An Incremental Sampling-based Algorithm for Stochastic Optimal Control

Vu Anh Huynh, Sertac Karaman, Emilio Frazzoli

Abstract—In this paper, we consider a class of continuous-time, continuous-space stochastic optimal control problems. Building upon recent advances in Markov chain approximation methods and sampling-based algorithms for deterministic path planning, we propose a novel algorithm called the incremental Markov Decision Process (iMDP) to compute incrementally control policies that approximate arbitrarily well an optimal policy in terms of the expected cost. The main idea behind the algorithm is to generate a sequence of finite discretizations of the original problem through random sampling of the state space. At each iteration, the discretized problem is a Markov Decision Process that serves as an incrementally refined model of the original problem. We show that with probability one, (i) the sequence of the optimal value functions for each of the discretized problems converges uniformly to the optimal value function of the original stochastic optimal control problem, and (ii) the original optimal value function can be computed efficiently in an incremental manner using asynchronous value iterations. Thus, the proposed algorithm provides an anytime approach to the computation of optimal control policies of the continuous problem. The effectiveness of the proposed approach is demonstrated on motion planning and control problems in cluttered environments in the presence of process noise.

I. INTRODUCTION

Stochastic optimal control has been an active research area for several decades with many applications in diverse fields ranging from finance, management science and economics [1], [2] to biology [3] and robotics [4]. Unfortunately, general continuous-time, continuous-space stochastic optimal control problems do not admit closed-form or exact algorithmic solutions, and are known to be computationally challenging [5]. Many algorithms are available to compute approximate solutions of such problems. For instance, a popular approach is based on the numerical solution of the associated Hamilton-Jacobi-Bellman PDE (see, e.g., [6]–[8]). Other methods approximate a continuous problem with a discrete Markov Decision Process (MDP), for which an exact solution can be computed in finite time [9], [10]. However, the complexity of these two classes of deterministic algorithms scale exponentially with the dimension of the state and control spaces, due to discretization. Remarkably, algorithms based on random (or quasi-random) sampling of the state space provide a possibility to alleviate the curse of dimensionality in the case in which the control inputs take values from a finite set, as noted in [5], [11], [12].

Algorithms based on random sampling of the state space have recently been shown to be very effective, both in theory and in practice, for computing solutions to deterministic path planning problems in robotics and other disciplines. For example, the Probabilistic RoadMap (PRM) algorithm first proposed by Kavraki et al. [13] was the first practical planning algorithm that could handle high-dimensional path planning problems. Their incremental counterparts, such as RRT* [14], later emerged as sampling-based algorithms suited for online applications and systems with differential constraints on the solution (e.g., dynamical systems). The RRT algorithm has been used in many applications and demonstrated on various robotic platforms [15], [16]. Recently, optimality properties of such algorithms were analyzed in [17]. In particular, it was shown that the RRT algorithm fails to converge to optimal solutions with probability one. The authors have proposed the RRT* algorithm that guarantees almost-sure convergence to globally optimal solutions without any substantial computational overhead when compared to the RRT.

Although the RRT* algorithm is asymptotically optimal and computationally efficient (with respect to RRT), it cannot handle problems involving systems with uncertain dynamics. In this work, building upon asymptotically-optimal sampling-based planning algorithms [17] and the Markov chain approximation method [18], we introduce a novel algorithm called the incremental Markov Decision Process (iMDP) to approximately solve a wide class of stochastic optimal control problems. More precisely, we consider a continuous-time optimal control problem with continuous state and control spaces, full state information, and stochastic process noise. In iMDP, we iteratively construct a sequence of discrete Markov Decision Processes (MDPs) as discrete approximations to the original continuous problem, as follows. Initially, an empty MDP model is created. At each iteration, the discrete MDP is refined by adding new states sampled from the boundary as well as from the interior of the state space. Subsequently, new stochastic transitions are constructed to connect the new states to those already in the model. For the sake of efficiency, stochastic transitions are computed only when needed. Then, an anytime policy for the refined model is computed using an incremental value iteration algorithm, based on the value function of the previous model. The policy for the discrete system is finally converted to a policy for the original continuous problem. This process is iterated until convergence.

Our work is mostly related to the Stochastic Motion Roadmap (SMR) algorithm [19] and Markov chain approximation methods [18]. The SMR algorithm constructs an MDP over a sampling-based roadmap representation to maximize the probability of reaching a given goal region. However, in SMR, actions are discretized, and the algorithm does not offer any formal optimality guarantees. On the other hand, while available Markov chain approximation methods [18] provide formal optimality guarantees under very general conditions, a sequence of a priori discretizations of state and control spaces still impose expensive computation. The iMDP algorithm addresses this issue by sampling in the state space and sampling or discovering necessary controls.
The main contribution of this paper is a method to incrementally refine a discrete model of the original continuous problem in a way that ensures convergence to optimality while maintaining low time and space complexity. We show that with probability one, the sequence of optimal value functions induced by optimal control policies for each of the discretized problems converges uniformly to the optimal value function of the original stochastic control problem. In addition, the optimal value function of the original problem can be computed efficiently in an incremental manner using asynchronous value iterations. Thus, the proposed algorithm provides an anytime approach to the computation of optimal control policies of the continuous problem. Distributions of approximating trajectories and control processes returned from the iMDP algorithm approximate arbitrarily well distributions of optimal trajectories and optimal control processes of the original problem. Each iteration of the iMDP algorithm can be implemented with the time complexity $O(k^\theta \log k)$ where $0 < \theta \leq 1$ while the space complexity is $O(k^\theta)$. The entire processing time until the algorithm stops can be implemented in $O(k^{1+\delta} \log k)$. We can increase $k$ linearly over iterations, which leads to linear sampling complexity. Hence, the above sampling, space and time complexities make iMDP a practical incremental algorithm. The effectiveness of the proposed approach is demonstrated on motion planning and control problems in cluttered environments in the presence of process noise.

This paper is organized as follows. In Section II, a formal problem definition is given. The Markov chain approximation methods and the iMDP algorithm are described in Sections III and IV. The analysis of the iMDP algorithm is presented in Section V. Section VI is devoted to simulation examples and experimental results. The paper is concluded with final remarks in Section VII.

II. PROBLEM DEFINITION

Stochastic Dynamics: Let $d_x$, $d_u$, and $d_w$ be positive integers. The $d_x$-dimensional and $d_u$-dimensional Euclidean spaces are $\mathbb{R}^{d_x}$ and $\mathbb{R}^{d_u}$ respectively. Let $S$ be a compact subset of $\mathbb{R}^{d_x}$, which is the closure of its interior $S^o$ and has a smooth boundary $\partial S$. The state of the system at time $t$ is $x(t) \in S$, which is fully observable at all times. We also define a compact subset $U$ of $\mathbb{R}^{d_u}$ as a control set.

Suppose that a stochastic process $\{w(t); t \geq 0\}$ is a $d_w$-dimensional Brownian motion, also called a Wiener process, on some probability space $(\Omega, F, P)$. Let a control process $\{u(t); t \geq 0\}$ be a $U$-valued, measurable process also defined on the same probability space. We say that the control process $u(\cdot)$ is nonanticipative with respect to the Wiener process $w(\cdot)$ if there exists a filtration $\{F_t; t \geq 0\}$ defined on $(\Omega, F, P)$ such that $u(\cdot)$ is $F_t$-adapted, and $w(\cdot)$ is an $F_t$-Wiener process. In this case, we say that $u(\cdot)$ is an admissible control inputs with respect to $w(\cdot)$, or the pair $(u(\cdot), w(\cdot))$ is admissible. Let $\mathbb{R}^{d_x \times d_u}$ denote the set of all $d_x \times d_u$ real matrices. We consider stochastic dynamical systems, also called controlled diffusions, of the form

$$dx(t) = f(x(t), u(t)) \, dt + F(x(t), u(t)) \, dw(t), \forall t \geq 0 \quad (1)$$

where $f : S \times U \rightarrow \mathbb{R}^{d_x}$ and $F : S \times U \rightarrow \mathbb{R}^{d_x \times d_w}$ are bounded measurable and continuous functions as long as $x(t) \in S^o$. The matrix $F(\cdot, \cdot)$ is assumed to have full rank. More precisely, a solution to the differential form given in Eq. (1) is a stochastic process $\{x(t); t \geq 0\}$ such that $x(t)$ equals the following stochastic integral in all sample paths:

$$x(0) + \int_0^t f(x(t), u(t)) \, dt + \int_0^t F(x(t), u(t)) \, dw(t), \quad (2)$$

until $x(\cdot)$ exits $S^o$, where the last term on the right hand side is the usual Itô integral (see, e.g., [20]). When the process $x(\cdot)$ hits $\partial S$, the process $x(\cdot)$ is stopped.

Weak Existence and Weak Uniqueness of Solutions: Useful concepts to construct numerical solutions of Eq. 1 are weak sense solutions and their weak uniqueness [18], [21]. Essentially, these concepts allow us to assert the existence and uniqueness of the stochastic process $x(\cdot)$ via the existence and uniqueness of its probability distribution. In this paper, given the boundedness of the set $S$, and the definition of the functions $f$ and $F$ in Eq. (1), we have a weak solution to Eq. (1) that is unique in the weak sense [21]. The boundedness requirement is naturally satisfied in many applications and is also needed for the implementation of the proposed numerical method. We will also handle the case in which $f$ and $F$ are discontinuous in Section III.

Policy and Cost-to-go Function: A particular class of admissible controls, called Markov controls, depends only on the current state, i.e., $u(t)$ is a function only of $x(t)$, for all $t \geq 0$. It is well known that in control problems with full state information, the best Markov control performs as well as the best admissible control (see, e.g., [20], [21]). A Markov control defined on $S$ is also called a policy, and is represented by the function $\mu : S \rightarrow U$. The set of all policies is denoted by $\Pi$. Define the first exit time $T_\mu : \Pi \rightarrow [0, +\infty]$ under policy $\mu$ as

$$T_\mu = \inf \left\{ t : x(t) \notin S^o \mbox{ and Eq. (1) and } u(t) = \mu(x(t)) \right\}.$$ Intuitively, $T_\mu$ is the first time that the trajectory of the dynamical system given by Eq. (1) with $u(t) = \mu(x(t))$ hits the boundary $\partial S$ of $S$. By definition, $T_\mu = +\infty$ if $x(\cdot)$ never exits $S^o$. Clearly, $T_\mu$ is a random variable. Then, the expected cost-to-go function under policy $\mu$ is a mapping from $S$ to $\mathbb{R}$ defined as

$$J_\mu(z) = \mathbb{E}_z \left[ \int_0^{T_\mu} \alpha \, g(x(t), \mu(x(t))) \, dt + h(x(T_\mu)) \right],$$

where $\mathbb{E}_z$ denotes the conditional expectation given $x(0) = z, g : S \times U \rightarrow \mathbb{R}$ and $h : S \rightarrow \mathbb{R}$ are bounded measurable and continuous functions, called the cost rate function and the terminal cost function, respectively, and $\alpha \in [0, 1)$ is the discount rate. We further assume that $g(x, u)$ is uniformly Hölder continuous [22] in $x$ with exponent $2\rho \in (0, 1]$. We will address the discontinuity of $g$ and $h$ in Section III. The optimal cost-to-go function $J^* : S \rightarrow \mathbb{R}$ is defined as $J^*(z) = \inf_{\mu \in \Pi} J_\mu(z)$ for all $z \in S$. A policy $\mu^*$ is called optimal if $J_{\mu^*} = J^*$. For any $\varepsilon > 0$, a policy $\mu$ is called an $\varepsilon$-optimal policy if $||J_\mu - J^*||_{\infty} \leq \varepsilon$.

In this paper, we consider the problem of computing the optimal cost-to-go function $J^*$ and an optimal policy $\mu^*$ if
obtainable. Our approach, outlined in Section IV, approximates the optimal cost-to-go function and an optimal policy in an anytime fashion using incremental sampling-based algorithms. This sequence of approximations is guaranteed to converge uniformly to the optimal cost-to-go function and to find an ε-optimal policy for an arbitrarily small non-negative ε in a suitable sense, as the number of samples approaches infinity.

We emphasize that the above generic formulation extends the classical motion planning problem in [17]. By setting appropriate terminal cost h(·) for goal and obstacle regions, we can control a system to reach a goal region while avoiding collision with obstacles in stochastic environments.

To approximate this continuous stochastic dynamics with discrete structures, we use the Markov chain approximation method described in the next section.

### III. MARKOV CHAIN APPROXIMATION

A discrete-state Markov decision process (MDP) is a tuple $\mathcal{M} = (X, A, P, G, H)$ where $X$ is a finite set of states, $A$ is a set of actions that is possibly a continuous space, $P(\cdot | \cdot, \cdot) : X \times X \times A \rightarrow \mathbb{R}_{\geq 0}$ is a function that denotes the transition probabilities satisfying $\sum_{z \in X} P(z | x, a) = 1$ for all $x \in X$ and all $a \in A$, $G(\cdot, \cdot) : X \times X \rightarrow \mathbb{R}$ is an immediate cost function, and $H : X \rightarrow \mathbb{R}$ is a terminal cost function. If we start at time 0 with a state $x_0 \in X$, and at time 0, we apply an action $a_0 \in A$ at a state $x_0$ to arrive at a next state $x_1 \in X$ according to the transition probability function $P(\cdot | \cdot, \cdot)$, we have a controlled Markov chain $\{x_i; i \in \mathbb{N}\}$ due to the control sequence $\{a_i; i \in \mathbb{N}\}$ and an initial state $x_0$ which will also be called the trajectory of $\mathcal{M}$ under the said sequence of controls and initial state.

Given a continuous-time dynamical system as described in Eq. (1), the Markov chain approximation method approximates the continuous stochastic dynamics using a sequence of MDPs $\{\mathcal{M}_n\}_{n=0}^{\infty}$ in which $\mathcal{M}_n = (S_n, A_n, P_n, G_n, H_n)$ where $S_n$ is a discrete subset of $S$ and $U$ is the original control set. We define $\partial S_n = \partial S \cap S_n$. For each $n \in \mathbb{N}$, let $\{\xi_n^i; i \in \mathbb{N}\}$ be a controlled Markov chain on $\mathcal{M}_n$ until it hits $\partial S_n$. We associate with each state $z$ in $S$ a non-negative interpolation interval $\Delta t_n(z)$, known as a holding time.

We use $\Delta t_n$ as the short form of $\Delta t_n(z)$. We define $\Delta t_n = \sum_{i=0}^{n-1} \Delta t_i$ for $i \geq 1$ and $t_0 = 0$. Let $\xi_n^i = \xi_{n+1} - \xi_i$. In addition, we define $G_n(z, v) = g(z, v)\Delta t_n(z)$ and $H_n(z) = h(z)$ for each $z \in S_n$ and $v \in U$. Let $\Omega_n$ be the sample space of $\mathcal{M}_n$. Let $u_n(x)$ denote the control used at step $i$ for the controlled Markov chain. Holding times $\Delta t_n$ and transition probabilities $P_n$ are chosen to satisfy the local consistency property given by the following conditions:

- For all $z \in S$, $\lim_{n \rightarrow \infty} \Delta t_n(z) = 0$.
- For all $z \in S$ and all $v \in U$:
  
  $$\lim_{n \rightarrow \infty} \mathbb{E}_{P_n}[\Delta t_n^0 | \xi_n^0 = z, u_n^0 = v] = f(z, v),$$
  $$\lim_{n \rightarrow \infty} \text{Cov}_{P_n}[\Delta t_n^0 | \xi_n^0 = z, u_n^0 = v] = F(z, v)F(z, v)^T,$$
  $$\lim_{n \rightarrow \infty} \sup_{\xi_n^0 \in [0,\infty)} ||\Delta t_n^0||_2 = 0.$$

The chain $\{\xi_n^i; i \in \mathbb{N}\}$ is a discrete-time process. In order to approximate the continuous-time process $x(\cdot)$ in Eq. (2), we use an approximate continuous-time interpolation. We define the (random) continuous-time interpolation $\xi^0(\cdot)$ of the chain $\{\xi_n^i; i \in \mathbb{N}\}$ and the continuous-time interpolation $u_n^0(\cdot)$ of the control sequence $\{u_n^i; i \in \mathbb{N}\}$ under the holding times function $\Delta t_n$ as follows: $\xi^0(\tau) = \xi^0_\tau$, and $u_n^0(\tau) = u_n^\tau$ for all $\tau \in [\tau^0_n, \tau^1_n+1]$. Let $D^\mathcal{M}_n([0, +\infty))$ denote the set of all $\mathbb{R}^{dx}$-valued functions that are continuous from the left and has limits from the right. The process $\xi^0_n$ can be thought of as a random mapping from $\Omega_n$ to the function space $D^\mathcal{M}_n([0, +\infty))$.

A control problem for the MDP $\mathcal{M}_n$ is analogous to that defined in Section II. Similar to previous section, a policy $\mu_n$ is a function that maps each state $z \in S_n$ to a control $\mu_n(z) \in U$. The set of all such policies is $\Pi_n$. Given a policy $\mu_n$, the (discounted) cost-to-go due to $\mu_n$ is:

$$J_{n, \mu_n}(z) = \mathbb{E}^n_{P_n} \left[ \sum_{i=0}^{\tau_n f} \alpha_i G_n(\xi^0_i, \mu_n(u_n^0)) + \alpha^{\tau_n f} H_n(\xi^0_{\tau_n f}) \right],$$

where $\mathbb{E}^n_{P_n}$ denotes the conditional expectation given $\xi^0_n = z$ under $P_n$, and $\{\xi^0_i; i \in \mathbb{N}\}$ is the sequence of states of the controlled Markov chain under the policy $\mu_n$ and $I_n$ is termination time defined as $I_n = \min \{i; \xi^0_i \in \partial S_n\}$.

The optimal cost function, denoted by $J^*_{n}(z) = \inf_{\mu_n \in \Pi_n} J_{n, \mu_n}(z)$, for $z \in S_n$. An optimal policy, denoted by $\mu^*_{n}$, satisfies $J^*_{n}(z) = J_{n, \mu^*_{n}}(z)$ for all $z \in S_n$. For any $\epsilon > 0$, $\mu^*_n$ is an $\epsilon$-optimal policy if $|J_{n, \mu^*_{n}} - J^*_{n}| \leq \epsilon$.

As stated in the next theorems, local consistency implies (i) the convergence of continuous-time interpolations of the trajectories of the controlled Markov chain to the trajectories of the stochastic dynamical system described by Eq. (1), and (ii) the convergence of optimal cost-to-go of discrete MDPs to the optimal cost-to-go of the original problem.

Theorem 1 (see Theorem 10.4.1 in [18]) Let $f(\cdot, \cdot)$ and $F(\cdot, \cdot)$ be defined as in Section II, and thus Eq. (1) has a weakly unique solution. Let $\{\mathcal{M}_n\}_{n=0}^{\infty}$ be a sequence of MDPs, and $\{\Delta t_n\}_{n=0}^{\infty}$ be a sequence of holding times that are locally consistent with the stochastic dynamical system described by Eq. (1). Let $\{u_n^0; i \in \mathbb{N}\}$ be a sequence of controls defined for each $n \in \mathbb{N}$. For all $n \in \mathbb{N}$, let $\{\xi_n^0(t); t \in \mathbb{R}_{\geq 0}\}$ denote the continuous-time interpolation to the chain $\{\xi_n^i; i \in \mathbb{N}\}$ under the control sequence $\{u_n^i; i \in \mathbb{N}\}$ starting from $z_{n, \text{init}}$, and $\{u_n^0(t); t \in \mathbb{R}_{\geq 0}\}$ denote the continuous-time interpolation of $\{u_n^i; i \in \mathbb{N}\}$, according to the holding time $\Delta t_n$. The sequence of $\{\xi_n^0(t); u_n^0(t)\}_{n=0}^{\infty}$ then converges in distribution to $(x(\cdot), u(\cdot))$ with

$$x(t) = z_{n, \text{init}} + \int_0^t f(x(\tau), u(\tau))d\tau + \int_0^t F(x(\tau), u(\tau))dw(\tau).$$

Theorem 2 (see Theorem 10.5.2 in [18], Theorem 2.3 in [23], and Theorem 2.1 in [24]) Let $f(\cdot, \cdot)$, $F(\cdot, \cdot)$, $g(\cdot, \cdot)$ and $h(\cdot)$ be defined as in Section II. For any trajectory $x(\cdot)$ of the system described by Eq. (1), define $\hat{x}(t) := \inf \{t; x(t) \notin S\}$. Let $\{\mathcal{M}_n = (S_n, U, P_n, G_n, H_n)\}_{n=0}^{\infty}$ and $\{\Delta t_n\}_{n=0}^{\infty}$ be locally consistent with the system described by Eq. (1).

We suppose that the function $\tau(\cdot)$ is continuous (as a mapping from $D^\mathcal{M}_n([0, +\infty))$ to the compactified interval $[0, +\infty]$ with probability one relative to the measure induced by any solution to Eq. (1) for an initial state $z$, which is
satisfied when the matrix $F(\cdot, \cdot)F(\cdot, \cdot)^T$ is nondegenerate. Then, due to the Hölder continuity of $g(\cdot, \cdot)$, we have:
\[
\lim_{n \to \infty} \sup_{z \in S_n} |J_n^z(z) - J^z(z)| = 0.
\]

Discontinuity of dynamics and objective functions

We note that the above theorems continue to hold even when the functions $f, F, g$, and $h$ are discontinuous. In this case, the following conditions are sufficient to use the theorems: (i) For $r$ to be $f, F, g$, or $h, r(x, u)$ takes either the form $r_0(x) + r_1(u)$ or $r_0(x)r_1(u)$ where the control dependent terms are continuous and the $x$-dependent terms are measurable, and (ii) $f(\cdot, \cdot), F(\cdot, \cdot), g(\cdot, \cdot)$, and $h(x)$ are nondegenerate for each $x$, and the set of discontinuity in $x$ of each function is a uniformly smooth surface of lower dimension. Furthermore, instead of uniform Hölder continuity, the cost rate $g$ can be relaxed to be locally Hölder continuous with exponent $2\rho$ on $S$ (see, e.g., page 275 in [18]).

The above theorems assert the asymptotic optimality given a sequence of $a$ priori discretizations of the state space and the availability of $c$-optimal policies. In what follows, we describe an algorithm that incrementally generates a sequence of MDPs along with holding times such that induced approximating Markov chains are locally consistent with the original process. This algorithm also computes an optimal policy for each chain in an incremental manner.

IV. THE iMDP ALGORITHM

Based on Markov chain approximation results, the iMDP algorithm incrementally builds a sequence of discrete MDPs with probability transitions and cost functions that consistently approximate the original continuous counterparts. The algorithm refines the discrete models by using a number of primitive procedures to add new states into the current approximate model. Finally, the algorithm improves the quality of discrete-model policies in an iterative manner by effectively using the computations inherited from the previous iterations. Before presenting the algorithm, some primitive procedures on which the algorithm relies are presented in this section.

A. Primitive Procedures

1) Sampling: The Sample() and SampleBoundary() procedures sample states independently and uniformly from the interior $S^o$ and the boundary $\partial S$, respectively.

2) Nearest Neighbors: Given $z \in S$ and a set $Y \subseteq S$ of states. For any $k \in \mathbb{N}$, the procedure Nearest($z, Y, k$) returns the $k$ nearest states $z' \in Y$ that are closest to $z$ in terms of the Euclidean norm.

3) Time Intervals: Given a state $z \in S$ and a number $k \in \mathbb{N}$, the procedure ComputeHoldingTime($z, k$) returns a holding time computed as $\gamma t \left( \frac{\log k}{k} \right)^{\beta_{\mu / \delta}}$, where $\gamma t > 0$ is a constant, and $\beta, \theta$ are constants in $(0, 1)$ and $(0, 1]$ respectively.

4) Transition Probabilities: Given a state $z \in S$, a subset $Y \subseteq S$, a control $v \in U$, and a positive number $\tau$ describing a holding time, the procedure ComputeTransProb($z, v, \tau, Y$) returns (i) a finite set $Z_{\text{near}} \subseteq S$ of states such that the state $z + f(z, v)\tau$ belongs to the convex hull of $Z_{\text{near}}$ and $||z' - z|| = O(\tau)$ for all $z' \neq z \in Z_{\text{near}}$, and (ii) a function $p$ that maps $Z_{\text{near}}$ to a non-negative real numbers such that $p(\cdot)$ is a probability distribution over the support $Z_{\text{near}}$. It is crucial to ensure that these transition probabilities result in a sequence of locally consistent chains in the algorithm.

There are several ways to construct such transition probabilities. One possible construction by solving a system of linear equations can be found in [18]. In particular, we choose $Z_{\text{near}} = \text{Nearest}(z + f(z, v)\tau, Y, s)$ where $s \in \mathbb{N}$ is some constant. We define the transition probabilities $p: Z_{\text{near}} \to \mathbb{R}_{\geq 0}$ such that satisfies: (i) $\sum_{z' \in \text{near}} p(z'|z) = f(z, v)\tau + o(\tau)$, (ii) $\sum_{z' \in \text{near}} p(z'|z)(z' - z) = F(z, v)F(z, v)^T\tau + f(z, v)f(z, v)^T\tau^2 + o(\tau)$.

An alternative way to compute the transition probabilities is to approximate using local Gaussian distributions. We choose $Z_{\text{near}} = \text{Nearest}(z + f(z, v)\tau, Y, s)$ where $s = \Theta(\log(|Y|))$. Let $N_{\mu, \sigma}(\cdot)$ denote the density of the (possibly multivariate) Gaussian distribution with mean $\mu$ and variance $\sigma$. Define the transition probabilities as follows:

$$p(z') = \sum_{z \in \text{near}} N_{\mu(z'), \sigma(z')}$$

An alternative way to compute the transition probabilities is to approximate using local Gaussian distributions. We choose $Z_{\text{near}} = \text{Nearest}(z + f(z, v)\tau, Y, s)$ where $s = \Theta(\log(|Y|))$. Let $N_{\mu, \sigma}(\cdot)$ denote the density of the (possibly multivariate) Gaussian distribution with mean $\mu$ and variance $\sigma$. Define the transition probabilities as follows:

$$p(z') = \sum_{z \in \text{near}} N_{\mu(z'), \sigma(z')}$$

5) Backward Extension: Given $T > 0$ and two states $z, z' \in S$, the procedure ExtendBackwards($z, z', T$) returns a triple $(x, v, \tau)$ such that (i) $x(t) = f(x(t), u(t))dt$ and $u(t) = v \in U$ for all $t \in [0, \tau]$, (ii) $\tau \leq T$, (iii) $x(t) \in S$ for all $t \in [0, \tau]$. If no such trajectory exists, then the procedure returns failure.

An alternative way to compute the transition probabilities is to approximate using local Gaussian distributions. We choose $Z_{\text{near}} = \text{Nearest}(z + f(z, v)\tau, Y, s)$ where $s = \Theta(\log(|Y|))$. Let $N_{\mu, \sigma}(\cdot)$ denote the density of the (possibly multivariate) Gaussian distribution with mean $\mu$ and variance $\sigma$. Define the transition probabilities as follows:

$$p(z') = \sum_{z \in \text{near}} N_{\mu(z'), \sigma(z')}$$

6) Sampling and Discovering Controls: The procedure ConstructControls($k, z, Y, T$) returns a set of $k$ controls in $U$. We can uniformly sample $k$ controls in $U$. Alternatively, for each state $z' \in \text{Nearest}(z, Y, k)$, we solve for a control $v \in U$ such that (i) $x(t) = f(x(t), u(t))dt$ and $u(t) = v \in U$ for all $t \in [0, T]$, (ii) $x(t) \in S$ for all $t \in [0, T]$, (iii) $x(0) = z$ and $x(T) = z'$.

B. Algorithm Description

The iMDP algorithm is given in Algorithm 1. The algorithm incrementally refines a sequence of (finite-state) MDPs $\mathcal{M}_n = (S_n, U, P_n, G_n, H_n)$ and the associated holding time function $\Delta n_z$ that consistently approximates the system in Eq. (1). In particular, given a state $z \in S_n$, and a holding time $\Delta n_z$, we can implicitly define the state cost function $G_n(z, v) = \Delta n_z(z)(g(z, v) for all $v \in U$ and terminal cost function $H_n(z) = h(z)$. We also associate with $z \in S_n$, a cost value $J_n(z)$, and a control $\mu_n(z)$. We refer to $J_n$ as a cost value function over $S_n$. In the following discussion, we describe how to construct $S_n, P_n, J_n, \mu_n$ over iterations.

We note that, in most cases, we only need to construct and access $P_n$ on demand. In every iteration of the main loop (Lines 4-16), we sample an additional state from the boundary of the state space $S$. We set $J_n, \mu_n, \Delta n_z$ for those
generally, we can uniformly sample the set of controls, called $\bar{U}_n$, in the control space $U$. Hence, we can evaluate the right hand side (RHS) of Eq. (3) for each $v \in \bar{U}_n$ to find the best $v^*$ in $U_n$ with the smallest RHS value and thus to update $J_n(z)$ and $\mu_n(z)$. When $\lim_{n \to \infty} |U_n| = \infty$, we can solve Eq. (3) arbitrarily well.

Thus, it is sufficient to construct the set $U_n$ with $\Theta(\log(|S_n|))$ controls using the procedure ConstructControls as described in Algorithm 2 (Line 2). The set $Z_{\text{near}}$ and the transition probability $P_n(\cdot; z, v)$ constructed consistently over the set $Z_{\text{near}}$ are returned from the procedure ComputeTranProb for each $v \in U_n$ (Line 4). Depending on a particular method to build $P_n$ (i.e., solving a system of linear equations or sampling a Gaussian distribution), the cardinality of $Z_{\text{near}}$ is set to a constant or increases as $\Theta(\log(|S_n|))$. Subsequently, the procedure chooses the best control among the constructed controls to update $J_n(z)$ and $\mu_n(z)$ (Line 7).

**Complexity of \text{iMDP}**

The time complexity per iteration of the implementation in Algorithms 1-2 is either $O(|S_n|^\theta \log |S_n|)$ or $O(|S_n|^\theta (\log |S_n|)^2)$ depending on the size of $Z_{\text{near}}$ in ComputeTranProb, which is either constant or $\Theta(\log(|S_n|))$. Although computing local Gaussian approximation yields higher time complexity, this approximation is more convenient to compute. The processing time from the beginning until the iMDP algorithm stops after $n$ iterations is thus either $O(|S_n|^{1+\theta} \log |S_n|)$ or $O(|S_n|^{1+\theta} (\log |S_n|)^2)$. The space complexity of the iMDP algorithm is $O(|S_n|)$ where $|S_n| = \Theta(n)$ due to our sampling strategy.

**C. Feedback Control**

Given a level of approximation $n \in \mathbb{N}$, the control policy $\mu_n$ generated by the iMDP algorithm is used for controlling the original system described by Eq. (1) using the procedure given in Algorithm 3. This procedure computes the state in $\mathcal{M}$ that is closest to the current state of the original system, and applies the control attached to this closest state over the associated holding time.
V. ANALYSIS

In this section, we analyze the iMDP algorithm. For brevity, we exclude the proofs of lemmas and theorems. An extended version of this paper is [26], where the detailed proofs of the main results can be found.

Let $\mathcal{M}_n = (S_n, U_n, F_n, G_n, H_n, J_n, \mu_n)$ denote the MDP, cost value function, and policy returned by Algorithm 1 at the end $n$ iterations. Let $A_n$ denote the event that the approximating controlled Markov chain on $\mathcal{M}_n$ (under a control sequence and an initial state) is locally consistent with respect to the stochastic dynamical system (1).

**Lemma 3** The approximating chain on $\mathcal{M}_n$ is locally consistent for large $n$, with probability one, i.e., $P(\liminf_{n \to \infty} A_n) = 1$.

The proof is based on the fact that, if we partition $\mathbb{R}^d_z$ into cells of volume $O(\log(|S_n|)/|S_n|)$, then, almost surely, every cell contains at least an element of $S_n$, as $|S_n|$ approaches infinity. Theorem 1 and Lemma 3 together imply that the trajectories of the controlled Markov chains approximate those of the original stochastic dynamical system in Eq. (1) arbitrarily well as $n$ approaches infinity. In addition, Theorem 2 and Lemma 3 guarantee that discrete optimal cost functions $J_n^*$ converges uniformly to the optimal cost-to-go $J^*$ of the original problem with probability one (w.p.1). The next theorem presents stronger results.

**Theorem 4** For all $z \in S_n$, $J_n(z)$ is the cost value of the state $z$ computed by Algorithm 1 and Algorithm 2 after $n$ iterations with $1 \leq L_n$, and $K_n = \Theta(|S_n|^\alpha) < |S_n|$. Let $J_{n,\mu_n}$ be the cost-to-go function of the returned policy $\mu_n$ on the discrete MDP $\mathcal{M}_n$. If the Bellman update at Eq. 3 is solved exactly, then $J_n$ converges uniformly to $J^*$ w.p.1. Otherwise, if the Bellman update at Eq. 3 is solved by sampling $|U_n|$ controls such that $\lim_{n \to \infty} |U_n| = |S_n|$, then $J_n$ converges uniformly to $J^*$ in probability.

Theorem 4 enables an incremental computation of the optimal cost $J^*$ without the need to compute $J_n^*$ exactly before processing more samples.

VI. EXPERIMENTS

We used a computer with a 2.0-GHz-Intel-Core-2-Duo-T6400 processor and 4 GB of RAM to run experiments. In the first experiment, we investigated the convergence of the iMDP algorithm on a stochastic LQR problem: \( \inf \mathbb{E} \left[ \int_0^\tau 0.95 \left\{ 3.5 x(t)^2 + 200 u(t)^2 \right\} dt + h(x(\tau)) \right] \) such that \( dx = (3x + 11u)dt + \sqrt{2}dw \) on the state space \( S = [-6, 6] \) where \( \tau \) is the first hitting time to the boundary of \( S \) = \{-6, 6\}, and \( h(z) = 414.55 \) for \( z \in S \) and 0 otherwise. The optimal cost-to-go from \( x(0) = z \) is 10.39z^2 + 40.51, and the optimal control policy is \( u(t) = -0.571 z(t) \). Since the cost-rate function is bounded on \( S \) and Hölder continuous with exponent 1.0, we use \( \rho = 0.5 \). In addition, we choose \( \theta = 0.5 \), and \( \zeta = 0.99 \) in the procedure ComputeHoldingTime. We used the procedure Update as presented in Algorithm 2 with \( \log(n) \) sampled controls and transition probabilities having constant support size. Figures 1(a)-1(c) show the convergence of approximated cost-to-go, anytime controls and trajectory to the optimal analytical counterparts over iterations. The log-log plot in Fig. 1(d) clearly indicates that both mean and standard deviation of the error \( \sup_{z \in S_n} |J_n(z) - J^*(z)| \) continue to decrease. This observation is consistent with Theorems 4. Finally, Fig. 1(e) plots the ratio of running time per iteration \( T_n \) to \(|S_n|^{0.5} \log(|S_n|)\) asserting that the time complexity per iteration is \( O(|S_n|^{0.5} \log(|S_n|)) \).

In the second experiment, we controlled a system with stochastic single integrator dynamics to a goal region with free ending time in a cluttered environment. The cost objective function is discounted with \( \alpha = 0.95 \). The system pays zero cost for each action it takes and pays a cost of -1 when reaching the goal region \( x_{goal} \). The maximum velocity of the system is one. The system stops when it collides with obstacles. We show how the system reaches the goal in the upper right corner and avoids obstacles with different anytime controls. Anytime control policies after up to 10,000 iterations in Figs. 2(a)-2(c), in which acceptable solutions were obtained within 1.2 seconds, indicate that iMDP quickly explores the state space and refines control policies over time. Corresponding contours of cost value functions are shown in Figs. 2(d)-2(f) further illustrate the refinement and convergence of cost value functions to the original optimal cost-to-go over time. We observe that the performance is suitable for real-time control.

In the third experiment, we tested the effect of process noise magnitude on the solution trajectories. In Figs. 3(a)-3(c), the system wants to arrive at a goal area either by passing through a narrow corridor or detouring around the two blocks. In Fig. 3(a), when the dynamics is noise-free (by setting a small diffusion matrix), the iMDP algorithm quickly determines to follow a narrow corridor. In contrast, when the environment affects the dynamics of the system (Figs. 3(b)-3(c)), the iMDP algorithm decides to detour to have a safer route. This experiment demonstrates the benefit of iMDP in handling process noise compared to RRT-like algorithms [14], [17]. We emphasize that although iMDP spends slightly more time on computation per iteration, iMDP provides feedback policies rather than open-loop policies; thus, re-planning is not crucial in iMDP.

In the fourth experiment, we examined the performance of the iMDP algorithm for a manipulator with six degrees of freedom. The manipulator is modeled as a single integrator where states represents angles between segments and the horizontal line. The maximum control magnitude for all joints is 0.3. The standard deviation of noise at each joint is 0.032 rad. The manipulator is controlled to reach a goal with the final upright position in minimum time. In Fig. 4(a), we show a resulting trajectory after 3000 iterations computed in 15.8 seconds. In addition, we show the mean and standard deviation of the computed cost values for the initial position using 50 trials on a semi-log plot in Fig. 4(b). As shown in the plots, the solution is convergent quickly after about 1000 iterations. These results highlight the suitability of iMDP to compute feedback policies for complex high dimensional systems in stochastic environments.
VII. CONCLUSIONS

We have introduced and analyzed the incremental sampling-based iMDP algorithm for stochastic optimal control. The algorithm natively handles continuous time, continuous state space as well as continuous control space. The main idea is to consistently approximate underlying continuous problems by discrete structures in an incremental manner. In particular, we incrementally build discrete MDPs by sampling and extending states in the state space. The iMDP algorithm refines the quality of anytime control policies from discrete MDPs in terms of expected costs over iterations. The iMDP algorithm can be implemented such that its time complexity per iteration grows as $O(k^\theta \log k)$ with $0 < \theta \leq 1$ leading to a total processing time $O(k^{1+\theta} \log k)$, where $k$ is the number of states in MDPs. The enabling technical ideas lie in novel methods to compute Bellman updates.

In the future, we would like to study the effect of biased-sampling techniques on the performance of iMDP. Remarkably, Markov chain approximation methods are also tools to handle deterministic control and non-linear filtering.
and discovers a safer route after a few seconds. (In Fig. 3(b), we temporarily let the system continue even after collision to observe the entire trajectory.)

![Trajectory snapshots](image)

Fig. 3. Performance against different process noise magnitude. The system starts from (0, -5) to reach the goal. In Fig. 3(a), the environment is noise-free. In Figs. 3(b)-(3(c), standard deviation of noise in x and y directions is 0.37. In the latter, the system first discovers an unsafe route that is prone to collisions and discovers a safer route after a few seconds. (In Fig. 3(b), we temporarily let the system continue even after collision to observe the entire trajectory.)

![Mean and standard deviation](image)

Fig. 4. Results of a 6D manipulator example. Control magnitude is bounded by 0.3. The standard deviation of noise at each joint is 0.032 rad. In Fig. 4(b), the mean and standard deviation of the computed cost values for the initial position are plotted using 50 trials.

problems. Thus, we plan to extend the methodology used in iMDP to handle these problems. Finally, although POMDPs are fundamentally more challenging than the problem that is studied in this paper, our approach differentiates itself from existing sampling-based POMDP solvers (see, e.g., [27]) with its incremental nature and computationally-efficient search. Hence, the research presented in this paper opens a new alley to handle POMDPs in our future work.

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation, grant CNS-1016213. V. A. Huynh gratefully thanks the Arthur Gelb Foundation for partially supporting him during this work.

REFERENCES