2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

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

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

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683281

***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) can be utilized for the forward integration pass, as long as the controlled drift term is appropriately compensated in the backward integration pass. 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.**

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], differential games [3], [4], and reachability problems [1]. 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.

Monte Carlo-based FBSDE numerical methods originated in the mathematical finance community (see [5] for a review). These methods are primarily concerned with the solution of a single pair of forward and backward SDEs, typically beginning by densely sampling the forward SDE, then solving the backward SDE via a variety of techniques, including estimating conditional expectations using a least-squares Monte Carlo (LSMC) scheme [6], refining the process with a Picard-iteration scheme [7], and refining over shorter intervals using a multilevel scheme [8]. It was recognized in [1] that applying Girsanov's theorem to both of the pair of forward and backward SDEs can be used to change the sampling measure of the forward SDE at will, as long as an appropriate compensation is added to the backward SDE. For optimal control problems it is desired that the value function associated with the backward process be approximated around the distribution of the optimally controlled forward process. Since this distribution is not known a priori, this necessitates an *iterative-FBSDE* (iFBSDE) numerical method, which alternates between solving a particular pair of FBSDEs and improving the forward sampling measure to more closely match an optimal distribution, leading to the methods proposed in [1], [2], [4].

In this work we offer a novel iFBSDE method which incorporates rapidly-exploring random trees (RRTs) (see, e.g., [9] and the recent survey in [10]), in order to more efficiently explore the state space in the forward SDE sampling. 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. In the backward pass, we take advantage of the path-integrated running costs along with the estimates of the value function to produce a heuristic which weighs paths according to a local-entropy measure-theoretic optimization. Although local-entropy path integral theory and RRTs have been used together in [11], that method is more closely related to the path-integral approach to control [12]. 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:

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

For the sake of brevity all the proof of the main theorems have been removed. They can be found in the extended version of the paper in [13].

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

K. Hawkins is with Toyota Research Institute, Ann Arbor, MI, USA; A. Pakniyat is with the department of Mechanical Engineering at the University of Alabama, Tuscaloosa, AL, USA; E. Theodorou and P. Tsiotras are with the Georgia Institute of Technology, Atlanta, GA, USA. Contact at kelsey.hawkins@tri.global, apakniyat@ua.edu {evangelos.theodorou, tsiotras}@gatech.edu

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

measure Q and adapted to the filtration $\{\mathcal{F}_t\}_{t\in[0,T]}$. Consider a stochastic system whose dynamics are governed by

$$\mathrm{d}X_s = f(s, X_s, u_s)\,\mathrm{d}s + \sigma(s, X_s)\,\mathrm{d}W_s^{\mathsf{Q}}, \quad X_0 = x_0, \quad (1)$$

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 $f : [0,T]\times\mathbb{R}^n\times U \to \mathbb{R}^n$, $\sigma : [0,T]\times\mathbb{R}^n \to \mathbb{R}^{n\times n}$ are the Markovian drift and diffusion functions respectively. The cost associated with a given control signal is

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

where $\ell : [0,T]\times\mathbb{R}^n\times U \to \mathbb{R}^+$ is the running cost, and $g : \mathbb{R}^n \to \mathbb{R}^+$ is the terminal cost. We tacitly assume all necessary regularity assumptions for ℓ and g to guarantee existence and uniqueness of solutions [14, p. 156; Chapter 3, Theorem 4.2, Theorem 4.4].

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

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

Under mild regularity assumptions, in particular that $\sigma\sigma^\top$ is uniformly positive definite, there exists a unique classical solution V^* 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 $V^*(t,x) = \mathbf{E}_{\mathsf{Q}}^{t,x}[\, S_t[\pi^*]\,]$, where $\partial_x V^*$ is the partial derivative of V^* with respect to state x [15, 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 $\mu : [0,T]\times\mathbb{R}^n \to U$ and their associated value functions V^μ, and use iterative methods to approximate V^* and π^*. The on-policy value function is defined as

$$V^\mu(t,x) = \mathbf{E}_{\mathsf{Q}}^{t,x}[\, S_t^\mu\,], \quad S_t^\mu := \int_t^T \ell_s^\mu\,\mathrm{d}s + g(X_T), \quad (4)$$

with the process X_s satisfying the forward SDE (FSDE)

$$\mathrm{d}X_s = f_s^\mu\,\mathrm{d}s + \sigma_s\,\mathrm{d}W_s^{\mathsf{Q}}, \quad X_t = x, \quad (5)$$

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

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

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 admissible μ.

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. [16]) it is linked to to the solution (X_s, Y_s, Z_s) of the pair of FBSDEs consisting of the FSDE (5) and the backward SDE (BSDE)

$$\mathrm{d}Y_s = -\ell_s^\mu\,\mathrm{d}s + Z_s^\top\,\mathrm{d}W_s^{\mathsf{Q}}, \qquad Y_T = g(X_T), \quad (6)$$

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,

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

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

$$\widehat{Y}_{t,\tau} := Y_\tau + \int_t^\tau \ell_s^\mu \mathrm{d}s. \quad (9)$$

B. Off-Policy FBSDEs

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

$$\mathrm{d}X_s = K_s\,\mathrm{d}s + \sigma_s\,\mathrm{d}W_s^{\mathsf{P}}, \qquad X_0 = x_0, \quad (10)$$
$$\mathrm{d}Y_s = -(\ell_s^\mu + Z_s^\top D_s)\,\mathrm{d}s + Z_s^\top \mathrm{d}W_s^{\mathsf{P}}, \; Y_T = g(X_T), \quad (11)$$

with $D_s := \sigma_s^{-1}(f_s^\mu - K_s)$, where K_s is an arbitrary $\mathcal{F}_s$-progressively measurable and bounded process satisfying the smoothness conditions of [17, Chapter 1, Theorem 6.16], P is the new probability measure associated with K_s, and W_s^{P} a Brownian process over the new, complete, filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in[0,T]}, \mathsf{P})$.

Theorem 3.2: For the solution (X_s, Y_s, Z_s) to the FBSDE characterized by (10) and (11), it holds that

$$\begin{aligned} Y_s &= V^\mu(s, X_s), && s \in [0,T], \\ Z_s &= \sigma_s^\top \partial_x V^\mu(s, X_s), && \text{a.e. } s \in [0,T], \end{aligned} \quad (12)$$

P-a.s., and in particular,

$$Y_t = \mathbf{E}_{\mathsf{P}}[\widehat{Y}_{t,\tau} | X_t] = V^\mu(t, X_t), \qquad \mathsf{P}\text{-a.s.}, \quad (13)$$

where

$$\widehat{Y}_{t,\tau} := Y_\tau + \int_t^\tau (\ell_s^\mu + Z_s^\top D_s)\,\mathrm{d}s. \quad (14)$$

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 (13) 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.

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 Monte Carlo samples. 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 later times of the value function have been approximated, we can form 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 the use of a bounded heuristic random variable ρ_t to produce a new measure R_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

$$\mathsf{R}_t \in \arg\min_{\mathsf{R}_t} \left\{ \mathbf{E}_{\mathsf{R}_t}[\rho_t] + \lambda \mathcal{H}(\mathsf{R}_t \| \mathsf{P}_t) \right\}, \tag{15}$$

with $\lambda > 0$ a tuning variable and $\mathcal{H}(\mathsf{R}_t \| \mathsf{P}_t) = \mathbf{E}_{\mathsf{R}_t}\left[\log\left(\frac{\mathrm{d}\mathsf{R}_t}{\mathrm{d}\mathsf{P}_t}\right)\right]$ is the relative entropy of R_t which takes its minimum value when $\mathsf{R}_t = \mathsf{P}_t$, the distribution in which all sampled paths have equal weight.

The minimizer (15), which balances between the value of ρ and the relative entropy of its induced measure, has a solution of R_t^* determined [18, p. 2] as

$$\mathrm{d}\mathsf{R}_t^* = \Theta_t \mathrm{d}\mathsf{P}_t, \quad \Theta_t := \frac{\exp(-1/\lambda \rho_t)}{\mathbf{E}_{\mathsf{P}_t}[\exp(-1/\lambda \rho_t)]}. \tag{16}$$

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 *softmin* operation over paths according to the heuristic, a method often used in deep learning literature [19].

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

$$Y_t = \mathbf{E}_{\mathsf{P}_\tau}[\widehat{Y}_{t,\tau} | X_t] = V^\mu(t, X_t), \qquad \mathsf{R}_\tau\text{-a.s.}, \tag{17}$$

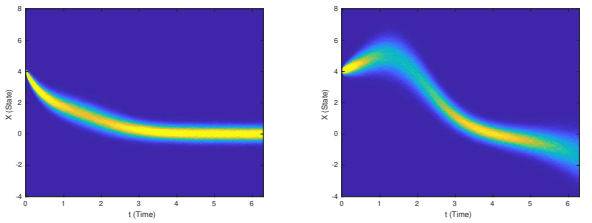
where $\widehat{Y}_{t,\tau}$ is defined in (14). Furthermore, the minimizer ϕ^* of the optimization

$$\inf_{\phi \in L_2} \mathbf{E}_{\mathsf{R}_\tau}[(\widehat{Y}_{t,\tau} - \phi(X_t))^2] = \inf_{\phi \in L_2} \mathbf{E}_{\mathsf{P}_\tau}[\Theta_\tau^{\mathsf{R}|\mathsf{P}} (\widehat{Y}_{t,\tau} - \phi(X_t))^2], \tag{18}$$

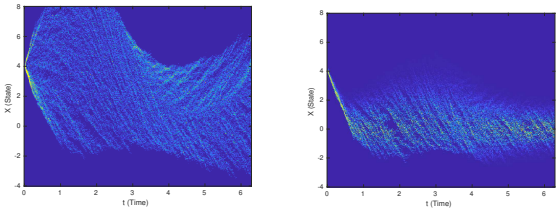
over X_t-measurable square integrable variables $\phi(X_t)$ coincides with the value function $\phi^*(X_t) = V^\mu(t, X_t)$.

In the following section, we evaluate the minimization of the right hand side of (18) 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. 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.



(a) Optimal Distribution (b) Parallel-Sampled Suboptimal



(c) RRT-Sampled (P) (d) RRT-Sampled, Weighted (R)

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.

IV. FORWARD-BACKWARD RRT

A. McKean-Markov Branched Sampling

We approximate the continuous-time sampling distributions with discrete-time McKean-Markov branch sampled paths as presented in [20]. First, for a given Δt, the interval $[0, T]$ is partitioned according to the time steps $(t_0 = 0, \ldots, t_i = (\Delta t)i, \ldots, t_N = T)$. 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 $\{\overrightarrow{\mathsf{P}}_i\}_{i=0}^N$, $\overrightarrow{\mathsf{P}}_i := \frac{1}{M}\sum_{j=1}^M \delta_{\xi_i^j}$, where δ is the Dirac-delta measure acting on sample paths $\xi_i^j := (x_{0,i}^j, k_{0,i}^j, x_{1,i}^j, k_{1,i}^j, \ldots, k_{i-1,i}^j, x_{i,i}^j)$, with $x_{\ell,i}^j, k_{\ell,i}^j \in \mathbb{R}^n$. The path notation $x_{\ell,i}^j$ indicates that this element is the sample of random variable X_ℓ that is the ancestor of sample $x_{i,i}^j$ in the jth path ξ_i^j of the ith measure $\overrightarrow{\mathsf{P}}_i$. Fig. 2 (b) illustrates how these measures are represented using a tree data structure. Each node in the tree x_i^j, alternatively called

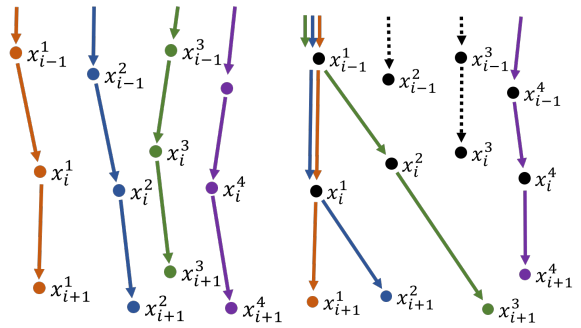


Fig. 2: Comparing parallel-sampling of the path measure $\overrightarrow{\mathsf{P}}_{i+1}$, in which SDE paths are sampled independently, to the proposed representation. Dotted edges are present in the data structure but do not contribute to the path measure $\overrightarrow{\mathsf{P}}_{i+1}$ (but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

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 (10). 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

$$x_{i+1,i+1}^j = x_{i,i+1}^j + k_{i,i+1}^j \Delta t + \sigma(t_i, x_{i,i+1}^j) w_{i,i+1}^j. \quad (19)$$

The measures $\overrightarrow{\mathsf{P}}_i$ and $\overrightarrow{\mathsf{P}}_{i+1}$ 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 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 of that theorem is that each $\overrightarrow{\mathsf{P}}_i$ is consistent with the assumptions placed on P_i, and this is satisfied by the Euler-Maruyama sampling scheme.

In the construction of $\overrightarrow{\mathsf{P}}_{i+1}$ 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 evolution of the process X_i with law $\overrightarrow{\mathsf{P}}_{X_i}$ in this way is called McKean-Markov because the distribution of X_i depends on the law $\overrightarrow{\mathsf{P}}_{X_{i-1}}$ of X_{i-1}, not just the realization of X_{i-1}. That is, each particle x_i^j is allowed to depend on the full collection of particles $\{x_{i-1}^j\}$ at the previous time step, instead of just its ancestor. Some guarantees about the desirable properties of such representations—used also in particle filters, Markov Chain Monte Carlo methods, and evolutionary algorithms—are available in [20].

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 $\{\overrightarrow{\mathsf{P}}_i\}_{i=1}^N$. Given that the optimal policy has the form (3), we define the target policy $\mu_i(x; \alpha_{i+1})$ as the solution of the problem

$$\min_{u \in U} \{\ell(t_i, x, u) + f(t_i, x, u)^\top \partial_x V(x; \alpha_{i+1})\}, \quad (20)$$

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 choose methods inspired by kinodynamic RRT, proposed in [9]. 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 $\varepsilon_i^{\text{rrt}} \in [0, 1]$ 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 $\varepsilon_i^{\text{rrt}}$ 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 $K_i = f(t_i, X_i, \mu_i(X_i; \alpha_i))$. For exploration we choose $K_i = f(t_i, X_i, u^{\text{rand}})$, 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 (20) 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 produce a policy μ, so we set $\varepsilon^{\text{rrt}} = 1$ 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 this choice. 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

Algorithm 1 RRT Branched-Sampling

1: **procedure** FORWARDEXPAND($\mathcal{G}, (\alpha_1, \ldots, \alpha_N)$)
2: **for** $k = \widetilde{M} + 1, \cdots, M$ **do** ▷ Add node each loop
3: **for** $i = 0, \cdots, N-1$ **do** ▷ For each time step
4: $\{x_i^j\}_j \leftarrow \mathcal{G}$.nodesAtTime$(i)$
5: **if** $\varepsilon^{\text{rrt}} > \kappa^{\text{rrt}} \sim$ Uniform$([0,1])$ **then**
6: $x_i^{\text{rand}} \sim$ Uniform$(\mathcal{X}^{\text{roi}})$
7: $(x_i^{\text{near}}, j^{\text{near}}) \leftarrow$ Nearest$(\{x_i^j\}_j, x_i^{\text{rand}})$
8: **else**
9: $(x_i^{\text{near}}, j^{\text{near}}) \sim$ Uniform$(\{x_i^j\}_j)$
10: **end if** ▷ j^{near} is index of selected node
11: **if** $\varepsilon^{\text{opt}} > \kappa^{\text{opt}} \sim$ Uniform$([0,1])$ **then**
12: $u_i \leftarrow \mu_i(x_i^{\text{near}}; \alpha_{i+1})$ ▷ (20)
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: $j^{\text{next}} \leftarrow \mathcal{G}$.addEdge$(i, j^{\text{near}}, (x_i^{\text{near}}, k_i, x_{i+1}^{\text{next}}))$
20: $\overrightarrow{\ell}_{0:i-1} \leftarrow \mathcal{G}$.getRunCost$(i-1, j^{\text{near}})$
21: $\overrightarrow{\ell}_{0:i} \leftarrow \overrightarrow{\ell}_{0:i-1} + \ell_i(x_i^{\text{near}}, u_i)\Delta t$
22: $\mathcal{G}$.setRunCost$(i, j^{\text{next}}, \overrightarrow{\ell}_{0:i})$
23: **end for**
24: **end for**
25: **return** $\mathcal{G}$
26: **end procedure**

and the expected cost. A dynamic programming principle result following directly from [14, Chapter 4, Corollary 7.2] indicates that

$$V^*(0, x_0) = \min_{u[0,t_{i+1}]} E_{\mathsf{P}_{i+1}^u}[\int_0^{t_{i+1}} \ell(s, X_s, u_s)\,\mathrm{d}s + V^*(t_{i+1}, X_{i+1})],$$

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

$$\rho_{i+1} = \int_0^{t_{i+1}} \ell(s, X_s, u_s)\,\mathrm{d}s + V^*(t_{i+1}, X_{i+1}), \qquad (21)$$

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 second 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

Algorithm 2 Local Entropy Weighted LSMC Backward Pass

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

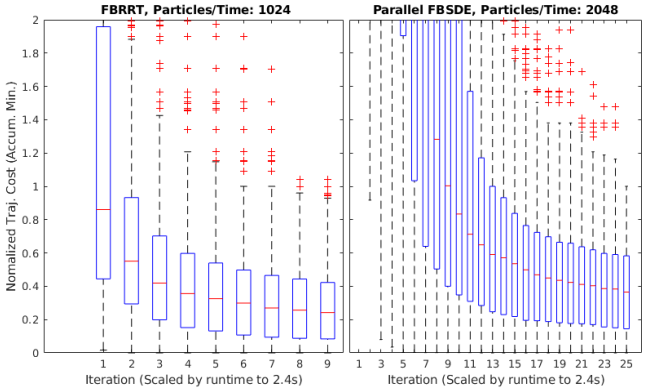


Fig. 3: Comparison of parallel-sampled FBSDE [2] and FBRRT for the L_1 double integrator problem for random initial states.

numerical conditioning of the exponential function computation as discussed in [19, 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 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 is a quadratic function centered at the origin. Examples ran in Matlab 2019b on an Intel G4560 CPU with 8GB RAM.

The first problem is a L_1 double integrator problem with dynamics $f = \begin{bmatrix} x_2 & u \end{bmatrix}^\top$ and $\sigma = \text{diag}(0.01, 0.1)$. We 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. 3. Since FBRRT is 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.

Fig. 4 illustrates FBRRT applied to the L_1 inverted pendulum problem with dynamics $f = \begin{bmatrix} x_2 & a_1 x_2 + a_2 \sin x_1 + a_3 u \end{bmatrix}^\top$ and $\sigma = \text{diag}(0.04, 0.4)$. 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. For the L_1 inverted pendulum problem evaluated in [2], convergence required 55 iterations, but for our method only 6 iterations were needed to get comparable performance.

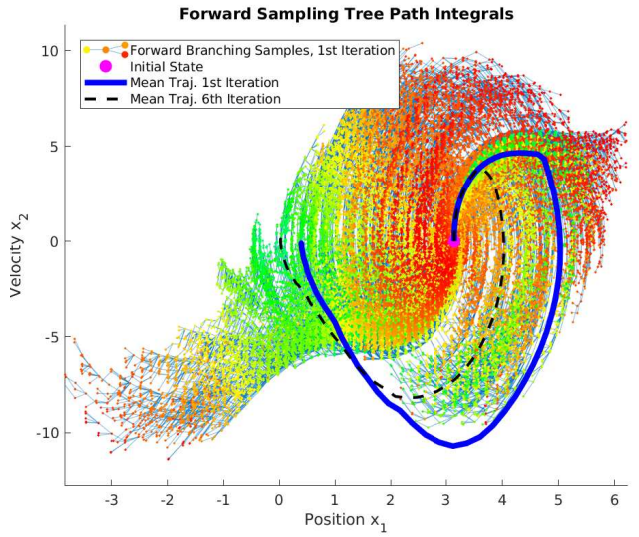


Fig. 4: Forward sampling tree for the first iteration of the L_1 inverted pendulum problem. Hue corresponds to the path-integral heuristic ρ used for weighing particles in the backward pass and for pruning the tree (green values are smaller). The blue and black dashed lines are the mean of trajectory rollouts, following the policies computed at the end of the 1st and 6th iterations respectively.

VI. CONCLUSIONS

In this work, we have proposed a novel generalization of the iFBSDE approach to solve stochastic optimal control problems. Leveraging the efficient space-filling properties of RRT methods, we have demonstrated that our method significantly improves convergence properties over previous iFBSDE methods. We have shown how branched-sampling 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 investigating better methods of value function representation and evaluation on higher dimensional problems 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] ——, "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.

[4] ——, "Stochastic Differential Games: A Sampling Approach via FBSDEs," *Dynamic Games and Applications*, 2018.

[5] C. Bender and J. Steiner, "Least-Squares Monte Carlo for Backward SDEs," *Springer Proceedings in Mathematics*, vol. 12, pp. 257–289, 2012.

[6] F. A. Longstaff and E. S. Schwartz, "Valuing American options by simulation: A simple least-squares approach," *Review of Financial Studies*, 2001.

[7] C. Bender and R. Denk, "A forward scheme for backward SDEs," *Stochastic Processes and their Applications*, 2007.

[8] M. B. Giles, "Multilevel Monte Carlo path simulation," *Operations Research*, vol. 56, no. 3, pp. 607–617, 2008.

[9] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, 2001.

[10] 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.

[11] 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.

[12] E. A. Theodorou, Y. Tassa, and E. Todorov, "Stochastic differential dynamic programming," in *American Control Conference, Baltimore, Maryland*. IEEE, 2010, pp. 1125–1132.

[13] K. P. Hawkins, A. Pakniyat, E. Theodorou, and P. Tsiotras, "Forward-backward rapidly-exploring random trees for stochastic optimal control," 2021. [Online]. Available: https://arxiv.org/abs/2006.12444

[14] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*. Springer Science and Business Media, 2006.

[15] W. H. Fleming and R. W. Rishel, *Deterministic and stochastic optimal control*. Springer, 1975.

[16] 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.

[17] J. Yong and X. Y. Zhou, *Stochastic Controls: Hamiltonian Systems and HJB Equations*. Springer Science and Business Media, 1999.

[18] 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.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[20] P. Del Moral, *Mean Field Simulation for Monte Carlo Integration*. Chapman and Hall/CRC, 2013.