , and use iterative methods to approximate V^* and π^* . The on-policy value function is defined as

$$V^\mu(t, x) = \mathbf{E}_{\mathbb{Q}}^{t, x}[S_t^\mu], \quad (4)$$

$$S_t^\mu := \int_t^T \ell_s^\mu ds + g(X_T),$$

with the process X_s satisfying the forward SDE (FSDE)

$$dX_s = f_s^\mu ds + \sigma_s dW_s^{\mathbb{Q}}, \quad X_t = x, \quad (5)$$

where, for brevity of exposition, we define

$$f_s^\mu := f(s, X_s, \mu(s, X_s)),$$

and similarly for ℓ , σ . We call μ an admissible Markov policy if it is Borel-measurable and its associated V^μ is the unique classic solution to the Hamilton-Jacobi PDE

$$\partial_t V^\mu + \frac{1}{2} \text{tr}[\sigma\sigma^\top \partial_{xx} V^\mu] + (\partial_x V^\mu)^\top f^\mu + \ell^\mu|_{t, x} = 0, \quad (HJ)$$

$$V^\mu(T, x) = g(x),$$

for $(t, x) \in [0, T) \times \mathbb{R}^n$, where ∂_t and ∂_x are the partial derivative operators in t and x , and ∂_{xx} is the Hessian in x .

Hence, the optimal control problem is expressed as $V^* = \min V^\mu$ over all μ such that (HJ) holds. Since the boundedness of σ^{-1} makes the PDE non-degenerate parabolic, a sufficient, but not necessarily tight, condition guaranteeing existence of the classical solution is if f^μ and ℓ^μ are in $C_b^{1,2}$ [22, p. 156; Chapter 3, Theorem 4.2, Theorem 4.4]. The same reference guarantees that $V^* \equiv V^{\pi^*}$.

III. FEYNMAN-KAC-GIRSANOV FBSDE REPRESENTATION

A. On-Policy FBSDEs

The positivity of $\sigma\sigma^\top$ yields that (HJ) is a parabolic PDE and, hence, by the Feynman-Kac Theorem (see, e.g. [23]) it is linked to the solution (X_s, Y_s, Z_s) of the pair of FBSDEs composed of the FSDE (5) and the backward SDE (BSDE)

$$dY_s = -\ell_s^\mu ds + Z_s^\top dW_s^Q, \quad Y_T = g(X_T), \quad (6)$$

where Y_s and Z_s are, respectively, 1 and n -dimensional adapted processes.

Theorem 3.1 (Feynman-Kac Representation): For the solution (X_s, Y_s, Z_s) to the FBSDE characterized by (5) and (6), it holds that

$$\begin{aligned} Y_s &= V^\mu(s, X_s), & s \in [0, T], \\ Z_s &= \sigma_s^\top \partial_x V^\mu(s, X_s), & \text{a.e. } s \in [0, T], \end{aligned} \quad (7)$$

Q-almost surely (a.s.), and, in particular,

$$Y_t = \mathbf{E}_Q[\hat{Y}_{t,\tau}|X_t] = V^\mu(t, X_t), \quad \text{Q-a.s.}, \quad (8)$$

for $0 \leq t \leq \tau \leq T$ where

$$\hat{Y}_{t,\tau} := Y_\tau + \int_t^\tau \ell_s^\mu ds. \quad (9)$$

□

Proof: Equations (7) are due directly to [13, Chapter 7, Theorem 4.5, (4.29)]. From the definition of Itô integrals, we have

$$\hat{Y}_{t,\tau} = Y_t - \int_t^\tau Z_s^\top dW_s^Q. \quad (10)$$

Taking the conditional expectation of both sides yields

$$\mathbf{E}_Q[\hat{Y}_{t,\tau}|\mathcal{F}_t] = \mathbf{E}_Q[Y_t|\mathcal{F}_t] = Y_t, \quad (11)$$

noting that the last term drops out due to the property of the Itô integral [13, p. 34, (5.26)], and Y_t passes through the conditional expectation because it is \mathcal{F}_t measurable. Equation (9) is a direct consequence of the definition of Itô integrals, [13, p. 33, (5.23)]. ■

B. Off-Policy FBSDEs

Consider, contrary to the on-policy FBSDEs, the off-policy drifted FBSDEs

$$dX_s = K_s ds + \sigma_s dW_s^P, \quad X_0 = x_0, \quad (12)$$

$$dY_s = -(\ell_s^\mu + Z_s^\top D_s) ds + Z_s^\top dW_s^P, \quad Y_T = g(X_T), \quad (13)$$

with

$$D_s := \sigma_s^{-1}(f_s^\mu - K_s), \quad (14)$$

where K_s , an arbitrary \mathcal{F}_s -progressively measurable and bounded process satisfying the smoothness conditions of [13, Chapter 1, Theorem 6.16], P the new probability measure associated with K_s and W_s^P a Brownian process over the new, complete, filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \in [0, T]}, P)$.

Theorem 3.2: For the solution (X_s, Y_s, Z_s) to the FBSDE characterized by (12) and (13), it holds that

$$\begin{aligned} Y_s &= V^\mu(s, X_s), & s \in [0, T], \\ Z_s &= \sigma_s^\top \partial_x V^\mu(s, X_s), & \text{a.e. } s \in [0, T], \end{aligned} \quad (15)$$

P-a.s., and in particular,

$$Y_t = \mathbf{E}_P[\hat{Y}_{t,\tau}|X_t] = V^\mu(t, X_t), \quad \text{P-a.s.}, \quad (16)$$

where

$$\hat{Y}_{t,\tau} := Y_\tau + \int_t^\tau (\ell_s^\mu + Z_s^\top D_s) ds. \quad (17)$$

□

Proof: Apply Girsanov's theorem to both (12) and (13), where the Brownian process W_s^Q is defined as $dW_s^Q := dW_s^P - D_s ds$ and the Radon-Nikodym derivative is defined according to [21, Chapter 5, Theorem 10.1]. Further, the theorem guarantees that P and Q are equivalent measures in a measure-theoretic sense. Since (7) holds Q-a.s., there exists an $N \in \mathcal{F}$, $Q(N) = 0$, such that $E^C \subseteq N$, where $E := \{\omega \in \Omega : (7) \text{ holds}\}$. It subsequently follows from the definition of absolute continuity that $P(N) = 0$, so (7) holds P-a.s. as well. The rest follows similarly to Theorem 3.1 ■

We can interpret this result in the following sense. As long as the diffusion function σ is the same as in the on-policy formulation, we can pick an arbitrary process K_s to be the drift term which generates a distribution for the forward process X_s in the corresponding measure P . The BSDE yields an expression for Y_t using the same process W_s^P as used in the FSDE. The term $Z_s^\top D_s$ acts as a correction in the BSDE to compensate for changing the drift of the FSDE. We can then use the relationship (16) to solve for the value function V^μ , whose conditional expectation can be evaluated in P . Although used in the analytic construction of the value function, the measure Q does not require approximation to solve for the value function.

It should be highlighted that K_s need not be a deterministic function of the random variable X_s , as is the case with f_s^μ . For instance, it can be selected as the function $K_s(\omega) = h(s, X_s(\omega), \omega)$ for some appropriate function h , producing a non-trivial joint distribution for the random variables (X_t, K_t) .

A remarkable feature of both the on- and off-policy FBSDEs is that the forward pass is decoupled from the backward pass, that is, the evolution of the forward SDE does not explicitly depend on Y_s or Z_s (whereas in the Stochastic Maximum Principle formulations (see, e.g., [13, Chapter 3]) the decoupling is irremovable). This feature forms the basis of FBSDE numerical investigations of stochastic optimal

control [24], [1], but the significant difference of Theorem 3.1 in comparison to those results is that the focus is shifted here from the solution of the HJB equation towards the broader class of functions satisfying the (HJ). This provides a stronger case for policy iteration methodologies, because the theory does not require or expect μ to be an optimal policy, as is in [24], [1]. Although not evaluated in this work, μ can be chosen according to design specifications other than estimating the optimal policy, such as to ensure the current policy is close to the previously estimated policy.

C. Local Entropy Weighing

As discussed in Section III-B, the disentanglement of the forward sampling from the backward function approximation provides the opportunity to employ broad sampling schemes to cover the state space with potential paths. However, fitting a value function broadly to a wide support distribution might degrade the quality of the function approximation since high accuracy of function approximation is more in demand in those parts of the state space in proximity to optimal trajectories. Once forward sampling has been performed and some parts of the value function have been approximated, we can begin forming a heuristic in which sample paths closer to optimal trajectories are weighted more to concentrate value function approximation accuracy in those regions.

To this end, we propose using a bounded heuristic random variable ρ_t to produce a new measure R_t , the weighted counterpart to P_t , where the subscript refers to the restriction of P to \mathcal{F}_t . In order to avoid underdetermination of the regression by concentration over a single or few samples, we select R_t as

$$R_t \in \arg \min_{R_t} \{ \mathbf{E}_{R_t}[\rho_t] + \lambda \mathcal{H}(R_t \| P_t) \}, \quad (18)$$

with $\lambda > 0$ a tuning variable and

$$\mathcal{H}(R_t \| P_t) = \mathbf{E}_{R_t} \left[\log \left(\frac{dR_t}{dP_t} \right) \right], \quad (19)$$

is the relative entropy of R_t which takes its minimum value when $R_t = P_t$, the distribution in which all sampled paths have equal weight.

The minimizer (18), which balances between the value of ρ and the relative entropy of its induced measure, has a solution of R_t^* determined [25, p. 2] as

$$dR_t^* = \Theta_t dP_t, \quad (20)$$

$$\Theta_t := \frac{\exp(-1/\lambda \rho_t)}{\mathbf{E}_{P_t}[\exp(-1/\lambda \rho_t)]}. \quad (21)$$

Henceforth, we let R_t refer to this minimizer R_t^* . In the numerical approximation of this heuristic we can interpret the weights as a *softmax* operation over paths according to the heuristic, a method often used in deep learning literature [26].

Theorem 3.3: Assume ρ_τ is selected such that W_s^P is Brownian on the interval $[t, \tau]$ in the induced measure R_τ . It holds that

$$Y_t = \mathbf{E}_{P_\tau}[\hat{Y}_{t,\tau} | X_t] = V^\mu(t, X_t), \quad R_\tau\text{-a.s.}, \quad (22)$$

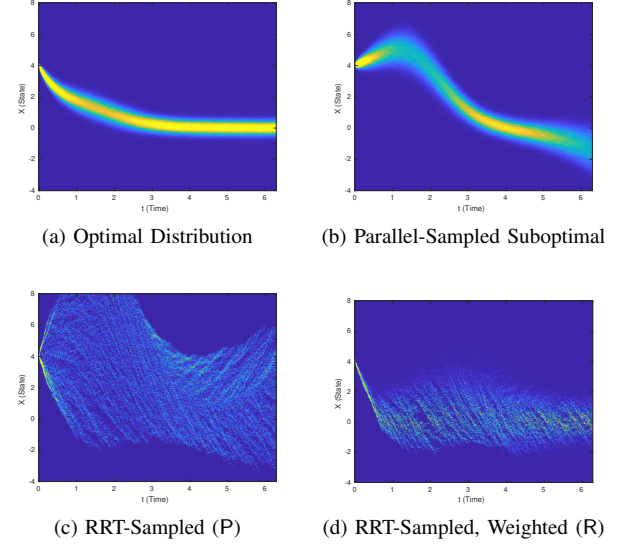


Fig. 1: Heatmap of different measure distributions for a 1-dimensional SOC problem, illustrating how RRT-sampling and local-entropy weighing can accelerate discovery of the optimal distribution.

where $\hat{Y}_{t,\tau}$ is defined in (17). Furthermore, the minimizer ϕ^* of the optimization

$$\begin{aligned} & \inf_{\phi \in L_2} \mathbf{E}_{R_\tau}[(\hat{Y}_{t,\tau} - \phi(X_t))^2] \\ &= \inf_{\phi \in L_2} \mathbf{E}_{P_\tau}[\Theta_\tau^{\text{RIP}}(\hat{Y}_{t,\tau} - \phi(X_t))^2], \end{aligned} \quad (23)$$

over X_t -measurable square integrable variables $\phi(X_t)$ coincides with the value function $\phi^*(X_t) = V^\mu(t, X_t)$. \square

Proof: First, note that X_s, Y_s, Z_s , and $\hat{Y}_{t,\tau}$ are \mathcal{F}_τ -measurable for $s, t \in [0, \tau]$. Thus, restricting P to \mathcal{F}_τ in Theorem 3.2, producing P_τ , results in the same assumptions for those variables. Since ρ_τ is bounded, $\Theta_\tau > 0$ P-a.s.. Further, we have $\mathbf{E}_{P_\tau}[\Theta_\tau] = 1$, so the variable is normalized. It is easy to see that this guarantees that R_τ is a probability measure and the measures R_τ and P_τ are equivalent. It follows that (22) holds. Equation (23) is a result of the L_2 -projective properties of conditional expectation [27] and then a change of measure with (20). \blacksquare

In the following section, we evaluate the minimization of the right hand side of (23) over parameterized value function models to obtain an estimate of the value function.

To summarize, in this section we introduced three measures, (a) Q , the measure associated with the target policy μ for the value function V^μ , (b) P , the sampling measure used in the forward pass to explore the state space, and (c) R_τ , the local-entropy weighted measure used in the backward pass to control function approximation accuracy. Fig. 1 illustrates how these results work together to rapidly discover the optimal distribution. An on-policy method assumes the knowledge of an initial suboptimal control policy, sampled as represented in Fig. 1 (b), and the suboptimal value function is solved in that distribution. This method requires iterative

improvement of the policy to produce a distribution which overlaps with the optimal distribution. However, if we begin with a sampling measure which broadly explores the state space as in Fig. 1 (c), we can produce an informed heuristic which weighs this distribution as in Fig. 1 (d), so that the function approximation is concentrated in a near-optimal distribution. These results leave open the choice for a target policy μ that produces Q , the drift process K_s that produces P and the weighing function ρ_τ that produces R_τ . In the following section we propose particular choices for each.

IV. FORWARD-BACKWARD RRT

In this section, we introduce a numerical method that leverages the continuous-time theory of the previous section. We begin by discussing a generalized approach to approximating the sampling distribution P with a branch-sampling representation. Next, we introduce FBRRT, an iterative algorithm for solving the SOC problem. We then propose an RRT-inspired algorithm that leverages the previous theory. Finally, we propose a heuristic variable ρ for weighing paths.

A. McKean-Markov Branched Sampling

We approximate the continuous-time sampling distributions with discrete-time McKean-Markov branch sampled paths as presented in [28]. First, for a given Δt , the interval $[0, T]$ is partitioned according to the time steps $(t_0 = 0, \dots, t_i = (\Delta t)i, \dots, t_N = T)$. For brevity, we abbreviate X_{t_i} as X_i and similarly for most variables.

In the forward sampling process, we produce a series of path measures $\{\vec{P}_i\}_{i=0}^N$,

$$\vec{P}_i := \frac{1}{M} \sum_{j=1}^M \delta_{\xi_i^j}, \quad (24)$$

where δ is the Dirac-delta measure acting on sample paths

$$\xi_i^j := (x_{0,i}^j, k_{0,i}^j, x_{1,i}^j, k_{1,i}^j, \dots, k_{i-1,i}^j, x_{i,i}^j), \quad (25)$$

with $x_{j,i}^j, k_{j,i}^j \in \mathbb{R}^n$. The path notation $x_{j,i}^j$ indicates that this element is the sample of random variable X_j that is the ancestor of sample $x_{i,i}^j$ in path ξ_i^j . Fig. 2 (b) illustrates how these measures are represented using a tree data structure. Each node in the tree $x_{i,i}^j$, alternatively called a particle, is associated with a path ξ_i^j whose final term is $x_{i,i}^j = x_i^j$.

The edges in the tree represent an Euler-Maruyama SDE step approximation of the forward SDE (12). When a node in the tree at time i is selected for expansion, it becomes the $x_{i,i+1}^j$ element in the path ξ_{i+1}^j , its ancestry also included. The element $k_{i,i+1}^j \sim h(x_{i,i+1}^j)$ is sampled from some random function which can depend on the state, and, independently, $w_{i,i+1}^j \sim \mathcal{N}(0, \Delta t I_n)$. The next state in the path is computed as

$$x_{i+1,i+1}^j = x_{i,i+1}^j + k_{i,i+1}^j \Delta t + \sigma(t_i, x_{i,i+1}^j) w_{i,i+1}^j. \quad (26)$$

The measures \vec{P}_i and \vec{P}_{i+1} may not agree on the interval $[0, t_i]$. To see why this is permissible, consider Theorem 3.3 with $\tau = t_{i+1}$ and $t = t_i$. In a backward step, some P_{i+1} is used to produce a relationship to solve for the deterministic

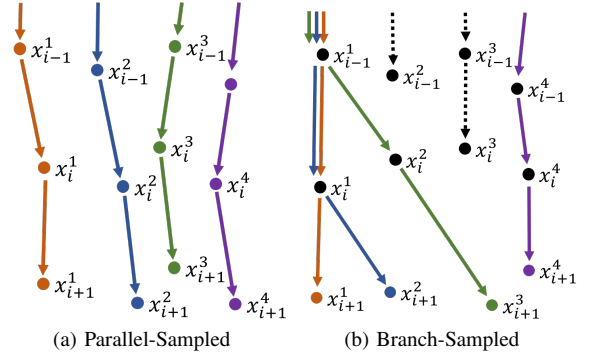


Fig. 2: Comparing parallel-sampling of the path measure \vec{P}_{i+1} , in which SDE paths are sampled independently, to the proposed representation. Dotted edges are present in the data structure but do not contribute to the path measure \vec{P}_{i+1} (but will contribute to \vec{P}_i and \vec{P}_{i-1}).

function $V^\mu(t_i, x)$. But an independent application of the theorem with $\tau = t_i$ and $t = t_{i-1}$ can use any new measure P_i . The only requirement is that each \vec{P}_i is consistent with the assumptions placed on P_i .

In the construction of \vec{P}_{i+1} in Fig. 2 (b) we can see that some edges are multiply represented in the distribution. If the drift term K_i were a deterministic function of X_i , such a construction would represent an unfaithful characterization of the path distribution because samples of the Brownian process are independent and thus should be sampled as in Fig. 2 (a). However, since K_i itself has a distribution, we can interpret overlapping paths as the drift having been selected so as to concentrate the paths in a certain part of the state space. The faithful representation of the independent process W_s^P might be weakened by this construction, but some guarantees about the convergence of such measures with increasing numbers of samples are available in [28].

B. FBRRT Iterative Algorithm

The goal of the FBRRT algorithm is to produce the set of parameters $\{\alpha_i\}_{i=1}^N$ which approximate the optimal value function $V(x; \alpha_i) \approx V^*(t_i, x)$. The forward pass produces a graph representation \mathcal{G} of the path measures $\{\vec{P}_i\}_{i=1}^N$. Given that the optimal policy has the form (3), we define the target policy

$$\begin{aligned} \mu_i(x; \alpha_{i+1}) &= \arg \min_{u \in U} \{ \ell(t_i, x, u) + f(t_i, x, u)^\top \partial_x V(x; \alpha_{i+1}) \}, \end{aligned} \quad (27)$$

so that it coincides with the optimal control policy when the value function approximation is exact. The backward pass uses \mathcal{G} , μ_i , and ρ_{i+1} to produce α_i , backwards in time. At the beginning of the next iteration, nodes with high heuristic value ρ_{i+1} are pruned from the tree and \mathcal{G} is regrown from those remaining.

C. Kinodynamic RRT Forward Sampling

In general, we desire sampling methods which seek to explore the whole state space, increasing the likelihood of

sampling in the proximity of optimal trajectories. For this reason, we chose methods inspired by kinodynamic RRT, proposed in [18]. The selection procedure for this method ensures that the distribution of the chosen particles is more uniformly distributed in a user-supplied region of interest $\mathcal{X}^{\text{roi}} \subseteq \mathbb{R}^n$, more likely to select particles which explore empty space, and less likely to oversample dense clusters of particles.

With some probability $\varepsilon_i^{\text{rt}} \in [0, 1]$ we choose the RRT sampling procedure, but otherwise choose a particle uniformly from $\{x_i^j\}_{j=1}^M$, each particle with equal weight. This ensures dense particle clusters will still receive more attention. Thus, the choice of the parameter $\varepsilon_i^{\text{rt}}$ balances exploring the state space against refining the area around the current distribution.

For drift generation we again choose a random combination of exploration and exploitation. For exploitation we choose

$$K_i = f(t_i, X_i, \mu_i(X_i; \alpha_i)). \quad (28)$$

For exploration we choose

$$K_i = f(t_i, X_i, u^{\text{rand}}). \quad (29)$$

where the control is sampled randomly from a user supplied set $u^{\text{rand}} \sim U^{\text{rand}}$. For example, for minimum fuel (L_1) problems where control is bounded $u \in [-1, 1]$ and the running cost is $L = |u|$, we select $U^{\text{rand}} = \{-1, 0, 1\}$ because the policy (27) is guaranteed to only return values in this discrete set.

Algorithm 1 sketches out the implementation of the RRT-based sampling procedure, producing the forward sampling tree \mathcal{G} . The algorithm takes as input any tree with width \widetilde{M} and adds nodes at each depth until the width is M , the parameter indicating the desired width. On the first iteration there are no value function estimate parameters available to exploit, so we set $\varepsilon^{\text{rt}} = 1$ to maximize exploration using the RRT sampling.

D. Path-Integral Backwards Weighing

We now propose a heuristic design choice for the backward pass weighing variables ρ_{i+1} , and justify their choice with some theoretical results. A good heuristic will give high weights to paths likely to have low value over the whole interval $[0, T]$. Thus, in the middle of the interval we care both about the current running cost and the expected cost. A dynamic programming principle result following directly from [22, Chapter 4, Corollary 7.2] indicates that

$$V^*(0, x_0) = \min_{u_{[0, t_{i+1}]}} E_{\mathbf{P}_{i+1}^u} \left[\int_0^{t_{i+1}} \ell(s, X_s, u_s) ds + V^*(t_{i+1}, X_{i+1}) \right],$$

where $u_{[0, t_{i+1}]}$ is any control process in U on the interval $[0, t_{i+1}]$ and \mathbf{P}_{i+1}^u is the measure produced by the drift $K_s = f(s, X_s, u_s)$. Following this minimization, we choose the heuristic to be

$$\rho_{i+1} = \int_0^{t_{i+1}} \ell(s, X_s, u_s) ds + V^*(t_{i+1}, X_{i+1}), \quad (30)$$

Algorithm 1 RRT Branched-Sampling

```

1: procedure FORWARDEXPAND( $\mathcal{G}, (\alpha_1, \dots, \alpha_N)$ )
2:   for  $k = \widetilde{M} + 1, \dots, M$  do  $\triangleright$  Add node each loop
3:     for  $i = 0, \dots, N - 1$  do  $\triangleright$  For each time step
4:        $\{x_i^j\}_j \leftarrow \mathcal{G}.\text{nodesAtTime}(i)$ 
5:       if  $\varepsilon_i^{\text{rt}} > \kappa^{\text{rt}} \sim \text{Uniform}([0, 1])$  then
6:          $x_i^{\text{rand}} \sim \text{Uniform}(\mathcal{X}^{\text{roi}})$ 
7:          $(x_i^{\text{near}}, j^{\text{near}}) \leftarrow \text{Nearest}(\{x_i^j\}_j, x_i^{\text{rand}})$ 
8:       else
9:          $(x_i^{\text{near}}, j^{\text{near}}) \sim \text{Uniform}(\{x_i^j\}_j)$ 
10:      end if  $\triangleright j^{\text{near}}$  is index of selected node
11:      if  $\varepsilon_i^{\text{opt}} > \kappa^{\text{opt}} \sim \text{Uniform}([0, 1])$  then
12:         $u_i \leftarrow \mu_i(x_i^{\text{near}}; \alpha_{i+1})$   $\triangleright$  (27)
13:      else
14:         $u_i \sim U^{\text{rand}}$ 
15:      end if
16:       $k_i \leftarrow f(t_i, x_i^{\text{near}}, u_i)$ 
17:       $w_i \sim \mathcal{N}(0, \Delta t I_n)$ 
18:       $x_{i+1}^{\text{next}} \leftarrow x_i^{\text{near}} + k_i \Delta t + \sigma(t_i, x_i^{\text{near}}) w_i$ 
19:       $j^{\text{next}} \leftarrow \mathcal{G}.\text{addEdge}(i, j^{\text{near}}, (x_i^{\text{near}}, k_i, x_{i+1}^{\text{next}}))$ 
20:       $\vec{\ell}_{0:i-1} \leftarrow \mathcal{G}.\text{getRunCost}(i-1, j^{\text{near}})$ 
21:       $\vec{\ell}_{0:i} \leftarrow \vec{\ell}_{0:i-1} + \ell_i(x_i^{\text{near}}, u_i) \Delta t$ 
22:       $\mathcal{G}.\text{setRunCost}(i, j^{\text{next}}, \vec{\ell}_{0:i})$ 
23:    end for
24:  end for
25:  return  $\mathcal{G}$ 
26: end procedure

```

where $u_{[0, t_{i+1}]}$ is chosen identically to how the control for the drift is produced. Although the theory does not require K_s to be a feasible drift under the dynamic constraints, for reasons like this it is useful for it to be chosen in this way. The running cost is computed in the forward sampling in line 21 of Algorithm 1.

Algorithm 2 details the implementation of the backward pass with local entropy weighting. The value function is represented by a linear combination of multivariate Chebyshev polynomials up to the 2nd order, $V(x; \alpha_i) = \Phi(x) \alpha_i$. Line 18 does not, theoretically, have an effect on the optimization, since it will come out of the exponential as a constant multiplier, but it has the potential to improve the numerical conditioning of the exponential function computation as discussed in [26, Chapter 5, equation (6.33)]. The λ value is, in general, a parameter which must be selected by the user. For some problems we choose to search over a series of possible λ parameters, evaluating each one with a backward pass and using the one that produces the smallest expected cost over a batch of trajectory rollouts executing the computed policy.

V. NUMERICAL RESULTS

We evaluated the FBRRT algorithm by applying it to a pair of nonlinear stochastic optimal control problems. For both problems, we used a minimum fuel (L_1) running cost of $L(u) = a|u|$, $a > 0$, $u \in [-1, 1]$, where the terminal cost

Algorithm 2 Local Entropy Weighted LSMC Backward Pass

```

1: procedure BACKWARDWLSMC( $\mathcal{G}$ )
2:    $\{\xi_N^j\}_j \leftarrow \mathcal{G}.\text{pathsAtTime}(N)$ 
3:    $\{x_N^j\}_j \leftarrow \{\xi_N^j\}_j$ 
4:    $y_N \leftarrow [g(x_N^1) \cdots g(x_N^M)]^\top$ 
5:    $\alpha_N \leftarrow \arg \min_{\alpha} \sum_j \Theta_N(\hat{y}_N^j - \Phi(x_N^j)\alpha)^2$ 
6:   for  $i = N-1, \dots, 1$  do  $\triangleright$  For each time step
7:      $\{\xi_{i+1}^j\}_j \leftarrow \mathcal{G}.\text{pathsAtTime}(i+1)$ 
8:     for  $j = 1, \dots, M$  do  $\triangleright$  For each path
9:        $(x_i^j, k_i^j, x_{i+1}^j) \leftarrow \xi_{i+1}^j$   $\triangleright x_i^j = x_{i,i+1}^j$ , etc.
10:       $y_{i+1}^j \leftarrow \Phi(x_{i+1}^j)\alpha_{i+1}$   $\triangleright$  (22)
11:       $z_{i+1}^j \leftarrow \sigma_{i+1}^\top(x_{i+1}^j)\partial_x \Phi(x_{i+1}^j)\alpha_{i+1}$   $\triangleright$  (15)
12:       $\mu_i^j \leftarrow \mu_i(x_i^j; \alpha_{i+1})$   $\triangleright$  (27)
13:       $d_i^j \leftarrow \sigma_{i+1}^{-1}(x_{i+1}^j)(f_i^\mu - k_i^j)$ 
14:       $\hat{y}_i^j \leftarrow y_{i+1}^j + (\ell_i^\mu + z_{i+1}^{\top} d_i^j)\Delta t$   $\triangleright$  (17)
15:       $\bar{\ell}_{0:i} \leftarrow \mathcal{G}.\text{getRunCost}(i, j)$ 
16:       $\rho_{i+1}^j \leftarrow y_{i+1}^j + \bar{\ell}_{0:i}$   $\triangleright$  (30)
17:   end for
18:    $\rho_{i+1} \leftarrow \rho_{i+1} - \min_j \{\rho_{i+1}^j\}$   $\triangleright$  exp conditioning
19:    $\Theta_{i+1} \leftarrow \exp(-1/\lambda \rho_{i+1})$   $\triangleright$  (21)
20:    $\alpha_i \leftarrow \arg \min_{\alpha} \sum_j \Theta_{i+1}^j (\hat{y}_i^j - \Phi(x_i^j)\alpha)^2$   $\triangleright$  (23)
21: end for
22: return  $(\alpha_1, \dots, \alpha_N)$ 
23: end procedure

```

is a quadratic function centered at the origin. Examples ran in Matlab 2019b on an Intel G4560 CPU with 8GB RAM.

Fig. 3 illustrates our method applied to the L_1 inverted pendulum problem. Note that even though there were no paths in the tree that continued along the 1st iteration’s mean trajectory (blue line) from beginning to end, the algorithm was still able to produce a policy in regions where no particles were produced. The green particles along the backward swing inform the policy in the beginning of the trajectory while the green particles near the origin inform it near the end, despite taking different paths in the tree.

The policies computed after the first few iterations are visualized in Fig. 4. Of significant note is that the policy obtained after only one iteration (red hue) does significantly well in general. For the L_1 inverted pendulum problem evaluated in [2], convergence required 55 iterations, but for our method only a handful of iterations were needed to get comparable performance. We also compared the convergence speed and robustness of FBRRT to parallel-sampled FBSDE [2] by randomly sampling different starting states and evaluating their relative performance over a number of trials. We normalized the final costs across the initial states by dividing all costs for a particular initial state by the largest cost obtained across both methods. For each iteration, we assign the value of the accumulated minimum value across previous iterations for that trial, i.e., the value is the current best cost after running that many iterations, regardless of the current cost. We aggregated these values across initial states and trials into the box plots in Fig. 5. Since the FBRRT is

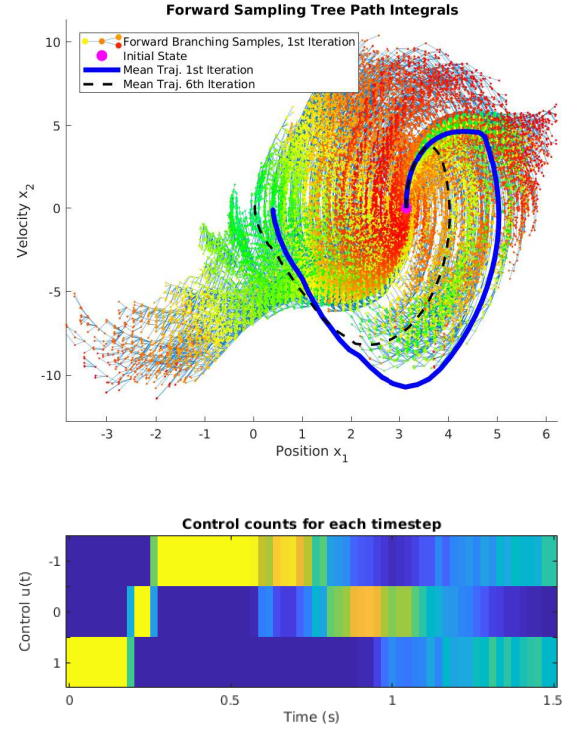


Fig. 3: Forward sampling tree for the first iteration of the L_1 inverted pendulum problem. Hue corresponds to the path-integral heuristic ρ used for weighing particles in the backward pass and for pruning the tree (green values are smaller). The blue and black dashed lines are the mean of trajectory rollouts, following the policies computed at the end of the 1st and 6th iterations respectively. Control counts are based on trajectory rollouts of the 6th iteration policy computed by FBRRT. The hue of each rectangle indicates the relative frequency of each control signal in $\{-1, 0, 1\}$ for each time step.

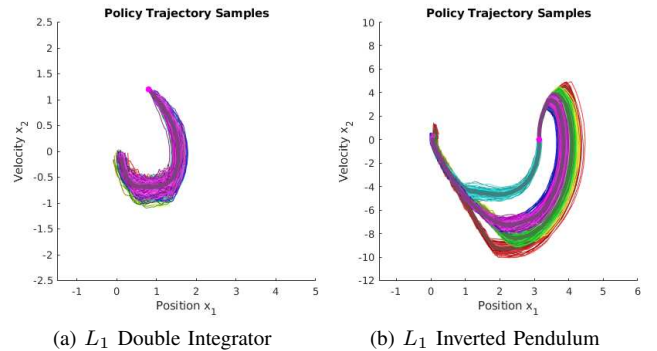


Fig. 4: Trajectory samples from policies generated after the first 6 iterations, the first iteration colored red, followed by yellow, green, cyan, dark blue, and magenta. All terminal costs are centered at $(0, 0)$. Dark thick lines are the mean trajectories.

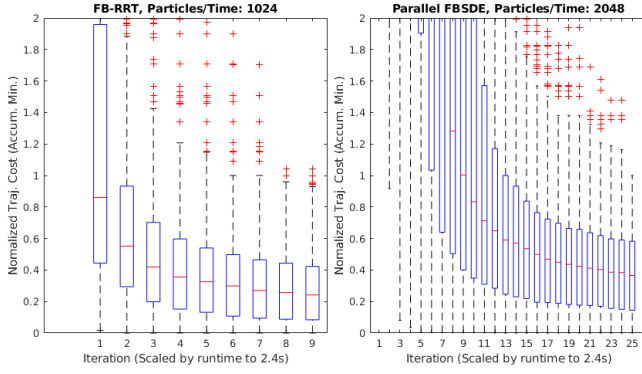


Fig. 5: Comparison of FBRRT and FBSDE for the L_1 double integrator problem for random initial states. Expected trajectory costs for the computed policies are normalized across different initial conditions.

significantly slower than the FBSDE per iteration due to the RRT nearest neighbors calculation, we scale each iteration by runtime. By nearly every comparison, FBRRT converges faster and in fewer iterations than FBSDE, and does so with half as many particle samples.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we have proposed a novel generalization of the FBSDE approach to solve stochastic optimal control problems, combining both branched sampling techniques with weighted least squares function approximation to greatly expand the flexibility of these methods. Leveraging the efficient space-filling properties of RRT methods, we have demonstrated that our method significantly improves convergence properties over previous FBSDE methods. We have shown how the proposed method works hand in hand with a proposed path integral-weighted LSMC method, concentrating function approximation in the regions where optimal trajectories are most likely to be dense. We have demonstrated that FBRRT can generate feedback control policies for nonlinear stochastic optimal control problems with non-quadratic costs.

Future work includes incorporating modern RRT algorithms, since most could be adapted to this approach with the proper book-keeping. Further, with very minor additions to the forward sampling algorithm, the methods might be applied to problems where the system must avoid obstacles, though experimental verification of the approach is needed. Another significant area of research worth investigating is to find better methods of value function representation. Although 2nd-order polynomials generally produce nice policy functions, they are unlikely to produce a good approximation of the value function outside of a local region. Finally, evaluation on higher dimensional problems would be useful to demonstrate the usefulness of this method.

REFERENCES

- [1] I. Exarchos and E. A. Theodorou, "Stochastic optimal control via forward and backward stochastic differential equations and importance sampling," *Automatica*, vol. 87, pp. 159–165, 2018.
- [2] I. Exarchos, E. A. Theodorou, and P. Tsiotras, "Stochastic L^1 -optimal control via forward and backward sampling," *Systems and Control Letters*, vol. 118, pp. 101–108, 2018.
- [3] —, "Stochastic Differential Games: A Sampling Approach via FBSDEs," *Dynamic Games and Applications*, 2018.
- [4] —, "Game-theoretic and risk-sensitive stochastic optimal control via forward and backward stochastic differential equations," in *Conference on Decision and Control, Las Vegas, Nevada*, 2016, pp. 6154–6160.
- [5] H. M. Soner and N. Touzi, "A stochastic representation for the level set equations," *Communications in Partial Differential Equations*, vol. 27, no. 9-10, pp. 2031–2053, 2002.
- [6] C. Bender and R. Denk, "A forward scheme for backward SDEs," *Stochastic Processes and their Applications*, 2007.
- [7] F. A. Longstaff and E. S. Schwartz, "Valuing American options by simulation: A simple least-squares approach," *Review of Financial Studies*, 2001.
- [8] J. Ma and J. Yong, *Forward-Backward Stochastic Differential Equations and their Applications*. Springer, 2007.
- [9] D. H. Jacobson and D. Q. Mayne, *Differential dynamic programming*. New York, NY: North-Holland, 1970.
- [10] E. A. Theodorou, Y. Tassa, and E. Todorov, "Stochastic differential dynamic programming," in *American Control Conference, Baltimore, Maryland*. IEEE, 2010, pp. 1125–1132.
- [11] Y. Tassa, T. Erez, and W. D. Smart, "Receding Horizon Differential Dynamic Programming," in *Advances in Neural Information Processing Systems 20*, 2008, pp. 1465–1472.
- [12] Y. Tassa, *Theory and Implementation of Biomimetic Motor Controllers (Ph.D. Thesis)*. Hebrew University of Jerusalem, 2011.
- [13] J. Yong and X. Y. Zhou, *Stochastic Controls: Hamiltonian Systems and HJB Equations*. Springer Science and Business Media, 1999.
- [14] E. Pardoux and S. G. Peng, "Adapted solution of a backward stochastic differential equation," *Systems and Control Letters*, vol. 14, no. 1, pp. 55–61, 1990.
- [15] S. Peng, "Backward stochastic differential equations and applications to optimal control," *Applied Mathematics and Optimization*, vol. 27, no. 2, 1993.
- [16] N. El Karoui, S. Peng, and M. C. Quenez, "Backward stochastic differential equations in finance," *Mathematical Finance*, vol. 7, no. 1, pp. 1–71, 1997.
- [17] I. M. Mitchell, "A toolbox of level set methods," *Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, Tech. Rep. TR-2004-09*, July, pp. 177–247, 2004.
- [18] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, 2001.
- [19] I. Noreen, A. Khan, and Z. Habib, "Optimal path planning using RRT* based approaches: a survey and future directions," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, pp. 97–107, 2016.
- [20] O. Arslan, E. A. Theodorou, and P. Tsiotras, "Information-theoretic stochastic optimal control via incremental sampling-based algorithms," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, Orlando, FL*, 2014.
- [21] W. H. Fleming and R. W. Rishel, *Deterministic and stochastic optimal control*. Springer, 1975.
- [22] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*. Springer Science and Business Media, 2006.
- [23] S. Peng, "Probabilistic interpretation for systems of quasilinear parabolic partial differential equations," *Stochastics and Stochastics Reports*, vol. 37, no. 1-2, pp. 61–74, 1991.
- [24] C. Bender and T. Moseler, "Importance sampling for backward SDEs," *Stochastic Analysis and Applications*, vol. 28, no. 2, 2010.
- [25] E. A. Theodorou and E. Todorov, "Relative entropy and free energy dualities: Connections to path integral and KL control," in *IEEE Conference on Decision and Control, Maui, Hawaii*. IEEE, 2012.
- [26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [27] S. Resnick, *A Probability Path*. Birkhäuser Verlag AG, 2003.
- [28] P. Del Moral, *Mean Field Simulation for Monte Carlo Integration*. Chapman and Hall/CRC, 2013.