A Topology-Guided Path Integral Approach for Stochastic Optimal Control

Jung-Su Ha and Han-Lim Choi

Abstract—This work presents an efficient method to solve a class of continuous-time, continuous-space stochastic optimal control problems of robot motion in a cluttered environment. The method builds upon a path integral representation of the stochastic optimal control problem that allows computation of the optimal solution through sampling and estimation process. As this sampling process often leads to a local minimum especially when the state space is highly non-convex due to the obstacle field, we present an efficient method to alleviate this issue by devising a proposed topological motion planning algorithm. Combined with a receding-horizon scheme in execution of the optimal control solution, the proposed method can generate a dynamically feasible and collision-free trajectory while reducing concern about local optima. Illustrative numerical examples are presented to demonstrate the applicability and validity of the proposed approach.

I. INTRODUCTION

Computing the optimal policy for a system driven by some uncertain disturbance, which is called a stochastic optimal control problem, is one of the most important problems in planning/control of robotic platforms in a complex environment. In a discrete-time/discrete-state and control space setting, the problem is formulated as a Markov decision process (MDP) and solved through the dynamic programming procedure, e.g. value iteration or policy iteration. The problem in a continuous setting, which is of the primary interest of this work, can be solved in a similar manner if transformed into a discretized version; however, this discretization approach is not scalable for a high-dimensional state space. Alternatively, an optimality condition for the continuous problem itself can be derived and utilized. It is well known that the optimality condition results in a nonlinear partial differential equation (PDE), called the Hamilton-Jacobi-Bellman equation; but, solving a nonlinear PDE is intractable in most robotic control applications.

Fortunately, there is a class of stochastic optimal control problem, called linearly-solvable optimal control (LSOC) [1], for which the HJB equation can be solved in a more efficient way with appropriate reformulation. For an LSOC problem, the notion of a desirability function, which is effectively an exponential value function, is introduced in order to transcribe the original nonlinear HJB equation on the value function into a linear PDE on the desirability function. In addition, it has been found that the Feynman-Kac formula allows the solution of such linear PDE to be expressed as an expectation of some path integral. As a result, the stochastic optimal control problem is transformed into an estimation problem, which can be solved by sampling a set of stochastic paths and then evaluating their expectation. This aforementioned procedure to solve a LSOC is referred to as path integral (PI) control [2]. For more interesting views and different derivations of PI control, we would refer the reader to [3] and references therein.

Advanced estimation techniques, such as importance sampling, can be applied to effectively solve the aforementioned transformed problem of a LSOC. In [4,5], the control policy is parameterized and then estimated using an importance sampling technique on the basis of the path integral formula. In [6], path-integral formula is utilized to construct a state-dependent feedback controller and theoretical analysis on how sampling strategies affect the estimation results is presented. In [7], the cross entropy method was applied to build an efficient importance sampler that reduces estimation variance. In [8], the rapidly-exploring random tree (RRT) algorithm was used to help the importance sampler to pick valuable samples.

This paper addresses a continuous LSOC problem, especially in a complex configuration space with obstacles, in the path integral control framework. This type of problem may have many local optima, since the state space is often highly non-convex due to obstacle regions. Thus, a sampler for PI control needs to be able to generate samples diverse and spread enough in order not to be trapped into a local minimum; however, it is not particularly easy for many conventional sampling schemes to generate samples very far from most of other samples. To tackle this issue, the approach in this work is, therefore, (i) first specifies all possible local minima caused by obstacles for deterministic approximation problem and then (ii) generates samples around all these local minima taking them as reference trajectories. If the global minimum of the original problem is near one of these references, this way eventually results in finding the global optimal solution.

Specifically in the context of motion planning in a cluttered environment, each local minimum can be associated with a different topological class; thus, a motion planner that can produce a optimized motion trajectory for every different topological class is required to support the above two-step process. There have been some attempts to build a topology-embedded path planner (although not in the context of stochastic control). In [9,10], the authors introduced/proposed the concept of $H$-signature to distinguish different homology classes of trajectories and incorporate it...
into a graph search algorithm. This concept of $H$-signature is valid, but it is known to be difficult for the graph search algorithm to handle high-dimensional state space and system dynamics. Therefore, this work proposes to incorporate the $H$-signature concept into a sampling-based motion planner that can handle more diverse class of motion planning problems with system dynamics.

Consequently, this paper presents an algorithm, termed path-integral with rapidly-exploring random homology-embedded tree star (PI-RRHT$^*$), that consists of a homology-embedded optimal motion planner to identify all the local minima of the deterministic approximation problem and an importance sampler that solves a transformed estimation problem of the original LSCO. Combined with a receding-horizon scheme for plan & execution of the stochastic optimal solution, the proposed method can produces the globally optimal, dynamically feasible collision-free trajectory for stochastic systems.

The procedure expanding trees to the topological concept augmented space is reminiscent of that of the rapidly-exploring random belief tree (RRBT) [11] algorithm, which expands a graph in the state space using the rapidly-exploring random graph (RRG) [12] algorithm and projects an associated tree onto the belief space. Also, the proposed method in this work takes advantage of the architecture of the goal-rooted feedback motion Trees (GR-FMTs) [13] algorithm: it first expands a goal-rooted backward tree over a topology-embedded state space without considering uncertainty in the dynamics. Then, instead of directly using the tree as a feedback policy, the control policy for LSCO is computed using the path integral control method.

II. Problem Description

Let a state space, $\chi$, and control input space, $U$, be a compact subset of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. An obstacle region, $\chi_{obs}$, and a goal region, $\chi_{goal}$, are compact subsets of $\chi$. Then a domain of the problem is defined by $D \subset \chi \setminus (\chi_{obs} \cup \chi_{goal})$, which is the closure of its interior, $D^\circ$ and has a smooth boundary $\partial D$; the boundary of goal region is denoted by $\partial D_{goal}$.

Suppose $x \in D$ and $u \in U$ are a state and control vector, respectively, $w$ is an $m$-dimensional zero-mean Brownian motion process. Consider the stochastic dynamics of which deterministic drift term is affine in control input:

$$dx = f(x)dt + G(x)u dt + B(x)\, dw \tag{1}$$

where $f : D \to \mathbb{R}^n$ is the passive dynamics and $G : D \to \mathbb{R}^{n \times m}$ is control transition matrix and $B : D \to \mathbb{R}^{n \times m}$ is the diffusion matrix function. In this work, the state is assumed to be partitioned as $x = [x'_m \, x'_c]'$ and then other terms are partitioned as $f(x) = \begin{bmatrix} f_m(x)' & f_c(x)' \end{bmatrix}'$, $G(x) = \begin{bmatrix} 0'_{(n-m) \times m} & G_c(x)' \end{bmatrix}'$ and $B(x) = \begin{bmatrix} 0'_{(n-m) \times m} & B_c(x)' \end{bmatrix}'$. It is also assumed that $G_c : D \to \mathbb{R}^{m \times m}$ and $B_c : D \to \mathbb{R}^{m \times m}$ are invertible.

The objective of the problem is to find a control policy which achieves the goal region while avoiding collision with other boundaries (e.g. obstacles) and also minimizes the cost functional. The problem is formulated as a first-exit stochastic optimal control problem. Let a function $q : D \times U \to \mathbb{R}$ and $\phi : \partial D \to \mathbb{R}$ be an instantaneous state cost rate and a final cost function, respectively, where $\mathbb{R}$ denotes the extended real number line $\mathbb{R} \cup \{-\infty, +\infty\}$. For given control policy $\pi : D \to U$, the cost functional which we want to minimize is defined as:

$$J^\pi(x) = E \left[ \phi(x(t_f)) + \int_{t}^{t_f} q(x) + \frac{1}{2} u' R(x) u \, dt \right], \tag{2}$$

where the first-exit time $t_f$ is the time when the resulting trajectory first reaches the boundary of the domain, i.e.

$$t_f \equiv \inf \{ t \geq 0 : x(t) \notin D^0 \}, \tag{3}$$

where $x(t)$ is a solution of (1) under the control policy $\pi$. The final cost function penalizes the situation the trajectory cannot reaches the goal boundary, $\partial D_{goal}$.

III. Linearly-Solvable Stochastic Optimal Control

A. Path Integral Control

The optimal cost-to-go function is defined as:

$$v(x) \equiv \inf_{\pi} J^\pi(x), \tag{4}$$

and the associated Hamilton-Jacobi-Bellman (HJB) equation is given by:

$$0 = \min_u \left( q + \frac{1}{2} u' R u + (f + Gu)' v_x + \frac{1}{2} \text{tr} (BB' v_{xx}) \right). \tag{5}$$

From the HJB equation, the optimal control law is obtained analytically as:

$$u^*(x) = -R(x)^{-1} G(x)' v_x(x). \tag{6}$$

Substituting this optimal control law to (5) yields the second order nonlinear partial differential equation (PDE):

$$0 = q(x) + v'_x f(x) - \frac{1}{2} v'_x G(x) R(x)^{-1} G(x)' v_x + \frac{1}{2} \text{tr} (v_{xx} B(x) B'(x)). \tag{7}$$

Due to its nonlinearity, solving the above PDE is intractable. The nonlinearity can be removed by introducing the desirability function:

$$\psi(x) = \exp(-\frac{1}{\lambda} v(x)), \tag{8}$$

where a scalar, $\lambda$ comes from the relation,

$$\lambda G(x) R(x)^{-1} G(x)' = B(x) B(x)'. \tag{9}$$

This restriction means that the control and noise affect the dynamics on the same subspace and in the same direction and the control cost is reversely related to the noise scale [1,2]. Roughly speaking, with the above restriction the control is more expensive for the direction that the noise is smaller.
Rewriting the PDE in \( \text{[7]} \) with respect to \( \psi(x) \) induces the second order linear PDE as:

\[
0 = -\frac{1}{\lambda} q(x) \psi + f(x) \psi' + \frac{1}{2} \text{tr}(\psi_{xx} B(x) B(x)'), \ x \in D^0, \tag{10}
\]

where the boundary condition is given by:

\[
\psi(x) = \exp(-\frac{1}{\lambda} \phi(x)), \ x \in \partial D. \tag{11}
\]

The problem in \( \text{[10]} \) and \( \text{[11]} \) is called the Dirichlet problem associated with an elliptic operator \( \text{[14]} \). The solution can be represented probabilistically by the Feynman-Kac formula. Following corollary is directly modified from Proposition 7.2 in \( \text{[14]} \).

**Corollary 1 (Feynman-Kac):** Let \( x(t) \) be a solution of

\[
dx = f(x)dt + B(x)d\omega(0) \tag{12}
\]

and \( t_f \) be a first-exit time as \( \text{[3]} \). If \( E_P[t_f] < \infty, \forall x \in D \) then, a solution of the Dirichlet problem \( \text{[10]} \) and \( \text{[11]} \) is given by:

\[
\psi(x) = E_P \left[ \psi(x(t_f)) \exp \left( -\frac{1}{\lambda} \int_t^{t_f} q(x) d\tau \right) \right], \tag{13}
\]

where the expectation \( E_P[\cdot] \) is taken over all trajectories \( x(t), \ t \in [0, t_f] \).

The optimal control \( \text{[6]} \) is written with respect to \( \psi \) as:

\[
\mathbf{u}^*(x) = \lambda R(x)^{-1} G(x) \frac{\psi(x)}{\psi(x)}. \tag{14}
\]

Equation \( \text{[13]} \) can be expressed as

\[
\psi(x) = \int W(\bar{x}) P(\bar{x}) d\bar{x}, \tag{15}
\]

where \( W(\bar{x}) = \psi(x(t_f)) \exp \left( -\frac{1}{\lambda} \int_t^{t_f} q(x(\tau)) d\tau \right) \) and \( \bar{x} \) and \( P(\bar{x}) \) represent trajectories and its probability measure, respectively. From the path integral formulation \( \text{[2]} \), the probability measure of trajectory is given by:

\[
P(\bar{x}) = c \lim_{dt \to 0} \exp \left( -\frac{1}{2\lambda} \sum_{j=1}^{N} \left[ \|\mu(x_j)\|^2_{\Sigma_e(x(t_j))^{-1}} \right] dt \right), \tag{16}
\]

where \( t_1 = t, \ t_N = t_f, \Sigma_e(x) = G(x)R(x)^{-1} G(x)' = B_e(x)B_e(x)'/\lambda \) and \( \mu(x_j) = x(t_1 + dt), x(t_f) - f_e(x(t_j)) \) and \( c \) is a normalization constant for \( \int dP(\bar{x}) = 1 \). Partial derivative of \( P \) is given by:

\[
\frac{\partial}{\partial x_e(t_1)} P(\bar{x}) = \frac{1}{\lambda} \mu(x_1)' \Sigma_e(x(t_1))^{-1} P(\bar{x}), \tag{17}
\]

which yields

\[
\psi_{x_e}(x) = \frac{1}{\lambda} \int W(\bar{x}) \Sigma_e^{-1}(x) \mu(x_1) P(\bar{x}) d\bar{x}, \tag{18}
\]

The optimal control \( \text{[14]} \) is expressed as

\[
\mathbf{u}^*(x) dt = \frac{1}{\psi(x)} R(x)^{-1} G_e(x)' \Sigma_e(x)^{-1} E_P [W(\bar{x}) \mu(x)] dt, \tag{19}
\]

\[
= \frac{1}{\psi(x)} G_e^{-1}(x) B_c(x) E_P \left[ W(\bar{x}) d\omega(0) \right], \tag{19}
\]

using \( \mu(x) dt = B_c(x) d\omega(0) \) and \( R(x)^{-1} G_e(x)' \Sigma_e(x)^{-1} = G_e^{-1}(x) \).

The desirability function and the optimal control can be estimated from Monte-Carlo (MC) sampling procedure; the estimations for state \( x \) with \( N \) sample trajectories are given by

\[
\hat{\psi}(x) = \frac{1}{N} \sum_{k=1}^{N} w^k, \tag{20}
\]

and

\[
\hat{u}(x) dt = \frac{1}{N \hat{\psi}(x)} G^{-1}_e(x) B_c(x) \sum_{k=1}^{N} w^k \delta w^k, \tag{21}
\]

where the weights, \( w \), and the first Brownian increments, \( \delta w \), of the \( k \)-th sample trajectory are obtained from following stochastic simulation.

1) Set \( i = 0, \ X_i = x. \)
2) \( X_{i+1} = X_i + f(X_i) \delta t + B(X_i) Z_i \sqrt{\delta t}, \) where \( Z_i \sim N(0, I_m). \)
3) If \( X_{i+1} \in D^0 \), then \( i = i + 1 \) and go to step 2.
4) If \( X_{i+1} \notin D^0 \), then finish the simulation. Return \( w^k = \exp \left( -\frac{1}{\lambda} \phi(X_{i+1}) \right) \) and \( \delta w^k = \mathbf{Z}_i \).

Note that \( \delta t \) is sufficiently small time step for simulation of a continuous stochastic process.

**B. Change of Measure**

In the naive MC sampling process, the sample trajectories for the estimation are collected from the passive diffusion dynamics \( \text{[12]} \). Most trajectories, however, may be useless (i.e. they hit the obstacle or reach the goal region through very awkward way, which are far from optimum). Rather than using naive MC sampling, it is possible to utilize advanced sampling technique to improve the quality of samples; the importance sampling scheme is widely adopted in the path integral control literature. Consider the new stochastic dynamics which drifts by some predefined deterministic process \( \mathbf{u}_m \).

\[
dx = f(x)dt + G(x)\mathbf{u}_m dt + B(x)d\omega(1), \tag{22}
\]

and let \( Q \) be the corresponding probability measure.

Then, the trajectories from the above stochastic dynamics can be used to estimate the desirability function and the optimal control, which is referred as a measure change or importance sampling. Rewriting \( \text{[15]} \) and \( \text{[19]} \) yields

\[
\psi(x) = \int W(\bar{x}) \frac{dP(\bar{x})}{dQ(\bar{x})} dQ(\bar{x}) = E_Q \left[ W(\bar{x}) \frac{dP(\bar{x})}{dQ(\bar{x})} \right], \tag{23}
\]

and

\[
\mathbf{u}^*(x) dt = \frac{1}{\psi(x)} G^{-1}_e(x) E_Q \left[ W(\bar{x}) \mu(x) dt \frac{dP(\bar{x})}{dQ(\bar{x})} \right]. \tag{24}
\]
The Radon-Nikodym derivative of $P$ with respect to $Q$, $\frac{dP}{dQ}$, can be obtained from following corollary.

**Corollary 2 (Girsanov’s Theorem [15]):** Suppose $P$ and $Q$ are the probability measures induced by the trajectories (12) and (22), respectively. Then the Radon-Nikodym derivative of $P$ with respect to $Q$, $\frac{dP}{dQ}$, is given by

$$\frac{dP}{dQ} = \exp\left(-\frac{1}{2\lambda} \int_0^{t_f} u_n'G_c'S^{-1}_cG_a u_n dt - \frac{1}{\lambda} \int_0^{t_f} u_n'G_c'S^{-1}_cB_c d\mathbf{w}(1)\right) = \exp\left(-\frac{1}{2\lambda} \int_0^{t_f} u_n'R u_n dt - \frac{1}{\lambda} \int_0^{t_f} u_n'G_c'S^{-1}_cB_c d\mathbf{w}(1)\right).$$

(25)

With new probability measure $Q$, sampling procedure is changed as

1) Set $i = 0$, $X_i = x$.
2) $X_{i+1} = X_i + f(X_i)\delta t + G(X_i)u_{in}(t_i)\delta t + B_i(X_i)\mathbf{Z}_i/\sqrt{\delta t}$, where $\mathbf{Z}_i \sim N(0, I_m)$.
3) If $X_{i+1} \notin D^o$, then $i = i + 1$ and go to step 2.
4) If $X_{i+1} \notin D^o$, then finish the simulation. Return $u^k = \exp\left(-\frac{1}{2}(\phi(X_{i+1}) + \delta t \sum_{j=0}^{i} L_j)\right)$ and $\delta w = Z_0$, where $L_j = g(X_j) + \frac{1}{2}u_{in}(t_j)/R\mathbf{u}_{in}(t_j) + u_{in}(t_j)'G_c'S^{-1}_cB_c Z_j/\sqrt{\delta t}$.

The estimation of the desirability function is the same as (20) but because $\mu(x)dt = G_c(x)u_{in} dt + B_c(x) d\mathbf{w}(1)$ by substituting it to (24), the estimation of the optimal control is given as,

$$\hat{u}(x)\delta t = u_{in}\delta t + \frac{1}{N\psi(x)}G^{-1}_cB_c(x) \sum_{k=1}^{N} w^k \delta w^k.$$  (26)

Note that all the estimations are unbiased [6,7]. Especially, it is proven that the variance of estimation decreases as $u_{in}$ becomes closer to the real optimal control $u^*$ [6].

**IV. Sampling with Topology-Embedded Planner**

One can view that by using importance sampling, sample trajectories are obtained around (or biased to) the reference trajectory induced by deterministic dynamics with $u_{in}$. Then through path integral procedure, the optimal trajectory/control is obtained by modifying the reference trajectory/control. One candidate of reference is the optimal trajectory for deterministic (noise-free) system.

However, the modification may force the result to local optimum if the amount of samples are not enough or the samples are far from global optimum. Also, the problems addressed in this work may have many local optima, because the state space has high non-convexity due to obstacle regions. The difficulty caused from non-convex space can be resolved if we have sample trajectories around every local optimum.

In this section, we propose the Path Integral with Rapidly-exploring Random Homology-embedded Tree (PI-RRHT*) algorithm in order to resolve such difficulty. The algorithm consists of expansion (Algorithm 1) and execution (Algorithm 2) phases, where the former operates in lead-time and intermittently between execution phases, and latter runs online; such construction has been utilized in [13]. In expansion phase, the algorithm finds all different topological classes of trajectories for deterministic optimal motion planning problem. And in execution phase, the control input for stochastic optimal control is computed in a receding horizon scheme with the path integral formula.

**A. Topological Representation of Trajectories in 2D**

Presence of obstacles in an environment differentiates topological classes among trajectories. Suppose the configuration space, $C$, and the obstacles are given by 2-dimensional subsets of $\mathbb{R}^2$. Let $\sigma : [0,1] \rightarrow \mathbb{C}$ be a trajectory in the configuration space and $\sigma_1$ and $\sigma_2$ connecting the same start and end coordinates. The two trajectories are called homologous if $\sigma_1$ together with $\sigma_2$ (the later with opposite orientation) forms the complete boundary of a 2-dimensional manifold embedded in $\mathbb{C}$ not containing/intersecting any of the obstacles [9].

The configuration space can be represented as a subset of the complex plane $\mathbb{C}$, i.e. $(x,y) \in C \Leftrightarrow x + iy \in C$. The obstacles are also represented as subsets of the complex plane, $O_1, O_2, ..., O_N \subset C$, and each obstacle has one representative point which is denoted as $\zeta_l \in O_l, \forall l = 1,..., N$. For a given set of representative points, the obstacle marker function, $F : C \rightarrow \mathbb{C}^N$, is defined as follows,

$$F(z) = \left[\frac{1}{z - \zeta_1}, \frac{1}{z - \zeta_2}, \cdots, \frac{1}{z - \zeta_N}\right]'.$$  (27)

Then, we can define $H$-signature, $\mathcal{H}_2 : C_1(\mathbb{C}) \rightarrow \mathbb{C}^N$, which represent homology class of trajectory as:

$$\mathcal{H}_2(\sigma) = \frac{1}{2\pi} \text{Im} \left( \int_{\mathcal{C}} F(z) dz \right),$$  (28)

where $C_1(\mathbb{C})$ is the set of all curves/trajctories in $\mathbb{C}$.

Especially, when the trajectory from $z_1$ to $z_2$ is short enough (that is, a straight line connecting the same points is in same homology class), its $H$-signature can be calculated analytically as

$$(\mathcal{H}_2(e))_l = \frac{1}{2\pi} \text{abs}\text{min}_{k \in \mathbb{Z}} (\arg(z_2 - \zeta_l) - \arg(z_1 - \zeta_l) + 2k\pi),$$  (29)

where function $\text{abs}\text{min}$ returns the value which have the minimum absolute value.

If two trajectories $\sigma_1$ and $\sigma_2$ connecting the same points have the same $H$-signatures, $\mathcal{H}_2(\sigma_1) = \mathcal{H}_2(\sigma_2)$, they are homologous and the reverse is also true. Also, we can restrict the homology class of trajectories by defining disjoint sets of allowed and blocked $H$-signatures, $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{U} = \mathcal{A} \cup \mathcal{B}$ and $\mathcal{U}$ denotes the set of the $H$-signatures of all trajectories. By well restricting the allowed $H$-signature set, the topological motion planning algorithm can secure scalability with the number of obstacles. It can be observed from Fig. 1 and Fig. 3 that there are plenty of trajectories which connect the same points and have different $H$-signatures.
The $H$-signatures for a higher dimensional space can be constructed by defining it directly [10] or by using the configuration space mappings to 2-dimensional spaces [16].

B. Expansion phase of PI-RRHT*: Topology-Embedded Sampling-Based Planner

Algorithm 1 PI-RRHT* algorithm (Expansion)

1: $(V, E) \leftarrow (\emptyset, \emptyset)$;
2: for $i = 1$ to $N_{iter}$ do
3: $x_{new} \leftarrow$ SAMPLING();
4: if $x_{new} \in X_{goal}$ then
5: $v_{new} \leftarrow$ ADDROOT($x_{new}$); $E_{new} \leftarrow \emptyset$;
6: else
7: $(v_{new}, E_{new}) \leftarrow$ CHOOSEPARENT($V, x_{new}$);
8: end if
9: if $\sim$ isempty($v_{new}.N$) then
10: $V \leftarrow V \cup v_{new}; E \leftarrow E \cup E_{new}$;
11: $(V, E) \leftarrow$ REWIRE($V, E, x_{new}$);
12: end if
13: end for

Algorithm 2 CHOOSEPARENT($V, x_{new}$)

1: $E_{new} \leftarrow \emptyset; V_{near,f} \leftarrow$ NEARFORWARD($V, x_{new}$);
2: for $v_{near,f} \in V_{near,f}$ do
3: $e_{near,f} \leftarrow$ TPBVP($x_{new}, v_{near,f}, x$);
4: if OBSTACLEFREE($e_{near,f}$) then
5: $E_{new} \leftarrow E_{new} \cup e_{near,f}$;
6: for $n \in v_{new}.N$ do
7: $n_{new} \leftarrow$ PROPAGATE($e_{near,f}, n$);
8: APPENDNODE($v(x_{new}), n_{new}$);
9: end for
10: end if
11: end for
12: return ($v(x_{new}), E_{new}$)

This subsection will be devoted to explain the expansion phase of PI-RRHT* algorithm which aims to find all optimal trajectories in different homology classes for deterministic approximation problem. The algorithm constructs a graph on state space based on Rapidly-exploring Random Graph (RRG) algorithm [12] and search over the graph to project a tree into $H$-signature augmented space. The graph in state space is defined by a set of vertices, $V$, and edges, $E$, where each vertex is composed of a state, $v.x$, and set of associated nodes $v.N$. Each node $n \in v.N$ has its $H$-signature, $n.H$, a cost, $n.c$, and a parent node $n.parent$.

Some required functions are described as follows: SAMPLING() function returns a random state from the state space. ADDROOT($x_{new}$) creates root vertex as $v.x = x_{new}, v.n.c = 0, v.n.parent = 0$ and $v.n.H = 1 + H(e_{new})$ where $e_{new}$ is the straight line between $x_{new}$ and the goal representative point. OBSTACLEFREE(e) takes a trajectory $e$ as an argument and checks whether it lies in obstacle free region. NEARFORWARD($V, x$) and NEARBACKWARD($V, x$) functions return $O(\log |V|) \text{ number of vertices}$ (see [12]) among the set of vertices, $V$, from

Algorithm 3 REWIRE($V, E, x_{new}$)

1: $V_{near,b} \leftarrow$ NEARBACKWARD($V, x_{new}$);
2: for $v_{near,b} \in V_{near,b}$ do
3: $e_{near,b} \leftarrow$ TPBVP($v_{near,b}, x_{new}$);
4: if OBSTACLEFREE($e_{near,b}$) then
5: $E \leftarrow E \cup e_{near,b}$;
6: end if
7: end for
8: $Q \leftarrow v(x_{new}).N$;
9: while $Q \neq \emptyset$ do
10: $n \leftarrow$ POP($Q$); $\triangleright$ exhaustive search
11: for $v_{near,b}$ of $v(n)$ do
12: $n_{new} \leftarrow$ PROPAGATE($e_{near,b}, n$);
13: if APPENDNODE($v_{near,b}, n_{new}$) then
14: $Q \leftarrow$ INSERTQ($Q, n_{new}$);
15: end if
16: end for
17: end while
18: return $(V, E)$

and to $x$, respectively. TPBVP($x_{1}, x_{2}$) returns the optimal trajectory from $x_{1}$ to $x_{2}$ without considering obstacles (i.e. the solution of two point boundary value problem), which can be implemented in various ways with respect to the system dynamics and cost [13,17]-[19]. PROPAGATE($e, n$) returns the new node, $n_{new}$, which is created by propagating $n$ through $e$; the new node is given as $n_{new}.H = n.H + H(e)$, $n_{new}.c = n.c + \text{Cost}(e)$ and $n_{new}.parent = n$, where $H(e)$ and Cost(e) denotes $H$-signature and cost of trajectory, $e$, respectively. APPENDNODE($v, n_{new}$) checks if the new node is in the allowed homology class and imposes a partial ordering to the set of nodes in the vertex of the form:

$$n_{a} < n_{b} \Leftrightarrow (n_{a}.H = n_{b}.H) \land (n_{a}.c < n_{b}.c).$$

When above condition holds, $n_{a}$ is said to be dominated by $n_{b}$, meaning that the paths to the root from $n_{a}$ and $n_{b}$ are homologous but the cost of path from $n_{a}$ is smaller. This function takes a state vertex, $v$, and new node, $n_{new}$ as arguments and first checks if the new node is in blocked homology class, $n_{new}.H \in B$, or dominated by any existing nodes at $v$. Then, if it is blocked or dominated, the function returns failure. If it is not, the function appends the new node, checks if it dominates any nodes at $v$ and prunes when necessary. Finally, INSERTQ($Q, n_{new}$) adds a new node into a queue and prunes nodes there when necessary.

The algorithm operates as shown in Algorithm 1. In the main loop, it first samples a new state and checks if it is in the goal region. If so, a new vertex is created as a root of the tree and if not, the algorithm attempts to connect a new vertex to the graph in Algorithm 2 if such connection succeed, the new vertex will contain some nodes. With the successfully created new vertex, the algorithm adds it and the new (forward) edges to the graph on line 10 and finds/adds new backward edges on line 1-7 in Algorithm 3. These procedures are analogous to the RRG algorithm [12], which.
guarantees to asymptotically contain all possible trajectories through the space.

In Algorithm 3, after adding all new edges, all nodes in the new vertex are added to the queue, \( Q \), on line 8. Then the queue is exhaustively searched using uniform cost search, like Dijkstra’s algorithm on line 9-17. These procedures make the graph project the tree into \( H \)-signature augmented space by propagating nodes in the queue and pruning with criteria in \( (30) \).

C. Execution Phase of PI-RRHT*: Receding Horizon Path Integral Control with Topology-Guided Path Sampling

Algorithm 4 PI-RRHT* algorithm (Execution)

1. Given the current state \( x_{\text{cur}} \) and the Tree \((V,E)\);  
2. while \( x_{\text{cur}} \notin \partial D \) do  
3. \( \hat{U}_{in} \leftarrow \text{ExtractReference}(x_{\text{cur}}, (V,E)) \);  
4. \( \hat{u}(x_{\text{cur}}) \leftarrow \text{PathIntegral}(x_{\text{cur}}, \hat{U}_{in}) \);  
5. \( x_{\text{cur}} \leftarrow \text{ApplyControl}(x_{\text{cur}}, \hat{u}(x_{\text{cur}})) \);  
6. end while

Algorithm 5 ExtractReference(\( x_{\text{cur}}, (V,E) \))

1. \( (v_{\text{new}}, E_{\text{new}}) \leftarrow \text{ChooseParent}(V; x_{\text{cur}}) \);  
2. \( V \leftarrow V \cup v_{\text{new}}; \ E \leftarrow E \cup E_{\text{new}} \);  
3. \( \{ \hat{q}_{in}^{(h)}, \ h = 1, 2, \ldots, H \} \leftarrow \text{ReconstructPath}(G \leftarrow (V,E), v_{\text{new}}) \);  
4. return \( \hat{U}_{in} \leftarrow \{ \hat{u}_{in}^{(h)}, \ h = 1, 2, \ldots, H \} \)

The execution phase of PI-RRHT* presented in Algorithm 4 computes and executes the optimal control for stochastic problem in a receding horizon fashion. It consists of three procedures: ExtractReference(\( x_{\text{cur}}, (V,E) \)) shown in Algorithm 5 takes the current state \( x_{\text{cur}} \) and the tree \((V,E)\) constructed from Algorithm 1 as arguments and returns all open-loop optimal control tapes for the allowed homology trajectories from \( x_{\text{cur}} \) to the roots of the tree in \( \chi_{\text{goal}} \).

Then in PathIntegral(\( x_{\text{cur}}, \hat{U}_{in} \)), trajectories are sampled around each homology class, \( h \), and the optimal control is computed. Suppose there are \( H \) number of stochastic dynamics \( (22) \) controlled by \( \hat{u}_{in}^{(h)} \) and \( Q_h, \ h = 1, 2, \ldots, H \) are corresponding probability measures. Equation \( (23) \) and \( (24) \) can be rewritten as

\[
\hat{\psi}(x) = \frac{1}{H} \sum_{h=1}^{H} E_{Q_h} \left[ W(x) \frac{dP(x)}{dQ_h(x)} \right],
\]

and

\[
\hat{u}^*(x)dt = \frac{1}{H} \sum_{h=1}^{H} \frac{1}{\hat{\psi}(x)} G^{-1}_c(x) E_{Q_h} \left[ W(x) \mu_h(x)dt \frac{dP(x)}{dQ_h(x)} \right].
\]

Suppose we sample \( N \) trajectories from each homology class, \( h = 1, 2, \ldots, H \), by procedure described in section III-B and let the weights of \( k \)th sample trajectory in \( h \)th homotopy class be indexed by \(\hat{\psi}^{(h,k)}(x)\). Then we have

\[
\hat{\psi}(x) = \frac{1}{H} \sum_{h=1}^{H} \hat{\psi}^{(h)}(x),
\]

and

\[
\hat{\psi}^{(h)}(x) dt = \frac{1}{H} \sum_{h=1}^{H} \hat{\psi}^{(h)}(x) dt,
\]

where \(\hat{\psi}^{(h)}(x) = \frac{1}{N} \sum_{k=1}^{N} u^{(k,h)}(x)\) and

\[
\hat{u}^{(h)}(x) dt = \frac{1}{N \hat{\psi}(x)} \sum_{k=1}^{N} u^{(k,h)}(x) G^{-1}_c(x) B_c(x) \delta w^{(k,h)}(x).
\]

Note that from the above equations, the optimal control is only computed at the current time and state. However, if the control policy is restricted as the open loop formulation, the state dependence term is dropped and we can obtain the open loop control tape by replacing \( \delta w^{(k,h)} = Z_i, \forall i = 2, 3, \ldots \) (see [6]). As a result, PathIntegral(\( x_{\text{cur}}, \hat{U}_{in} \)) procedure computes the open loop policy for one-period of receding horizon. Such control is applied to the system for one-period by ApplyControl(\( x_{\text{cur}}, \hat{u}(x_{\text{cur}}) \)), then the overall algorithm repeats again until the state reaches the boundary of the domain.

V. Numerical Example

For the first example, we consider a simple two-dimensional stochastic single integrator in the environment having two obstacles. The dynamics and the cost rate are given by:

\[
f(x) = 0, \ G(x) = I_2, \ q(x) = 1 \text{ and } R(x) = 2I_2,
\]

When the sampling procedure is performed, the reference control tape \( u_{\text{ref}} \) needs to be augmented by \( 0 \) after its time length, because the problem is the first-exit type, i.e. final time is not fixed.
i.e. the position of a robot in the configuration space, \( x \in D \subset \mathbb{R}^2 \), is controlled by the velocity input, \( u \in \mathbb{R}^2 \), while the objective of control is to reach the goal region while minimizing the cost function, \( J = E \left[ \phi(x(t_f)) + \int_0^{t_f} 1 + u'dt \right] \). At the boundary, the final cost is given as:

\[
\phi(x) = \begin{cases} 
0 & \text{if } x \in \partial D_{\text{goal}}, \\
\infty & \text{otherwise}.
\end{cases}
\]  

(35)

The state is driven also by a diffusion term that contains the 2-dimensional Brownian motion; two diffusion matrices are considered in this example for comparison:

\[ B(x) = 0.1I_2, \ 0.3I_2. \]

Through the path integral procedure, the time step for stochastic simulations and the number of samples for each reference trajectory are set as \( \delta t = 0.1 \) and \( N = 200 \), respectively. Finally, the period of receding horizon control is given by \( \delta t \).

Fig. 1 shows the results of the expansion phase (Algorithm 1) and \textsc{ExtractReference} function (Algorithm 5) for \( x_{\text{cur}} = [1, 2.5]^T \). It is observed that three trajectories in different homology classes are returned and they all connect the query state \( x_{\text{cur}} \) to the goal region. When projecting the tree onto \( H \)-augmented space, the set of allowable \( H \)-signature value is defined as \( A \equiv \{ z : |1 - z_i| \leq H_{\text{limit}}, \ i = 1, 2 \} \) with \( H_{\text{limit}} = 0.6 \) to extract trajectories in physically meaningful homology classes; otherwise, infinitely many trajectories that include paths revolving around the obstacle.
could be obtained.

Fig. 2 depicts snapshots of the receding-horizon control process of the execution phase (Algorithm 4) with two different diffusion matrices. Note that with large degree of diffusion the effect of Brownian noise will be so critical that the robot cannot pass through the narrow slit between the obstacles. It is observed from the figure that when the noise is not critical, the robot goes to the goal region directly but it makes a detour when the noise increases. It can be seen that by considering topologically various trajectories as references, the path-integral formula provides not only computations of local optimum around each reference but also comparative advantages between references.

The second example deals with a stochastic wheeled-mobile robot dynamics in more cluttered environment. The state $x = [x, y, \theta]^T$ represents the position and heading of a car, which is controlled by the turn rate, $u$. The dynamics is given by:

$$f(x) = \begin{bmatrix} V \cos(\theta) \\ V \sin(\theta) \end{bmatrix}, \quad G(x) = \begin{bmatrix} 0 \\ 1/\rho \end{bmatrix} \quad \text{and} \quad B(x) = \begin{bmatrix} 0 \\ b \end{bmatrix},$$

with $V = \rho = 1$ and $b = 0.03$, respectively. The cost rate and the final cost are set as $q(x) = R(x) = 1$ and $\phi(x) = \begin{cases} 0 & \text{if } x \in \partial D_{\text{goal}}, \\ 1000 & \text{otherwise}. \end{cases}$

Finally, we set $N = 300$ and $H_{\text{limit}} = 0.8$.

In the expansion phase, TPBVP solver returns the minimum length Dubin’s curve for the deterministic dynamics with $q = 1$, $R = 0$ and $|u| \leq 1$. Fig. 3 shows the paths from $x_{\text{cur}} = [0, 0, 0]^T$ to goal region in each homology class when 1000 vertices are added to the trees and lengths of the paths at each iteration. As it proceeds, the algorithm finds paths in various homology classes and refines those. Fig. 4 depicts some snapshots of the execution phase. It is shown that, by using the diverse reference trajectories, the importance sampler can obtain sample trajectories exploring the entire state space to compute correct estimation of optimal control.

VI. CONCLUSIONS

This paper has addressed a class of continuous-time, continuous-space stochastic optimal control problems on complex environment with obstacles. A Feynman-Kac path integral formula and an importance sampling method have been presented for the first-exit time problem. A topological concept embedded motion planner has been proposed to generate numerous reference trajectories in different homology classes. Then we have proposed a receding-horizon scheme which samples the trajectories around each reference; as a result, the proposed algorithm not only provides a dynamically feasible and collision-free trajectory but also effectively alleviates concern about local optima. Numerical examples have demonstrated the validity of the proposed approach.

Note that in our scheme, a sample trajectory is obtained with open-loop reference control-tape. As a result, a sample trajectory diverges from the reference trajectory as time passes as shown in Fig. 2 and 3. It reduces sample efficiency; if sample trajectories deviate a lot from the reference, they will make collision with high probability. It is expected to be resolved by utilizing the parameterized (feedback) policy as a reference. We leave it for a future work.

ACKNOWLEDGMENT

This work was supported by Agency for Defense Development (under in part contract #UD140053JD) and in part contract #UD150047JD).

REFERENCES

[1] E. Todorov, “Efficient computation of optimal actions,” Proceedings of the national academy of sciences, vol. 106, no. 28, pp. 11 478–11 483, 2009.
[2] H. J. Kappen, “Path integrals and symmetry breaking for optimal control theory,” Journal of statistical mechanics: theory and experiment, vol. 2005, no. 11, p. P11011, 2005.
[3] A. Bry and N. Roy, “Rapidly-exploring random belief trees for motion planning using incremental sampling-based methods,” in Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on. IEEE, 2013, pp. 3623–3628.
[4] J.-S. Ha, J.-J. Lec, and H.-L. Choi, “A successive approximation-based approach for optimal kinodynamic motion planning with nonlinear differential constraints,” in Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on. IEEE, 2013, pp. 3623–3628.
[5] ——, “Reinforcement learning of motor skills in high dimensions: A path integral approach,” in Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010, pp. 2397–2403.
[6] S. Thijssen and H. Kappen, “Path integral control and state-dependent feedback,” Physical Review E, vol. 91, no. 3, p. 032104, 2015.
[7] ——, “Adaptive importance sampling for control and inference,” arXiv preprint arXiv:1505.01874, 2015.
[8] S. Bhattacharya, M. Likhachev, and V. Kumar, “Topological constraints in search-based robot path planning,” Autonomous Robots, vol. 33, no. 3, pp. 273–290, 2012.
[9] S. Bhattacharya, D. Lipsky, R. Ghrist, and V. Kumar, “Invariants for homology classes with application to optimal search and planning problem in robotics,” Annals of Mathematics and Artificial Intelligence, vol. 67, no. 3-4, pp. 251–281, 2013.
[10] A. Bry and N. Roy, “Rapidly-exploring random belief trees for motion planning under uncertainty,” in Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011, pp. 723–730.
[11] J. Jeon, S. Karaman, and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” The International Journal of Robotics Research, vol. 30, no. 7, pp. 846–894, 2011.
[12] S. Karaman and E. Frazzoli, “Optimal sampling-based feedback motion trees among obstacles for controllable linear systems with linear constraints,” in Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015, pp. 4195–4201.
[13] I. Karatzas and S. Shreve, Brownian motion and stochastic calculus. Springer Science & Business Media, 2012, vol. 113.
[14] C. W. Gardiner et al., Handbook of stochastic methods. Springer Berlin, 1985, vol. 4.
[15] F. T. Pokorny, D. Krage, L. E. Kavraki, and K. Goldberg, “High-dimensional winding-augmented motion planning with 2d topological task projections and persistent homology,” in Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016.
[16] D. J. Webb and J. van den Berg, “Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics,” in Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013, pp. 5054–5061.
[17] J.-S. Ha, J.-J. Lec, and H.-L. Choi, “A successive approximation-based approach for optimal kinodynamic motion planning with nonlinear differential constraints,” in Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on. IEEE, 2013, pp. 3623–3628.
[18] S. Karaman and E. Frazzoli, “Optimal kinodynamic motion planning using incremental sampling-based methods,” in Decision and Control (CDC), 2010 49th IEEE Conference on. IEEE, 2010, pp. 7681–7687.