A Martingale Approach and Time-Consistent Sampling-based Algorithms for Risk Management in Stochastic Optimal Control

Vu Anh Huynh  Leonid Kogan  Emilio Frazzoli

Abstract—In this paper, we consider a class of stochastic optimal control problems with risk constraints that are expressed as bounded probabilities of failure for particular initial states. We present here a martingale approach that diffuses a risk constraint into a martingale to construct time-consistent control policies. The martingale stands for the level of risk tolerance over time. By augmenting the system dynamics with the controlled martingale, the original risk-constrained problem is transformed into a stochastic target problem. We extend the incremental Markov Decision Process (iMDP) algorithm to approximate arbitrarily well an optimal feedback policy of the original problem by sampling in the augmented state space and computing proper boundary conditions for the reformulated problem. We show that the algorithm is both probabilistically sound and asymptotically optimal. The performance of the proposed algorithm is demonstrated on motion planning and control problems subject to bounded probability of collision in uncertain cluttered environments.

I. INTRODUCTION

Controlling dynamical systems in uncertain environments is a fundamental and essential problem in several fields, ranging from robotics [1], [2], healthcare [3], [4] to management science, economics and finance [5], [6]. Given a system with dynamics described by a controlled diffusion process, a stochastic optimal control problem is to find an optimal feedback policy to optimize an objective function. Risk management has always been an important part of stochastic optimal control problems to guarantee safety during the execution of control policies. For instance, in urban navigation applications, it is desirable that autonomous cars depart from origins to reach destinations with minimum energy and at the same time maintain bounded probability of collision. Ensuring such performance is critical before deploying autonomous cars in real life.

There has been intensive literature on stochastic optimal control without risk constraints. Even in this setting, it is well-known that closed-form or exact algorithmic solutions for general continuous-time, continuous-space stochastic optimal control problems are computationally challenging [7]. Thus, many approaches have been proposed to investigate approximate solutions of such problems. Deterministic approaches such as discrete Markov Decision Process approximation [8], [9] and solving associated Hamilton-Jacobi-Bellman PDEs [10]–[12]) have been proposed, but the complexities of these approaches scale poorly with the dimension of the state space. In [7], [13], [14], the authors show that randomized algorithms (or sampling-based algorithms) provide a possibility to alleviate the curse of dimensionality by sampling the state space while assuming discrete control inputs. Recently, in [15], [16], a new computationally-efficient sampling-based algorithm called the incremental Markov Decision Process (iMDP) algorithm has been proposed to provide asymptotically-optimal solutions to problems with continuous control spaces.

Built upon the approximating Markov chain method [17], [18], the iMDP algorithm constructs a sequence of finite-state Markov Decision Processes (MDPs) that consistently approximate the original continuous-time stochastic dynamics. Using the rapidly-exploring sampling technique [19] to sample in the state space, iMDP forms the structures of finite-state MDPs randomly over iterations. Control sets for states in these MDPs are constructed or sampled properly in the control space. The finite models serve as incrementally refined models of the original problem. Consequently, distributions of approximating trajectories and control processes returned from these finite models approximate arbitrarily well distributions of optimal trajectories and optimal control processes of the original problem. The iMDP algorithm also maintains low time complexity per iteration by asynchronously computing Bellman updates in each iteration. There are two main advantages to use the iMDP algorithm for solving stochastic optimal control problems. First, the iMDP algorithm provides a method to compute optimal control policies without the need to derive and characterize viscosity solutions of associated HJB equations. Second, the algorithm is suitable for various online robotics applications without a priori discretization of the state space.

Risk management in stochastic optimal control have also been received extensive attention by researchers in several fields. In robotics, a common risk management problem is chance-constrained optimization [20]–[22]. Chance constraints specify that starting from a given initial state, the time-$\theta$ probability of success must be above a given threshold where success means reaching goal areas safely. Alternatively, we call these constraints risk constraints if we concern more about failure probabilities. Despite intensive work done to solve this problem in last 20 years, designing computationally-efficient algorithms that respect chance constraints for systems with continuous-time dynamics is still an open question. In particular, most previous works use discrete-time multi-stage models for this problem and take different approaches to tackle it [20]–[23]. The Lagrangian approach [24]–[26] is a possible method for solving the mentioned constrained optimization. However, this approach
requires numerical procedures to compute Lagrange multipliers before obtaining a policy, which is often computationally demanding for high dimensional systems. Other works use linear programming [20], [21], but their approach requires the direct representation of convex obstacles into the formulation, which limits its use in practice. Therefore, several authors modify the formulation to have constraints where a static, single-period failure probability threshold (or other appropriate risk metric) is applied to the future stream of costs [27]–[29]. However, this modified formulation leads to potential inconsistent behaviors in which risk preferences change in an irrational manner between periods [30]. This irrational behavior is known as time-inconsistency of control policies. Recognizing this issue, in [22], the authors used Markov dynamic time-consistent risk measures [31]–[33] to assess the risk of future cost stream in a consistent manner and established a dynamic programming equation for this modified formulation. Solving the resulting dynamic programming equation is, however, computationally difficult as it involves functionals as control variables. Very recently, in [34], the authors modify the problem formulation to enforce the risk constraint for all states along executed trajectories. By extending the iMDP algorithm, the authors show that with probability one, the sequence of policies returned from the extended iMDP algorithm is both probabilistically sound and asymptotically optimal. Nevertheless, obtained solutions are more conservative compared to solutions that would be obtained from the original formulation.

In mathematical finance, closely-related problems have been studied in the context of hedging with portfolio constraints where constraints on terminal states are enforced to satisfy almost surely (a.s) or up-to some moment. When constraints on terminal states must be satisfied almost surely, the problems are known as stochastic target problems [35]–[39]. Research in this field focuses on deriving HJB equations for this class of problems, and several analytical tools such as weak dynamic programming [35] and geometric dynamic programming [40], [41] have been recently developed to achieve this goal. These tools allow us to derive HJB equations and find viscosity solutions for a larger class of problems while avoiding measurability issues.

In this paper, we consider the above risk-constrained problems. That is, we investigate stochastic optimal control problems with risk constraints that are expressed in terms of bounded probabilities of failure for particular initial states. We present here a martingale approach to solve these problems such that obtained control policies are time-consistent with the initial threshold of failure probability. The martingale approach enables us to transform a risk-constrained problem into a stochastic target problem. By sampling in the augmented state space and compute proper boundary conditions of the reformulated problem, we extend the iMDP algorithm to compute anytime solutions after a small number of iterations. When more computing time is allowed, the proposed algorithm refines the solution quality in an efficient manner.

The main contribution of this paper is twofold. First, we present a novel martingale approach that fully respects the considered risk constraints for systems with continuous-time dynamics in a time-consistent manner. The approach is a powerful tool to manage risk in several practical robotics applications without directly deriving HJB equations, which are hard to obtain in many situations. Second, we propose a computationally-efficient algorithm that guarantees probabilistically-sound and asymptotically-optimal solutions to the stochastic optimal control problem in the presence of risk constraints. That is, all constraints are satisfied in a suitable sense, and the objective function is minimized as the number of iterations approaches infinity. We demonstrate the effectiveness of the proposed algorithm on motion planning and control problems subject to bounded collision probability in uncertain cluttered environments.

This paper is organized as follows. A formal problem definition is given in Section II. In Section III, we discuss the martingale approach and the key transformation. The extended iMDP algorithm is described in Section IV. The analysis of the proposed algorithm is presented in Section V. We present simulation examples and experimental results in Section VI and conclude the paper in Section VII.

II. PROBLEM DEFINITION

In this section, we present a generic stochastic optimal control formulation with definitions and technical assumptions as discussed in [15], [16], [34]. We also explain how to formulate risk constraints.

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

Suppose that a stochastic process $\{w(t); t \geq 0\}$ is a $d_w$-dimensional Brownian motion on some probability space. We define $\{F_t; t \geq 0\}$ as the augmented filtration generated by the Brownian motion $w(\cdot)$. Let a control process $\{u(t); t \geq 0\}$ be a $U$-valued, measurable random process also defined on the same probability space such that the pair $(u(\cdot), w(\cdot))$ is admissible [15]. Let the set of all such control processes be $U$. Let $\mathbb{R}^{d_x \times d_w}$ denote the set of all $d_x$ by $d_w$ real matrices. We consider systems with dynamics described by the controlled diffusion process:

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

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 $x(t) \in S^\circ$. The initial state $x(0)$ is a random vector in $S$. We assume that the matrix $F(\cdot, \cdot)$ has full rank. The continuity requirement of $f$ and $F$ can be relaxed with mild assumptions [15], [17] such that we still have a weak solution to Eq. (1) that is unique in the weak sense [42].

**Cost-to-go Function and Risk Constraints:** We define the first exit time $T_u^z : U \times S \rightarrow [0, +\infty]$ under a control process $u(\cdot) \in U$ starting from $x(0) = z \in S$ as

$$T_u^z = \inf \{t : x(0) = z, x(t) \notin S^\circ, \text{ and Eq}(1)\}.$$
In other words, $T_{u}^{z}$ is the first time that the trajectory of the dynamical system given by Eq. 1 starting from $x(0) = z$ hits the boundary $\partial S$ of $S$. The random variable $T_{u}^{z}$ can take value $\infty$ if the trajectory $x(\cdot)$ never exits $S^\circ$.

The expected cost-to-go function under a control process $u(\cdot)$ is a mapping from $S$ to $\mathbb{R}$ defined as

$$J_u(z) = \mathbb{E}_z^{\eta} \left[ \int_0^{T_u^z} \alpha g(x(t), u(t)) \, dt + \alpha^{T_u^z} h(x(T_u^z)) \right],$$

(2)

where $\mathbb{E}_z^{\eta}$ denotes the conditional expectation given $x(t) = z$, and $g : S \times U \to \mathbb{R}$, $h : S \to \mathbb{R}$ 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 in $x$ with exponent $2\rho \in (0, 1]$ for all $u \in U$. We note that the discontinuity of $g, h$ can be treated as in [15], [17].

Let $\Gamma \subseteq \partial S$ be a set of failure states, and $\eta \in [0, 1]$ be a threshold for risk tolerance given as a parameter. We consider a risk constraint that is specified for an initial state $x(0) = z$ under a control process $u(\cdot)$ as follows:

$$\Pr \left[ \mathbb{E}_z^{\eta} \left[ \int_0^{T_u^z} \alpha g(x(t), u(t)) \, dt + \alpha^{T_u^z} h(x(T_u^z)) \right] \right] \leq \eta,$$

(3)

where $\Pr \left[ \mathbb{E}_z^{\eta} \left[ \int_0^{T_u^z} \alpha g(x(t), u(t)) \, dt + \alpha^{T_u^z} h(x(T_u^z)) \right] \right]$ denotes the conditional probability at time $t$ given $x(t) = z$. That is, controls that drive the system from time 0 until the first exit time must be consistent with the choice of $\eta$ and initial state $z$ at time $t$. Intuitively, the constraint enforces that starting from a given state $z$ at time $t = 0$, if we execute a control process $u(\cdot)$ for $N$ times, when $N$ is very large, there are at most $N\eta$ executions resulting in failure. Control processes $u(\cdot)$ that satisfy this constraint are called time-consistent.

Let $\mathbb{R}$ be the extended real number set. The optimal cost-to-go function $J^*: S \to \mathbb{R}$ is defined as follows:

$$J^*(z; \eta) = \inf_{u(\cdot) \in U} J_u(z) \text{ s.t. } \Pr \left[ \mathbb{E}_z^{\eta} \left[ \int_0^{T_u^z} \alpha g(x(t), u(t)) \, dt + \alpha^{T_u^z} h(x(T_u^z)) \right] \right] \leq \eta \text{ and Eq. 7}$$

(4)

A control process $u^*(\cdot)$ is called optimal if $J_u^*(z) = J^*(z; \eta)$. For any $\epsilon > 0$, a control process $u(\cdot)$ is called an $\epsilon$-optimal policy if $|J_u(z) - J^*(z; \eta)| \leq \epsilon$.

We call a sampling-based algorithm probabilistically-sound if the probability that a solution returned by the algorithm is feasible approaches one as the number of samples increases. We also call a sampling-based algorithm asymptotically-optimal if the sequence of solutions returned from the algorithm converges to an optimal solution in probability as the number of samples approaches infinity. Solutions returned from algorithms with such properties are called probabilistically-sound and asymptotically-optimal.

In this paper, we consider the problem of computing the optimal cost-to-go function $J^*$ and an optimal control process $u^*$ if obtainable. Our approach, outlined in Section IV, approximates the optimal cost-to-go function and an optimal policy in an anytime fashion using an incremental sampling-based algorithm that is both probabilistically-sound and asymptotically-optimal.

### III. Martingale Approach

We present the martingale approach that transforms the considered risk-constrained problem into an equivalent stochastic target problem. The following lemma to diffuse risk constraints is a key tool for our transformation.

#### A. Diffusing Risk Constraints

**Lemma 1 (see [38], [39])** From $x(0) = z$, a control process $u(\cdot)$ is feasible if and only if there exists a square-integrable (but unbounded) process $c(\cdot) \in \mathbb{R}^d$, and a martingale $q(\cdot)$ satisfying:

1. $q(0) = \eta$, and $dq(t) = c^T(t)dw(t)$,
2. For all $t$, $q(t) \in [0, 1]$ a.s.
3. $1_{x(T_u^z) \in \Gamma} \leq q(T_u^z)$ a.s.

The martingale $q(t)$ stands for the level of risk tolerance at time $t$. We call $c(\cdot)$ a martingale control process.

**Proof:** Assuming that there exists $c(\cdot)$ and $q(\cdot)$ as above, due to the martingale property of $q(\cdot)$, we have:

$$\Pr \left[ \mathbb{E}_z^{\eta} \left[ \int_0^{T_u^z} \alpha g(x(t), u(t)) \, dt + \alpha^{T_u^z} h(x(T_u^z)) \right] \right] \leq \mathbb{E} \left[ q(T_u^z) \right] = q(0) = \eta.$$

Thus, $u(\cdot)$ is feasible.

Now, let $u(\cdot)$ be a feasible control policy. Set $\eta_0 = \Pr \left[ \mathbb{E}_z^{\eta} \left[ \int_0^{T_u^z} \alpha g(x(t), u(t)) \, dt + \alpha^{T_u^z} h(x(T_u^z)) \right] \right]$. We note that $\eta_0 \leq \eta$. We define the martingale

$$\overline{q}(t) = \mathbb{E} \left[ 1_{x(T_u^z) \in \Gamma} | F_t \right].$$

Since $\overline{q}(T_u^z) \in [0, 1]$, we infer that $\overline{q}(t) \in [0, 1]$ almost surely. We now set

$$\tilde{q}(t) = \overline{q}(t) + (\eta - \eta_0),$$

then $\tilde{q}(t)$ is a martingale with $\tilde{q}(0) = \eta(0) + (\eta - \eta_0) = \eta_0 + (\eta - \eta_0) = \eta$ and $\tilde{q}(t) \geq 0$ almost surely.

Now, we define $\tau = \inf\{ t \in [0, T_u^z) | \tilde{q}(t) \geq 1 \}$, which is a stopping time. Thus,

$$q(t) = \tilde{q}(t)1_{t \leq \tau} + 1_{t > \tau},$$

as a stopped process of the martingale $\tilde{q}(t)$ at $\tau$, is a martingale with values in $[0, 1]$ a.s.

If $\tau < T_u^z$, we have

$$1_{x(T_u^z) \in \Gamma} \leq 1 = q(T_u^z),$$

and if $\tau = T_u^z$, we have

$$q(T_u^z) = \mathbb{E} \left[ 1_{x(T_u^z) \in \Gamma} | F_{T_u^z} \right] + (\eta - \eta_0) = 1_{x(T_u^z) \in \Gamma} + (\eta - \eta_0) \geq 1_{x(T_u^z) \in \Gamma}.$$

Hence, $q(\cdot)$ also satisfies $1_{x(T_u^z) \in \Gamma} \leq q(T_u^z)$.

The control process $c(\cdot)$ exists due to the martingale representation theorem [43], which yields $dq(t) = c^T(t)dw(t)$. We however note that $c(\cdot)$ is unbounded.

---

1. The semicolon in $J^*(z; \eta)$ signifies that $\eta$ is a parameter.
2. Compared to [34], we consider a larger set of control processes than the set of Markov control processes here. We will restrict again to Markov control processes in the reformulated problem.
B. Stochastic Target Problem

Using the above lemma, we augment the original system dynamics with the martingale \( q(t) \) into the following form:

\[
d \begin{bmatrix} x(t) \\ q(t) \end{bmatrix} = \begin{bmatrix} f(x(t), u(t)) \\ 0 \end{bmatrix} dt + \begin{bmatrix} F(x(t), u(t)) \\ c^2(t) \end{bmatrix} dw(t),
\]

where \((u(.), c(.))\) is the control process of the above dynamics. The initial value of the new state is \((x(0), q(0)) = (z, \eta)\). We will refer to the augmented state space \( S \times \mathbf{R} \) as \( S \) and the augmented controlled space \( U \times \mathbf{R}^d \) as \( \Phi \). We also refer to the nominal dynamics and diffusion matrix of Eq. 5 as \( \Phi_\lambda(x, q, u, c) \) and \( F(q, x, u, c) \) respectively.

It is well-known that in the following reformulated problem, optimal control processes are Markov controls [38, 39, 44]. Thus, let us now focus on the set of Markov controls that depend only on the current state, i.e., \((u(t), c(t))\) is a function only of \((x(t), q(t))\), for all \( t \geq 0 \). A function \( \varphi : S \rightarrow \Phi \) represents a Markov or feedback control policy, which is known to be admissible with respect to the process noise \( w(t) \). Let \( \Psi \) be the set of all such policies \( \varphi \). Let \( \mu : S \rightarrow U \) and \( \kappa : S \rightarrow \mathbf{R}^d \) so that \( \varphi = (\mu, \kappa) \). We rename \( T^*_U \) to \( T^*_\mu \) for the sake of notation clarity. Using these notations, \((\mu, 1)\) is thus a Markov control policy that maps from \( S \) to \( U \). Henceforth, we will use \( \mu(\cdot) \) to refer to \( \mu(\cdot, 1) \) when it is clear from the context. Let \( \Pi \) be the set of all such Markov control policies \( \mu(\cdot) \) on \( S \).

Now, let us rewrite cost-to-go function \( J_\mu(z) \) in Eq. 2 for the threshold \( \eta \) at time \( 0 \) in a new form:

\[
J_\varphi(z, \eta) = E \left[ \int_0^{T^*_\varphi} \alpha t \, g(x(t), \mu(x(t), q(t))) \, dt + \alpha T^*_\varphi h(x(T^*_\varphi)) \right] (x(0) = (z, \eta)).
\]

We therefore transform the risk-constrained problem in Eqs. 3, 4 into a stochastic target problem as follows:

\[
J^*(z, \eta) = \inf_{\varphi \in \Psi} J_\varphi(z, \eta)
\]

subject to \( 1_{x(T^*_\varphi) \epsilon \Gamma} \leq q(T^*_\varphi) \) a.s. and Eq. 5.

The constraint in the above formulation specifies the relationship of random variables at the terminal time as target, and hence the name of this formulation [38, 39]. In this formulation, we solve for feedback control policies \( \varphi \) for all \( (z, \eta) \in S \) instead of a particular choice of \( \eta \) for \( x(0) = z \) at time \( t = 0 \). We note that in this formulation, boundary conditions are not fully specified a priori. In the following subsection, we discuss how to remove the constraint in Eq. 5 by constructing its boundary and computing the boundary conditions.

C. Characterization and Boundary Conditions

The domain of the stochastic target problem is:

\[
D = \{(z, \eta) \in S \mid \exists \varphi \in \Psi, s.t. 1_{x(T^*_\varphi) \epsilon \Gamma} \leq q(T^*_\varphi) \text{ a.s.}\}.
\]

By the definition of the risk-constrained problem, we can see that if \((z, \eta) \in D \) then \((z, \eta') \in D \) for any \( \eta < \eta' \leq 1 \). Thus, for each \( z \in S \), we define

\[
\gamma(z) = \inf \{ \eta \in [0, 1] \mid (z, \eta) \in D \},
\]

as the infimum of risk tolerance at \( z \). Therefore, we also have:

\[
\gamma(z) = \inf_{\eta \in \mathbb{U}} \text{Prob}_0(\xi(T^*_z) \epsilon \Gamma) = \inf_{\eta \in \mathbb{U}} \mathbb{E} \left[ 1_{x(T^*_z) \epsilon \Gamma} \right].
\]

Thus, the boundary of \( D \) is

\[
\partial D = S \times \{ 1 \} \cup \{ (z, \gamma(z)) \mid z \in S \}
\]

\[
\cup \{(z, \eta) \mid z \in \partial S, \eta \in (\gamma(z), 1]\}.
\]

For states in \( \{(z, \eta) \mid z \in \partial S, \eta \in [\gamma(z), 1]\} \), the system stops on \( \partial S \) and takes terminal values according to \( h(\cdot) \).

Now, let \( \eta = 1 \), we notice that \( J^*(z, 1) \) is the optimal cost-to-go from \( z \) for the stochastic optimal problem without the risk constraint. Clearly, \( J^*(z, \eta) = +\infty \) for all \( 0 \leq \eta < \gamma(z) \). From the definition of the problem, we also have:

\[
0 \leq \eta < \eta' \leq 1 \Rightarrow J^*(z, \eta) \geq J^*(z, \eta').
\]

Therefore, we can infer that for augmented states \((x(t), q(t))\) with \( q(t) = 1.0 \), the optimal martingale control \( c^*(t) \) is 0.

The following lemma characterizes the optimal martingale control \( c^*(t) \) for augmented states \((x(t), q(t) = \gamma(x(t)))\).

**Lemma 2** Given the problem definition as in Eqs. 2, 4 when \( q(t) = \gamma(x(t)) \) and \( u(t) \) is chosen, we must have:

\[
c(t)^T = \frac{\partial \gamma}{\partial x(t)} F(x(t), u(t)).
\]

**Proof:** Using the geometric dynamic programming principle [40, 41], we have the following result, for all stopping time \( t \geq t \), when \( q(t) \geq \gamma(x(t)) \), a feasible control policy \( \varphi \in \Psi \) satisfies \( q(t) \geq \gamma(x(t)) \) almost surely.

We assume that \( \gamma(x(t)) \) is a smooth function. Take \( t = t+ \), under a feasible control policy \( \varphi \), we have \( q(t+) \geq \gamma(x(t+)) \) a.s. for all \( t \), and hence \( dq(t) \geq \gamma(x(t+)) \) a.s. By Itô’s lemma, we derive the following relationship:

\[
c(t)^T dw(t) = \frac{\partial \gamma}{\partial x(t)} \left( f(x(t), u(t))dt + F(x(t), u(t))dw(t) \right) + \frac{1}{2} \text{Tr} \left( F(x(t), u(t)) F(x(t), u(t))^T \frac{\partial^2 \gamma}{\partial x^2} \right) dt \text{ a.s.}
\]

The above inequality to hold almost surely, the coefficient of \( dw(t) \) must be 0. This leads to Eq. 12.

In addition, if a control process that solves Eq. 10 is obtainable, say \( u^* \), then we have \( J^*(z, \gamma(z)) = J_{u^*}(z) \). We henceforth denote \( J^*(z, \gamma(z)) \) as \( J^*(z) \). We also emphasize that when \((x(t), q(t))\) is inside the interior \( D^0 \) of \( D \), usual dynamic programming principle holds.

3The comma in \( J^*(z, \eta) \) signifies that \( \eta \) is a state component rather than a parameter, and \( J^*(z, \eta) \) is equal to \( J^*(z; \eta) \) in the previous formulation.

4When \( \gamma(x) \) is not smooth, we need the concept of viscosity solutions and weak dynamic programming principle. See [38, 39] for details.
The optimal cost-to-go function $J^*_n : S \rightarrow \mathbb{R}$ that approximates $J^*(z, 1)$ is denoted as

$$J^*_n(z, 1) = \inf_{\mu_n \in \Pi_n} J_{n, \mu_n}(z) \forall z \in S_n. \quad (13)$$

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

In addition, the min-failure probability $\gamma_n$ on $M_n$ that approximates $\gamma(z)$ is defined as:

$$\gamma_n(z) = \inf_{\mu_n \in \Pi_n} \mathbb{E}^\gamma P_n \left[I_{\xi_n^* \in \Gamma} \right] \forall z \in S_n. \quad (14)$$

We note that the optimization programs in Eq. (13) and Eq. (14) may have two different optimal feedback control policies. Let $\nu_n \in \Pi_n$ be a control policy on $M_n$ that achieves $\gamma_n$, then the cost-to-go $J_n^*$ due to $\nu_n$ approximates $J^*(z)$.

Similarly, in the augmented state space $\overline{S}$, we use a sequence of MDPs $\{\overline{M}_n = (\overline{S}_n, \overline{P}_n, \overline{G}_n, \overline{H}_n)\}_{n=0}^\infty$ and a sequence of holding times $\{\Delta t_n\}_{n=0}^\infty$ that are locally consistent with the augmented dynamics in Eq. 5. In particular, $\overline{S}_n$ is a random subset of $D \subset S$, $\overline{G}_n$ is identical to $G_n$, and $\overline{H}_n(z, \eta)$ is equal to $H_n(z)$ if $\eta \in \{\gamma_n(z), 1\}$ and $+\infty$ otherwise. Similar to the construction of $P_n$ and $\Delta t_n$, we also construct the transition probabilities $\overline{P}_n$ on $\overline{M}_n$ and holding time $\overline{H}_n$ that satisfy the local consistency conditions for nominal dynamics $\overline{f}(x, u, c)$ and diffusion matrix $\overline{F}(x, u, c)$. A trajectory on $\overline{M}_n$ is denoted as $\{\xi_n^\gamma, i \in N\}$ where $\xi_n^\gamma \in \overline{S}_n$. A Markov policy $\varphi_n$ is a function that maps each state $(z, \eta) \in \overline{S}_n$ to a control $(\mu_n(z, \eta), \kappa_n(z, \eta)) \in U$. Moreover, admissible $\kappa_n$ at $(z, 1) \in \overline{S}_n$ is $0$ and at $(z, \gamma_n(z)) \in \overline{S}_n$ is a function of $\mu_n(z, \gamma_n(z))$ as shown in Eq. 12. Admissible $\kappa_n$ for other states in $\overline{S}_n$ is such that the martingale-component process of $\{\xi_n^\gamma, i \in N\}$ belongs to $[0, 1]$ almost surely. We can show that equivalently, each control component of $\kappa_n(z, \eta)$ belongs to $[-\frac{\min(n-1, \eta) \min(n, \eta)}{\Delta t_n, \Delta t_{n+1}}, \frac{\min(n, \eta) - \min(n-1, \eta)}{\Delta t_n, \Delta t_{n+1}}]$. The set of such all policies $\varphi_n$ is $\Psi_n$.

Under a control policy $\varphi_n$, the cost-to-go on $\overline{M}_n$ that approximates $\mathbb{E}^\gamma_n$ is defined as:

$$J_{n, \varphi_n}(z, \eta) = \mathbb{E}^\gamma_n \left[\sum_{i=0}^{T_n-1} \alpha^i G_n(\xi_n^\gamma, \mu_n(\xi_n^\gamma)) + \alpha^T_n H_n(\xi_n^\gamma)\right],$$

where $T_n^\gamma = \sum_{i=0}^{T_n-1} \Delta t_n(\xi_n^\gamma)$ for $i \geq 1$ and $T_n^\gamma = 0$ if $i = 0$. Given a policy $\mu_n$ that approximates a Markov control process $u(\cdot)$ in Eq. 2, the corresponding optimal cost $J_n^*: \overline{S}_n \rightarrow \mathbb{R}$ for $J^*$ in Eq. 7 is:

$$J_n^*(z, \eta) = \inf_{\varphi_n \in \Psi_n} J_{n, \varphi_n}(z, \eta) \forall (z, \eta) \in \overline{S}_n. \quad (15)$$

To solve the above optimization, we compute approximate boundary values for states on the boundary of $D$ using the sequence of MDP $\{\overline{M}_n\}_{n=0}^\infty$ on $S$ as discussed above. For states $(z, \eta) \in \overline{S}_n \cap D^c$, the normal dynamic programming principle holds.

The extension of iMDP outlined below is designed to compute the sequence of optimal cost-to-go $\{J^*_n\}_{n=0}^\infty$, min-failure probabilities $\{\gamma_n\}_{n=0}^\infty$ min-failure cost value $\{J^*_n\}_{n=0}^\infty$, and the sequence of anytime control policies $\{\mu_n\}_{n=0}^\infty$ and $\{\kappa_n\}_{n=0}^\infty$ in an efficient iterative procedure.
B. Extension of iMDP

Before presenting the details of the algorithm, we discuss a number of primitive procedures. More details about these procedures can be found in [15, 16].

1) Sampling: The Sample(X) procedure sample states independently and uniformly in X.

2) Nearest Neighbors: Given \( \zeta \in X \subseteq \mathbb{R}^d \) and a set \( Y \subseteq X \), for any \( k \in \mathbb{N} \), the procedure Nearest(\( \zeta, Y, k \)) returns the \( k \) nearest states \( \zeta' \in Y \) that are closest to \( \zeta \) in terms of the \( d \)-dimensional Euclidean norm.

3) Time Intervals: Given a state \( \zeta \in X \) and a number \( k \in \mathbb{N} \), the procedure ComputeHoldingTime(\( \zeta, k, d \)) returns a holding time computed as follows:

\[
\text{ComputeHoldingTime}(\zeta, k, d) = \chi_t \left( \log \frac{k}{\zeta} \right)^{\theta \log d},
\]

where \( \chi_t > 0 \) is a constant, and \( \zeta, \theta \) are constants in \((0,1)\) and \((0,1]\) respectively. The parameter \( \rho \in (0,0.5) \) defines the Hölder continuity of the cost rate function \( g(\cdot, \cdot) \) as in Section [1].

4) Transition Probabilities: We are given a state \( \zeta \in X \), a subset \( Y \subseteq X \), a control \( v \) in some control set \( V \), a positive number \( \tau \) describing a holding time, \( k \) is a nominal dynamics, \( K \) is a diffusion matrix. The procedure ComputeTranProb(\( \zeta, v, \tau, Y, k, K \)) returns (i) a finite set \( Z_{\text{near}} \subseteq X \) of states such that the state \( \zeta + (k, v) \tau \) belongs to the convex hull of \( Z_{\text{near}} \), and \( ||z' - z||_2 = O(\tau) \) for all \( \zeta' \neq \zeta \in Z_{\text{near}} \), 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 that approximate \( k \) and \( K \) as presented in [15]–[17].

5) Backward Extension: Given \( T > 0 \) and two states \( z, z' \in S \), the procedure ExtBackwardsS(\( z, z', T \)) 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) \( x(T) = z \), and (iv) \( x(0) \) is close to \( z' \). When such trajectory exists, the procedure returns failure. We can solve for the triple \((x, v, \tau)\) by sampling several controls \( v \) and choose the control resulting in \( x(0) \) that is closest to \( z' \).

When \((z, \eta, (z', \eta'))\) are in \( S \), the procedure ExtBackwardsSM(\( (z, \eta, (z', \eta'), T) \)) returns \((x, q, v, \tau)\) in which \((x, v, \tau)\) is outer of ExtBackwardsS(\( z, z', T \)) and \( q \) is sampled according to a Gaussian distribution \( N(\eta', \sigma_q) \) where \( \sigma_q \) is a parameter.

6) Sampling and Discovering Controls: For \( z \in S \) and \( Y \subseteq S \), the procedure ConstructControlsS(\( k, z, Y, T \)) 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) \( x(T) = z \) and \( x(T) = z' \).

For \((z, \eta) \in S \) and \( Y \subseteq S \), the procedure ConstructControlsS(\( k, (z, \eta), Y, T \)) returns a set of \( k \) controls in \( U \) such that the \( U \)-component of these controls are computed as in ConstructControlsS, and the martingale-control-components of these controls are sampled in admissible sets.

The extended iMDP algorithm is presented in Algorithms [15]. The algorithm incrementally refines two MDP sequences, namely \( \{M_n\}_{n=0}^\infty \) and \( \{M^\infty_n\}_{n=0}^\infty \), and two holding time sequences, namely \( \{\Delta_n\}_{n=0}^\infty \) and \( \{\Delta^\infty_n\}_{n=0}^\infty \), that consistently approximate the original system in Eq. 1 and the augmented system in Eq. 5 respectively. We associate with \( z \in S \) a cost value \( J_n(z, 1) \), a control \( \mu_n(z, 1) \), a min-failure probability value \( \gamma_n(z) \), a cost-to-go \( J^\infty_n(z) \) induced by the obtained min-failure policy. Similarly, we associate with \( z \in S \) a cost value \( J_n(z, \tau) \), a control \( \mu_n(z, \tau) \).

As shown in Algorithm [1] initially, empty MDP models \( M_0 \) and \( \overline{M}_0 \) are created. The algorithm then executes \( N \) iterations in which it samples states on the pre-specified part of the boundary \( \partial D \), constructs the uns-specified part of \( \partial D \) and processes the interior of \( D \). More specifically, at Line 3 the procedure UpdateDataStorage(\( n - 1, n \)) indicates that refined models in the \( n \)th iteration are constructed from models in the \((n - 1)\)th iteration, which can be implemented by simply sharing memory among iterations. Using rejection sampling, the procedure SampleOnBoundary at Line 4 sample states in \( \partial S \) and \( \partial S \times [0, 1] \) to add to \( S_n \) and \( S_n \) respectively. We also initialize appropriate cost values for these sampled states. The procedure ConstructBoundary (Line 5) refines the MDP sequence \( \{M^\infty_n\}_{n=0}^\infty \) as done in the original iMDP algorithm. Thus, we can compute cost value \( J_n \) on \( S_n \times [1] \). At the same time, we compute min-failure probabilities \( \gamma_n \) as well as min-failure cost value \( J^\infty_n \) on \( S_n \). In other words, the algorithm effectively constructs the approximate boundary \( \partial D \) of \( D \) and approximate cost-to-go \( J_n \) on this boundary over iterations. To compute cost value for the interior \( D^n \) of \( D \), the procedure ProcessInterior...
ConstructControlsS

\[ \text{where the control set } J_n \text{ is constructed using the procedure ConstructControlsSM.} \]
\[ \text{in the augmented state space. In the following discussion, we first present the details of the implementation of the two procedures.} \]
\[ \text{In Algorithm 2, we discuss the implementation of the procedure ConstructBoundary. We construct a finer MDP model } \mathcal{M}_n \text{ based on the previous model as follows. A state } z_n \text{ is sampled from the interior of the state space } S. \]
\[ \text{The nearest state } z_{\text{near}} \text{ to } z_n \text{ is used to construct an extended state } z_{\text{near}} \text{ by using the procedure ExtendBackwardsS at Line 3. The extended states } z_{\text{near}} \text{ and } (z_n, 1) \text{ are added into } S_n \text{ and } S_n \text{ respectively.} \]
\[ \text{The associated cost value } J_n(z_n, 1), \text{ min-failure probability } \gamma_n(z_n), \text{ min-failure cost value } J_n^0(z_n) \text{ and control } \mu_n(z_n) \text{ are initialized at Line 7.} \]
\[ \text{We then perform } L_n \geq 1 \text{ updating rounds in each iteration (Lines 8-11). In particular, we construct the update-set } Z_{\text{update}} \text{ consisting of } K_n = \Theta(|S_n|^6) \text{ states and } z_{\text{near}} \text{ where } |K_n| < |S_n|. \]
\[ \text{For each state } z \in Z_{\text{update}}, \text{ the procedure UpdateSM as shown in Algorithm 4 implements the following Bellman update:} \]
\[ J_n(z, 1) = \min_{v \in U_n(z)} \{G_n(z, v) + \alpha \Delta_{z_n}(z) \mathbb{E}_{\mathcal{P}_n}|J_n-1(y)|z, v, c)\}, \]
\[ \text{The details of the implementation are as follows. A set of } U_n \text{ controls is constructed using the procedure ConstructControlsS where } |U_n| = \Theta(\log(|S_n|)) \text{ at Line 2. For each } v \in U_n, \text{ we construct the support } Z_{\text{near}} \text{ and compute the transition probability } P_n(\cdot|z, v) \text{ consistently over } Z_{\text{near}} \text{ from the procedure ComputeTranProb (Line 3).} \]
\[ \text{The cost values for the state } z \text{ and controls in } U_n \text{ are computed at Lines 5. We finally choose the best control in } U_n \text{ that yields the smallest updated cost value (Line 7). Correspondingly, we improve the min-failure probability } \gamma_n \text{ and its induced min-failure cost value } J_n^0 \text{ in Lines 8-11.} \]
\[ \text{Similarly, in Algorithm 5, we carry out the sampling and extending process in the augmented state space } S \text{ to refine the MDP sequence } \mathcal{M}_n \text{ (Lines 2). We then execute } L_n \text{ rounds (Lines 7-10)} \text{ to update the cost-to-go } J_n \text{ for states in the interior } D_n \text{ of } D \text{ using the procedure UpdateSM as shown in Algorithm 4.} \]
\[ \text{When a state } z \in S_n \text{ is updated in UpdateSM, we perform the following Bellman update:} \]
\[ J_n(z) = \min_{(v, c) \in U_n(z)} \{G_n(z, v) + \alpha \Delta_{z_n}(z) \mathbb{E}_{\mathcal{P}_n}|J_n-1(y)|z, v, c)\}, \]
\[ \text{where the control set } U_n \text{ is constructed by the procedure ConstructControlsSM, and the transition probability } \mathcal{P}_n(\cdot|z, v, c) \text{ consistently approximates the augmented dynamics in Eq. 5.} \]

C. Feedback Control

At the th iteration, given a state x ∈ S and a martingale component q, to find a policy control (v, c), we perform a Bellman update based on the approximated cost-to-go Jn for the augmented state (x, q). During the holding time period, the original system takes the control v and evolves in the original state space S while we simulate the dynamics of the martingale component under the martingale control c. After this holding time period, the augmented system has a new state (x′, q′), and we repeat the above process.

D. Complexity

The time complexity per iteration of the implementation in Algorithms 3 and 4 is \(O(|S_n|^6 \log(|S_n|)^2)\). The space complexity of the iMDP algorithm is \(O(|S_n|)\) where \(|S_n| = \Theta(n)\) due to our sampling strategy.

V. Analysis

In this section, we present main results on the performance of the extended iMDP algorithm with brief explanation. More detailed proofs can be found in [16]. We first review the following key results of the approximating Markov chain method when no additional risk
constraints are considered [17]. Local consistency implies 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. In particular, the sequence of optimal cost-to-go $J^*_n(\cdot,1)$ of discrete MDPs converge uniformly to $J^*(\cdot,1)$ of the original problem.

Previous results in [15] show that $J_n(\cdot,1)$ returned from the iMDP algorithm converges uniformly to $J^*(\cdot,1)$ in probability. That is, we are able to compute $J^*(\cdot,1)$ in an incremental manner without directly computing $J_n^*(\cdot,1)$. Using the same proof, we conclude that $\gamma_n(\cdot)$ and $J_n^*(\cdot)$ converges uniformly to $\gamma(\cdot)$ and $J^*(\cdot,\gamma)$ in probability respectively. Therefore, we have incrementally constructed the boundary conditions on $\partial D$ of the equivalent stochastic target problem presented in Eqs. 7-8. These results are established based on the approximation of the dynamics in Eq. 1 using the MDP sequence $\{\mathcal{M}_n\}_{n=0}^\infty$. Similarly, the uniform convergence of $J_n(\cdot,\cdot)$ to $J^*(\cdot,\cdot)$ in probability on the interior of $D$ is followed from the approximation of the dynamics in Eq. 5 using the MDP sequence $\{\bar{\mathcal{M}}_n\}_{n=0}^\infty$. In the following theorem, we formally summarize the key convergence results of the extended iMDP algorithm.

**Theorem 3** Let $\mathcal{M}_n$ and $\bar{\mathcal{M}}_n$ be two MDPs with discrete states constructed in $S$ and $\bar{S}$ respectively, and let $J_n : S_n \rightarrow R$ be the cost-to-function returned by the extended iMDP algorithm at the $n^{th}$ iteration. Let us define $||b||_X = \sup_{z \in X} b(z)$ as the sup-norm over a set $X$ of a function $b$ with a domain containing $X$. We have the following events happens in probability:

1. $\lim_{n \rightarrow \infty} ||J_n(\cdot,1) - J^*(\cdot,1)||_{S_n} = 0$,
2. $\lim_{n \rightarrow \infty} ||\gamma_n - \gamma||_{S_n} = 0$,
3. $\lim_{n \rightarrow \infty} ||J_n^* - J^*||_{S_n} = 0$,
4. $\lim_{n \rightarrow \infty} ||J_n - J^*||_{S_n} = 0$.

The first three events construct the boundary conditions on $\partial D$ in probability, which leads to the probabilistically sound property of the extended iMDP algorithm. The last event asserts the asymptotically optimal property through the convergence of the approximating cost-to-go $J_n$ to the optimal cost-to-go $J^*$ on the augmented state space $\bar{S}$.

**VI. Experiments**

In the following experiments, we used a computer with a 2.0-GHz Intel Core 2 Duo T6400 processor and 4 GB of RAM. We controlled a system with stochastic single integrator dynamics to a goal region with free ending time in a cluttered environment. The standard deviation of noise in each direction is 0.5. The system stops when it collides with obstacles or reach the goal region. The cost function is the weighted sum of total energy spent to reach the goal $X_{goal}$, which is measured as the integral of square of control magnitude, and terminal cost, which is $-1000$ for the goal region $G$ and 10 for the obstacle region $\Gamma$, with discount factor $\alpha = 0.9$. The maximum velocity of the system is one. At the beginning, the system starts from $(6.5, -3)$. The system can go through narrow corridors or go around the obstacles to reach the goal region. In this setting, failure is defined as collisions with obstacles, and thus we use failure probability and collision probability interchangeably.

We first show how the extended iMDP algorithm constructs the sequence of approximating MDPs on $S$ and $\bar{S}$ over iterations in Fig. 1. In particular, Figs. 1(a)-1(c) depict anytime policies on the boundary $S \times 1.0$ after 500, 1000, and 3000 iterations. Figures 1(d)-1(f) show Markov chains created by anytime policies found by the algorithm on $\mathcal{M}_n$ after 200, 500 and 1000 iterations. We observe that the structures of these Markov chains are indeed random graphs that are (almost-surely) connected to cover the state space $S$. It is worth noting that the structures of these Markov chains can be constructed on-demand during the execution of the algorithm. Similarly, Figures 1(g)-1(i) show the corresponding anytime policies in the augmented state space $\bar{S}$ over iterations. In Fig. 1(i) we show the top-down view of a policy for states in $\mathcal{M}_{3000} \setminus \mathcal{M}_{3000}$. Compared to Fig 1(c) we observe that the system will try to avoid narrow corridors when the risk tolerance is low. In Figs. 1(j)-1(l) we show Markov chains that are created by anytime policies in the augmented state space. As we can see again, the structures of these Markov chains quickly cover $\bar{S}$ with (almost-surely) connected random graphs.

We now examine how the algorithm constructs boundary conditions on $\partial D$ and process the interior $D^o$ of the reformulated stochastic target problem in Fig. 2. Figures 2(a)-2(c) show approximate cost-to-go $J_n$ when the probability threshold $\eta_n$ is 1.0 for $n = 200$, 2000 and 4000. As we recall, the value functions in these figures form the boundary conditions on $S \times 1$, which is a subset of $\partial D$. The logarithm of the min-collision probability $\gamma_{4000}$ on $S_{4000}$ is plotted in Fig. 2(d). The cost-to-go $J^*_{4000}$ attained by following a policy that achieves the minimum collision probability in Fig. 2(d) is shown in Fig. 2(f). The combination of the min-collision probability $\gamma_n$ and its induced cost-to-go $J_n^*$ forms the boundary conditions on the remaining part of $\partial D$. As a comparison, we show in Fig. 2(c) the collision probability induced by a policy that achieves the cost-to-go $J_{4000,1.0}$ in Fig. 2(c). It is clear from the plots that these probabilities are significantly higher than min-collision probabilities. In the interior $D^o$, Figs. 2(g)-2(i) present the approximate cost-to-go $J_{4000}$ for augmented states where their martingale components are 0.1, 0.5 and 0.9. As we can see, the lower the martingale state is, the higher the cost value is - which is consistent with the characteristics in Section III-C.

Lastly, we tested the performance of obtained anytime policies after 4000 iterations with different initial collision probability thresholds $\eta$. In Fig. 3(a)-3(c) we show 50 trajectories resulting from a policy induced by $J_{4000}$ where $\eta = 0.2, 0.6$, and 1.0 respectively. As we can see from these figures, the martingale-components vary along trajectories. With low initial thresholds, the system tends to go around obstacles to reach the goal region. In contrast, with high initial thresholds, the system tends to go through narrow corridors more often. When $\eta = 1.0$, the system takes risk to minimize travel time to reach the goal. In Fig. 3(d)
Fig. 1. A system with stochastic single integrator dynamics in a cluttered environment. The standard deviation of noise in $x$ and $y$ directions is 0.5. The cost function is the sum of total energy spent to reach the goal, which is measured as the integral of square of control magnitude, and terminal cost, which is $-1000$ for the goal region ($\mathcal{G}$) and 10 for the obstacle region ($\mathcal{C}$), with discount factor $\alpha = 0.9$. Figures 1(a)-1(c) depict anytime policies on the boundary $\mathcal{S} \times 0$ over iterations. Figures 1(d)-1(f) show Markov chain created by anytime policies on $\mathcal{M}_n$ over iterations. Similarly, Figures 1(g)-1(i) and Figures 1(j)-1(l) show the corresponding anytime policies and associated Markov chains on $\mathcal{M}_n$ respectively. In Fig. 1(i), we show the top-down view of a policy for states in $\mathcal{M}_{3000} \setminus \mathcal{M}_{3000}$. We observe that the system will try to avoid narrow corridors when the risk tolerance is low. We can also observe that the structures of induced Markov chains quickly cover the state spaces $\mathcal{S}$ and $\mathcal{S}$ with connected random graphs.
we show 2000 corresponding trajectories in the original state space $S$ with reported average costs and simulated collision probabilities. As we can see, the simulated collision probabilities are below the corresponding thresholds, which suggests that returned policies are feasible and consistent with the initial collision probability thresholds. Moreover, the lower the threshold is, the higher the average cost is as we expect. When $\eta = 0.2$, the average cost is $-49.26$ and when $\eta_0 = 1$, the average cost is $-125.72$. The observations in this experiment illustrate how the extended iMDP algorithm processes risk tolerance along trajectories in different executions to minimize expected costs using feasible and time-consistent anytime policies.

VII. CONCLUSIONS

We have introduced and analyzed the extension of the incremental Markov Decision Process (iMDP) algorithm for stochastic optimal control in the presence of bounded probabilities of failure for initial states. We present here a martingale approach to construct time-consistent control policies by diffusing the probability constraint into a martingale. As a result, we formulate and solve an equivalent stochastic target problem in an augmented state space. The extended iMDP algorithm is designed to find anytime solutions to the equivalent problem by sampling both in the original and augmented state spaces. The extended algorithm inherits the efficient computation from iMDP to compute the expected costs using asynchronous value iterations. In addition, the
algorithm guarantees the probabilistic soundness and asymptotic optimality of computed control policies as the number of iterations approaches infinity.

The future extension of the work is broad. We intend to incorporate logical rules expressed as temporal logic constraints to achieve high degree of autonomy for systems to operate safely in uncertain and highly dynamic environments with complex mission specifications. We also plan to implement the algorithm outlined in this paper on robotic platforms for practical demonstration.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation grant CNS-1016213 and the Army Research Office MURI grant W911NF-11-1-0046.

REFERENCES

[1] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. How, “Real-time motion planning with applications to autonomous urban driving,” IEEE Trans. on Control Systems Technologies, vol. 17, no. 5, pp. 1105–1118, 2009.

[2] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. Cambridge, MA: MIT Press, 2005.

[3] E. Todorov, “Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system,” Neural Computation, vol. 17, pp. 1084–1108, 2005.

[4] R. Alterovitz, T. Siméon, and K. Goldberg, “The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty,” in in Robotics: Science and Systems III (Proc. RSS 2007. MIT Press, 2008, pp. 246–253

[5] W. H. Fleming and J. L. Stein, “Stochastic optimal control, international finance and debt,” Journal of Banking and Finance, vol. 28, pp. 979–996, 2004.

[6] S. P. Sethi and G. L. Thompson, Optimal Control Theory: Applications to Management Science and Economics, 2nd ed. Springer, 2006.

[7] V. D. Blondel and J. N. Tsitsiklis, “A survey of computational complexity results in systems and control,” Automatica, vol. 36, no. 9, pp. 1249–1274, 2000.

[8] C. Chow and J. Tsitsiklis, “A survey of computational complexity results in systems and control,” Automatica, vol. 36, no. 9, pp. 1249–1274, 2000.

[9] M. Boulbrachene and B. Chentouf, “The finite element approximation of hamilton-jacobi-bellman equations: the noncoercive case,” Applied Mathematics and Computation, vol. 158, no. 2, pp. 585–592, 2004.

[10] J. Rust, “Using Randomization to Break the Curse of Dimensionality,” Econometrica, vol. 56, no. 3, May 1997.

[11] S. Wang, L. S. Jennings, and K. L. Teo, “Numerical solution of hamilton-jacobi-bellman equations by an upwind finite volume method,” J. of Global Optimization, vol. 27, pp. 177–192, November 2003.

[12] M. Boulbrachene and B. Chentouf, “The finite element approximation of hamilton-jacobi-bellman equations: the noncoercive case,” Applied Mathematics and Computation, vol. 158, no. 2, pp. 585–592, 2004.

[13] J. Rust, “Using Randomization to Break the Curse of Dimensionality,” Econometrica, vol. 56, no. 3, May 1997.

[14] ———, “A comparison of policy iteration methods for solving continuous-state, infinite-horizon markovian decision problems using random, quasi-random, and deterministic discretizations,” EconWPA,” Computational Economics, 1997.
