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 timeconsistent control policies. The martingale stands for the level of risk tolerance that is contingent on available information 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 critical applications such as self-driving cars and robotic surgery, regulatory authorities can impose a threshold of failure probability during operation of these systems. Thus, finding control policies that fully respect this type of constraint is important in practice.

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 (HJB) 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 finitestate 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 when using the iMDP algorithm to solve 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 has 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-0* 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. The Lagrangian approach [23]-[25] 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 computationally demanding for high dimensional

Huynh and Frazzoli are affiliated with or members of the Laboratory for Information and Decision Systems, Kogan is with the Sloan School of Management, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139. huyn0002@gmail.com, {lkogan2, frazzoli}@mit.edu

systems and unsuitable for online robotics applications.

In another approach (see, e.g., [26]–[30]), most previous works use discrete-time multi-stage formulations to model this problem. In these modified formulations, failure is defined as collision with convex obstacles which can be represented as a set of linear inequalities. Probabilities of safety for states at different time instants as well as for the entire path are pre-specified by users. The proposed algorithms to solve these formulations often involve two main steps. In the first step, these algorithms often use heuristic [26] or iterative [27] risk allocation procedures to identify the tightness of different constraints. In the second step, the formulations with identified active constraints can be solved using mixed integer-linear programming with possible assistance of particle sampling [20] and linear programming relaxation [21]. Computing risk allocation fully is computationally intensive. Thus, in more recent works [28]-[30], the authors make use of the Rapidly-Exploring Random Tree (RRT) and RRT* algorithms to build tree data structures that also store incremental approximate allocated risks at tree nodes. Based on the RRT* algorithm, the authors have proposed the Chance-Constrained-RRT* (CC-RRT*) algorithm that would provide asymptotically-optimal and probabilistically-feasible trajectories for linear Gaussian systems subject to process noise, localization error, and uncertain environmental constraints. In addition, the authors have also proposed a new objective function that allows users to trade-off between minimizing path duration and riskaverse behavior by adjusting the weights of these additive components in the objective function.

We note that the modified formulations in the above approach do not preserve well the intended guarantees of the original chance constraint formulation. In addition, the approach requires the direct representation of convex obstacles into the formulations. Therefore, solving the resulting mixed integer-linear programming in the presence of a large number of obstacles is computationally demanding. The proposed algorithms are also over-conservative due to loose union bounds when performing the risk allocation procedures. To counter these conservative bounds, CC-RRT* constructs more aggressive trajectories by adjusting the weights of the path duration and risk-averse components in the objective function. As a result, it is hard to automate the selection of trajectory patterns.

Moreover, specifying in advance probabilities of safety for states at different time instants and for the entire path can lead to policies that have irrational behaviors due inconsistent risk preference over time. This phenomenon is known as *time-inconsistency* of control policies. For example, when we execute a control policy returned by one of the proposed algorithms, due to noise, the system can be in an area surrounded by obstacles at some later time t, it would be safer if the controller takes into account this situation and increases the required probability of safety at time t to encourage careful maneuvers. Similarly, if the system enters an obstacle-free area, the controller can reduce the required probability of safety at time t to encourage more aggressive maneuvers. Therefore, to maintain time-consistency of control policies, the controller should adjust safety probabilities so that they are *contingent* on available information along the controlled trajectory.

In other related works [31]–[33], several authors have proposed new formulations in which the objective functions and constraints are evaluated using (different) single-period risk metrics. However, these formulations again lead to potential inconsistent behaviors as risk preferences change in an irrational manner between periods [34]. Recently, in [22], the authors used Markov dynamic time-consistent risk measures [35]–[37] to assess the risk of future cost stream in a consistent manner and established a dynamic programming equation for this modified formulation. The resulting dynamic programming equation has functionals over the state space as control variables. When the state space is continuous, the control space has infinite dimensionality, and therefore, solving the dynamic programming equation in this case is computationally challenging.

In mathematical finance, closely-related problems have been studied in the context of hedging with portfolio constraints where constraints on terminal states are enforced almost surely (a.s.), yielding so-called *stochastic target problems* [38]–[42]. Research in this field focuses on deriving HJB equations for this class of problems. Recent analytical tools such as weak dynamic programming [38] and geometric dynamic programming [43], [44] have been 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 failure probabilities 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 represents the level of risk tolerance that is contingent on available information over time. Thus, the martingale approach transforms a risk-constrained problem into a stochastic target problem. By sampling in the augmented state space and computing 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 continuoustime dynamics in a time-consistent manner. The approach enable us 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], [45]. 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^o and has a smooth boundary ∂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 $\{w(t); t \ge 0\}$ is a d_w dimensional Brownian motion on some probability space. We define $\{\mathcal{F}_t; t \ge 0\}$ as the augmented filtration generated by the Brownian motion $w(\cdot)$. Let a control process $\{u(t); t \ge 0\}$ 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 \mathcal{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:

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

where $f : S \times U \to \mathbb{R}^{d_x}$ and $F : S \times U \to \mathbb{R}^{d_x \times d_w}$ are bounded measurable and continuous functions as long as $x(t) \in S^o$. 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 [46].

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

$$T_u^z = \inf \{ t : x(0) = z, x(t) \notin S^o, \text{ and Eq.}(1) \}.$$

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 ∂S of S. The random variable T_u^z can take value ∞ if the trajectory $x(\cdot)$ never exits S^o .

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

$$J_u(z) = \mathbb{E}_0^z \left[\int_0^{T_u^z} \alpha^t \, g\big(x(t), u(t)\big) \, dt + \alpha^{T_u^z} h(x(T_u^z)) \right],$$
(2)

where \mathbb{E}_t^z 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 $\Gamma \subset \partial S$ 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:

$$P_0^z(x(T_u^z) \in \Gamma) \le \eta,$$

where P_t^z 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 η and initial state z at time 0. 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. To have time-consistent control processes, the risk tolerance along controlled trajectories must vary consistently with the initial choice of risk tolerance η based on available information over time.

Let $\overline{\mathbb{R}}$ be the extended real number set. The *optimal cost*to-go function $J^*: S \to \overline{\mathbb{R}}$ is defined as follows ¹²:

$$\mathcal{OPT}1: \quad J^*(z;\eta) = \inf_{u(\cdot)\in\mathcal{U}} J_u(z)$$
 (3)

s/t
$$P_0^z(x(T_u^z) \in \Gamma) \le \eta$$
 and Eq. (1). (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 ϵ -optimal policy if $|J_u(z) - J^*(z;\eta)| \le \epsilon$.

We call a sampling-based algorithm probabilisticallysound 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 now 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.

¹The semicolon in $J^*(z; \eta)$ signifies that η is a parameter.

²Compared to [45], 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.

A. Diffusing Risk Constraints

Lemma 1 (see [41], [42]) From x(0) = z, a control process $u(\cdot)$ is feasible for OPT1 if and only if there exists a square-integrable (but possibly unbounded) process $c(\cdot) \in \mathbb{R}^{d_w}$ 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) $1_{\Gamma}(x(T_u^z)) \le q(T_u^z)$ a.s.

where $1_{\Gamma}(x) = 1$ if and only if $x \in \Gamma$ and 0 otherwise. 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:

$$P_0^z(x(T_u^z) \in \Gamma) = \mathbb{E}\left[1_{\Gamma}(x(T_u^z))|\mathcal{F}_0\right]$$

$$\leq \mathbb{E}\left[q(T_u^z)|\mathcal{F}_0\right] = q(0) = \eta.$$

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

Now, let $u(\cdot)$ be a feasible control policy. Set $\eta_0 = P_0^z(x(T_u^z) \in \Gamma)$. We note that $\eta_0 \leq \eta$. We define the martingale

$$\overline{q}(t) = \mathbb{E}[1_{\Gamma}(x(T_u^z))|\mathcal{F}_t].$$

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

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

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

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

$$q(t) = \widehat{q}(t)\mathbf{1}_{t \le \tau} + \mathbf{1}_{t > \tau},$$

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

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

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

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

$$q(T_u^z) = \mathbb{E}[1_{\Gamma}(x(T_u^z)) | \mathcal{F}_{T_u^z}] + (\eta - \eta_0) = 1_{\Gamma}(x(T_u^z)) + (\eta - \eta_0) \ge 1_{\Gamma}(x(T_u^z)).$$

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

The control process $c(\cdot)$ exists due to the martingale representation theorem [47], which yields $dq(t) = c^T(t)dw(t)$. We however note that c(t) is possibly unbounded. We also emphasize that the risk tolerance η becomes the initial value of the martingale $q(\cdot)$.

B. Stochastic Target Problem

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

$$d\begin{bmatrix} x(t)\\ q(t)\end{bmatrix} = \begin{bmatrix} f(x(t), u(t))\\ 0\end{bmatrix} dt + \begin{bmatrix} F(x(t), u(t))\\ c^{T}(t)\end{bmatrix} dw(t),$$
(5)

where $(u(\cdot), c(\cdot))$ 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 $S \times [0, 1]$ as \overline{S} and the augmented control space $U \times \mathbb{R}^{d_w}$ as \overline{U} . We also refer to the nominal dynamics and diffusion matrix of Eq. (5) as $\overline{f}(x, q, u, c)$ and $\overline{F}(x, q, u, c)$ respectively.

It is well-known that in the following reformulated problem, optimal control processes are Markov controls [41], [42], [48]. 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 $t \ge 0$. A function $\varphi: \overline{S} \to \overline{U}$ represents a *Markov or feedback control policy*, which is known to be admissible with respect to the process noise $w(\cdot)$. Let Ψ be the set of all such policies φ . Let $\mu: \overline{S} \to U$ and $\kappa: \overline{S} \to \mathbb{R}^{d_w}$ so that $\varphi = (\mu, \kappa)$. We rename T_u^z to T_{φ}^z for the sake of notation clarity. Using these notations, $\mu(\cdot, 1)$ is thus a Markov control policy for the unconstrained problem, i.e. the problem without the risk constraint, 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 II be the set of all such Markov control policies $\mu(\cdot)$ on S.

Now, let us rewrite cost-to-go function $J_u(z)$ in Eq. (2) for the threshold η at time 0 in a new form:

$$J_{\varphi}(z,\eta) = \mathbb{E}\left[\int_{0}^{T_{\varphi}^{z}} \alpha^{t} g\left(x(t), \mu(x(t), q(t))\right) dt + \alpha^{T_{\varphi}^{z}} h(x(T_{\varphi}^{z})) \Big| (x,q)(0) = (z,\eta)\right].$$
 (6)

We therefore transform the risk-constrained problem OPT1 into a stochastic target problem as follows³:

$$\mathcal{OPT}2: \quad J^*(z,\eta) = \inf_{\varphi \in \Psi} J_{\varphi}(z,\eta) \tag{7}$$

s/t
$$1_{\Gamma}(x(T_{\varphi}^z)) \le q(T_{\varphi}^z)$$
 a.s. and Eq. (5). (8)

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 [41], [42]. In this formulation, we solve for feedback control policies φ for all $(z, \eta) \in \overline{S}$ instead of a particular choice of η 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. (8) by constructing its boundary and computing the boundary conditions.

C. Characterization and Boundary Conditions

The domain of the stochastic target problem in OPT_2 is:

$$D = \{(z,\eta) \in \overline{S} \mid \exists \varphi \in \Psi \text{ s/t } 1_{\Gamma}(x(T_{\varphi}^z)) \leq q(T_{\varphi}^z) \text{ a.s.} \}$$

By the definition of the risk-constrained problem $\mathcal{OPT}1$, 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

$$\gamma(z) = \inf \{ \eta \in [0, 1] \mid (z, \eta) \in D \},$$
(9)

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

$$\gamma(z) = \inf_{u \in \mathcal{U}} P_0^z \left(x(T_u^z) \in \Gamma \right) = \inf_{u \in \mathcal{U}} \mathbb{E}_0^z \left[\mathbb{1}_{\Gamma} (x(T_u^z)) \right].$$
(10)

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

Thus, the boundary of D is

$$\partial D = S \times \{1\} \cup \{(z, \gamma(z)) \mid z \in S\} \\ \cup \{(z, \eta) \mid z \in \partial S, \eta \in [\gamma(z), 1]\}.$$
(11)

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

The domain D is illustrated in Fig. 1(a). In this example, the state space S is a bounded two-dimensional area with boundary ∂S containing a goal region G and an obstacle region $\Gamma = Obs$. The augmented state space \overline{S} augments S with an extra dimension for the martingale state q. The infimum probability of reaching into Γ from states in S is depicted as γ . As we can see, γ takes value 1 in Γ . The volume between γ and the hyper-plane q = 1 is the domain D of OPT2.

Now, let $\eta = 1$, we notice that $J^*(z, 1)$ is the optimal costto-go from z for the stochastic optimal problem without the risk constraint:

$$J^*(z,1) = \inf_{u \in \mathcal{U}} J_u(z).$$

An optimal control process that solves this optimization problem is given by a Markov policy $\mu^*(\cdot, 1) \in \Pi$. We now define the failure probability function $\Upsilon : S \to [0, 1]$ under such an optimal policy $\mu^*(\cdot, 1)$ as follows:

$$\Upsilon(z) = \mathbb{E}\big[\mathbf{1}_{\Gamma}(x(T_{\mu^*}^z))\big], \ \forall z \in S,$$
(12)

where $T_{\mu^*}^z$ is the first exit time when the system follows the control policy $\mu^*(\cdot, 1)$ from the initial state z. By the definitions of γ and Υ , we can recognize that $\Upsilon(z) \ge \gamma(z)$ for all $z \in S$. Figure 1(b) shows an illustration of Υ for the same example in Fig. 1(a).

Since following the policy $\mu^*(\cdot, 1)$ from an initial state z yields a failure probability $\Upsilon(z)$, we infer that:

$$J^{*}(z,1) = J^{*}(z,\Upsilon(z)).$$
(13)

From the definition of the problem OPT_1 , we also have:

$$0 \le \eta < \eta' \le 1 \Rightarrow J^*(z,\eta) \ge J^*(z,\eta').$$
(14)

Thus, for any $\Upsilon(z) < \eta < 1$, we have:

$$J^*(z,1) \le J^*(z,\eta) \le J^*(z,\Upsilon(z)).$$
 (15)

Combining Eq. (13) and Eq. (15), we have:

$$\forall \eta \in [\Upsilon(z), 1] \Rightarrow J^*(z, \eta) = J^*(z, 1). \tag{16}$$

As a consequence, when we start from an initial state z with a risk threshold η that is at least $\Upsilon(z)$, it is optimal to execute an optimal control policy of the corresponding unconstrained problem from the initial state z.

It also follows from Eq. (14) that reducing the risk tolerance from 1.0 along the controlled process can not reduce the optimal cost-to-go function evaluated at (x(t), q(t) = 1.0). Thus, we infer that for augmented states (x(t), q(t)) where q(t) = 1.0, the optimal martingale control $c^*(t)$ is 0.

Now, under all admissible policies φ , we can not obtain a failure probability for an initial state z that are lower than $\gamma(z)$. Thus, it is clear that $J^*(z,\eta) = +\infty$ for all $0 \leq \infty$ $\eta < \gamma(z)$. 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. (3)-(4), we assume that $\gamma(x)$ is a smooth function⁴. When $q(t) = \gamma(x(t))$ and u(t) is chosen, we must have:

$$c(t)^{T} = \frac{\partial \gamma}{\partial x(t)}^{T} F(x(t), u(t)).$$
(17)

Proof: Using the geometric dynamic programming principle [43], [44], we have the following result: for all stopping time $\tau \ge t$, when $q(t) = \gamma(x(t))$, a feasible control policy $\varphi \in \Psi$ satisfies $q(\tau) \ge \gamma(x(\tau))$ almost surely.

Take $\tau = t+$, under a feasible control policy φ , we have $q(t+) \ge \gamma(x(t+))$ a.s. for all t, and hence $dq(t) \ge d\gamma(x(t))$ a.s. By Itô lemma, we derive the following relationship:

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

For the above inequality to hold almost surely, the coefficient of dw(t) must be 0. This leads to Eq. (17).

In addition, if a control process that solves Eq. (10) is obtainable, say u_{γ} , the cost-to-go due to that control process is $J_{u_{\gamma}}(z)$. We will conveniently refer to $J_{u_{\gamma}}(z)$ as $J^{\gamma}(z)$. Under the mild assumption that u_{γ} is unique, it follows that $J^{\gamma}(z) = J^*(z, \gamma(z))$.

We also emphasize that when (x(t), q(t)) is inside the interior D^o of D, the usual dynamic programming principle holds. The extension of iMDP outlined below is designed to compute the sequence of approximate cost-to-go values on the boundary ∂D and in the interior D^o .

IV. Algorithm

In this section, we briefly overview how the Markov chain approximation technique is used in both the original and augmented state spaces. We then present the extended iMDP algorithm that incrementally constructs the boundary values and computes solutions to our problem. In particular, we sample in the original state space S to compute $J^*(\cdot, 1)$ and its induced collision probability $\Upsilon(\cdot)$ as in Eq. (12), the minfailure probability $\gamma(\cdot)$ as in Eq. (10) and its induced costto-go $J^{\gamma}(\cdot)$. Concurrently, we also sample in the augmented state space \overline{S} with appropriate values for samples on the boundary of D and approximate the optimal cost-to-go function $J^*(\cdot, \cdot)$ in the interior D^o . As a result, we construct a sequence of anytime control policies to approximate an optimal control policy $\varphi^* = (\mu^*, \kappa^*)$ in an efficient iterative procedure.

⁴When $\gamma(x)$ is not smooth, we need the concept of viscosity solutions and weak dynamic programming principle. See [41], [42] for details.



(a) A domain of OPT1.

(b) Failure probabilities due to optimal policies of the unconstrained problem.

Fig. 1. In Fig. 1(a), we show an example of the domain of $\mathcal{OPT2}$. The state space S is a bounded two-dimensional area with boundary ∂S containing a goal region G and an obstacle region $\Gamma = Obs$. The augmented state space S augments S with an extra dimension for the martingale state q. The infimum probability of reaching into Γ from states in S is depicted as γ , which takes value 1 in Γ . The volume between γ and the hyper-plane q = 1 is the domain D of $\mathcal{OPT2}$. In Fig. 1(b), we show an illustration of the failure probability function Υ due to an optimal control policy $\mu^*(\cdot, 1)$ of the unconstrained problem. We plot Υ for the same two-dimensional example. By the definitions of γ and Υ , we have $\Upsilon \geq \gamma$.

A. 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, $P(\cdot | \cdot, \cdot) : X \times X \times A \to \mathbb{R}_{\geq 0}$ is the transition probability function, $G(\cdot, \cdot) : X \times A \to \mathbb{R}$ is an immediate cost function, and $H : X \to \mathbb{R}$ is a terminal cost function. From an initial state ξ_0 , under a sequence of controls $\{v_i; i \in \mathbb{N}\}$, the induced trajectory $\{\xi_i; i \in \mathbb{N}\}$ is generated by following the transition probability function P.

On the state space S, we want to approximate $J^*(z, 1)$, $\Upsilon(z)$, $\gamma(z)$ and $J^{\gamma}(z)$ for any state $z \in S$, and it is suffice to consider optimal Markov controls as shown in [15], [16]. The Markov chain approximation method approximates the continuous dynamics in Eq. (1) using a sequence of MDPs $\{\mathcal{M}_n = (S_n, U, P_n, G_n, H_n)\}_{n=0}^{\infty}$ and a sequence of holding times $\{\Delta t_n\}_{n=0}^{\infty}$ that are locally consistent. In particular, we construct $G_n(z, v) = g(z, v)\Delta t_n(z)$, $H_n(z) =$ h(z) for each $z \in S_n$ and $v \in U$. We also require that $\lim_{n\to\infty} \sup_{i\in\mathbb{N}, \omega\in\Omega_n} ||\Delta\xi_i^n||_2 = 0$ where Ω_n is the sample space of \mathcal{M}_n , $\Delta\xi_i^n = \xi_{i+1}^n - \xi_i^n$, and

- For all $z \in S$, $\lim_{n \to \infty} \Delta t_n(z) = 0$,
- For all $z \in S$ and all $v \in U$:

$$\lim_{n \to \infty} \frac{\mathbb{E}_{P_n}[\Delta \xi_i^n \mid \xi_i^n = z, u_i^n = v]}{\Delta t_n(z)} = f(z, v),$$
$$\lim_{n \to \infty} \frac{\operatorname{Cov}_{P_n}[\Delta \xi_i^n \mid \xi_i^n = z, u_i^n = v]}{\Delta t_n(z)} = F(z, v)F(z, v)^T.$$

The main idea of the Markov chain approximation approach for solving the original continuous problem is to solve a sequence of control problems defined on $\{\mathcal{M}_n\}_{n=0}^{\infty}$ as follows. A Markov or feedback policy μ_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 Π_n . We define $t_i^n = \sum_{0}^{i-1} \Delta t_n(\xi_i^n)$ for $i \geq 1$ and $t_0^n = 0$. Given a policy μ_n that approximates a Markov control process $u(\cdot)$ in Eq. (2), the corresponding cost-to-go due to μ_n on \mathcal{M}_n is:

$$J_{n,\mu_n}(z) = \mathbb{E}_{P_n}^{z} \left[\sum_{i=0}^{I_n-1} \alpha^{t_i^n} G_n(\xi_i^n, \mu_n(\xi_i^n)) + \alpha^{t_{I_n}^n} H_n(\xi_{I_n}^n) \right].$$

where $\mathbb{E}_{P_n}^z$ denotes the conditional expectation given $\xi_0^n = z$ under P_n , and $\{\xi_i^n; i \in \mathbb{N}\}$ is the sequence of states of the controlled Markov chain under the policy μ_n , and I_n is termination time defined as $I_n = \min\{i : \xi_i^n \in \partial S_n\}$ where $\partial S_n = \partial S \cap S_n$.

The optimal cost-to-go function $J_n^* : S \to \overline{\mathbb{R}}$ that approximates $J^*(z, 1)$ is denoted as

$$J_n^*(z,1) = \inf_{\mu_n \in \Pi_n} J_{n,\mu_n}(z) \ \forall z \in S_n.$$
(18)

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

We also define the failure probability function $\Upsilon_n : S_n \to [0,1]$ due to an optimal policy μ_n^* as follows:

$$\Upsilon_n(z) = \mathbb{E}_{P_n} \left[\mathbb{1}_{\Gamma}(\xi_{I_n}^n) \mid x(0) = z \; ; \; \mu_n^* \right] \; \forall z \in S_n, \quad (19)$$

where we denote μ_n^* after the semicolon (as a parameter) to emphasize the dependence of the Markov chain on this control policy.

In addition, the *min-failure probability* γ_n on \mathcal{M}_n that approximates $\gamma(z)$ is defined as:

$$\gamma_n(z) = \inf_{\mu_n \in \Pi_n} \mathbb{E}_{P_n}^z \left[\mathbb{1}_{\Gamma}(\xi_{I_n}^n) \right] \ \forall z \in S_n.$$
(20)

We note that the optimization programs in Eq. (18) and Eq. (20) may have two different optimal feedback control policies. Let $\nu_n \in \Pi_n$ be a control policy on \mathcal{M}_n that achieves γ_n , then the cost-to-go function due to ν_n is J_{n,ν_n} which approximates J^{γ} . For this reason, we conveniently refer to J_{n,ν_n} as J_n^{γ} .

Similarly, in the augmented state space \overline{S} , we use a sequence of MDPs $\{\overline{\mathcal{M}}_n = (\overline{S}_n, \overline{U}, \overline{P}_n, \overline{G}_n, \overline{H}_n)\}_{n=0}^{\infty}$ and a sequence of holding times $\{\overline{\Delta t}_n\}_{n=0}^{\infty}$ that are locally consistent with the augmented dynamics in Eq. (5). In particular, \overline{S}_n is a random subset of $D \subset \overline{S}$, \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 Δt_n , we also construct the transition probabilities \overline{P}_n on $\overline{\mathcal{M}}_n$ and holding time $\overline{\Delta t}_n$ that satisfy the local consistency conditions for nominal dynamics $\overline{f}(x, q, u, c)$ and diffusion matrix $\overline{F}(x, q, u, c)$.

A trajectory on $\overline{\mathcal{M}}_n$ is denoted as $\{\overline{\xi}_i^n; i \in \mathbb{N}\}$ where $\overline{\xi}_i^n \in \overline{S}_n$. A Markov policy φ_n is a function that maps each state $(z,\eta) \in \overline{S}_n$ to a control $(\mu_n(z,\eta), \kappa_n(z,\eta)) \in \overline{U}$. Moreover, admissible κ_n at $(z,1) \in \overline{S}_n$ is 0 and at $(z,\gamma_n(z)) \in \overline{S}_n$ is a function of $\mu(z,\gamma_n(z))$ as shown in Eq. (17). Admissible κ_n for other states in \overline{S}_n is such that the martingale-component process of $\{\overline{\xi}_i^n; i \in \mathbb{N}\}$ belongs to [0,1] almost surely. We can show that equivalently, each control component of $\kappa_n(z,\eta)$ belongs to $[-\frac{\min(\eta,1-\eta)}{\Delta t_n d_w}, \frac{\min(\eta,1-\eta)}{\Delta t_n d_w}]$. The set of all such policies φ_n is Ψ_n .

Under a control policy φ_n , the cost-to-go on $\overline{\mathcal{M}}_n$ that approximates Eq. (6) is defined as:

$$J_{n,\varphi_n}(z,\eta) = \mathbb{E}_{\overline{P}_n}^{z,\eta} \left[\sum_{i=0}^{\overline{I}_n-1} \alpha^{\overline{t}_i^n} \overline{G}_n(\overline{\xi}_i^n, \mu_n(\overline{\xi}_i^n)) + \alpha^{\overline{t}_{\overline{I}_n}^n} \overline{H}_n(\overline{\xi}_{\overline{I}_n}^n) \right],$$

where $\overline{t}_i^n = \sum_0^{i-1} \overline{\Delta t}_n(\overline{\xi}_i^n)$ for $i \ge 1$ with $\overline{t}_0^n = 0$, and \overline{I}_n is index when the *x*-component of $\underline{\overline{\xi}}_i^n$ first arrives at ∂S . The approximating optimal cost $J_n^* : \overline{S}_n \to \overline{\mathbb{R}}$ for J^* in Eq. (7) is:

$$J_n^*(z,\eta) = \inf_{\varphi_n \in \Psi_n} J_{n,\varphi_n}(z,\eta) \ \forall (z,\eta) \in \overline{S}_n.$$
(21)

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

The extension of iMDP outlined below is designed to compute the sequence of optimal cost-to-go functions $\{J_n^*\}_{n=0}^{\infty}$, associated failure probability functions $\{\Upsilon_n\}_{n=0}^{\infty}$, min-failure probability functions $\{\gamma_n\}_{n=0}^{\infty}$, min-failure cost functions $\{J_n^n\}_{n=0}^{\infty}$, and the sequence of anytime control policies $\{\mu_n\}_{n=0}^{\infty}$ and $\{\kappa_n\}_{n=0}^{\infty}$ in an incremental 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 $\zeta \in X \subset \mathbb{R}^{d_X}$ and a set $Y \subseteq X$, for any $k \in \mathbb{N}$, the procedure $\operatorname{Nearest}(\zeta, Y, k)$ returns the k nearest states $\zeta' \in Y$ that are closest to ζ in terms of the d_X -dimensional Euclidean norm.

3) Time Intervals: Given a state $\zeta \in X$ and a number $k \in \mathbb{N}$, the procedure ComputeHoldingTime (ζ, k, d) holding time computed returns a as follows: $\theta \varsigma \rho/d$ $\texttt{ComputeHoldingTime}(\zeta, k, d) = \chi_t \left(\frac{\log k}{k} \right)$, where $\chi_t > 0$ is a constant, and ς, θ 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 $q(\cdot, \cdot)$ as in Section II.

4) Transition Probabilities: We are given a state $\zeta \in X$, a subset $Y \in X$, a control v in some control set V, a positive number τ 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 $Z_{\text{near}} \subset X$ of states such that the state $\zeta + k(\zeta, v)\tau$ belongs to the convex hull of Z_{near} and $||z' - z||_2 = O(\tau)$ for all $\zeta' \neq \zeta \in Z_{\text{near}}$, and (ii) a function P that maps Z_{near} to a non-negative real numbers such that $P(\cdot)$ is a probability distribution over the support Z_{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, τ) such that (i) $\dot{x}(t) = f(x(t), u(t))dt$ and $u(t) = v \in U$ for all $t \in [0, \tau]$, (ii) $\tau \leq T$, (iii) $x(t) \in S$ for all $t \in [0, \tau]$, (iv) $x(\tau) = z$, and (v) x(0) is close to z'. If no such trajectory exists, the procedure returns failure. We can solve for the triple (x, v, τ) 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 \overline{S} , the procedure ExtBackwardsSM $((z,\eta), (z',\eta'), T)$ returns (x,q,v,τ) in which (x,v,τ) is output of ExtBackwardsS(z,z',T) and q is sampled according to a Gaussian distribution $N(\eta', \sigma_q)$ where σ_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) $\dot{x}(t) = f(x(t), u(t))dt$ 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'.

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

Algorithm 1: Risk Constrained iMDP() $(S_0, \overline{S}_0, J_0, \gamma_0, \Upsilon_0, J_0^{\gamma}, \mu_0, \kappa_0, \Delta t_0, \overline{\Delta t}_0) \leftarrow \emptyset;$ for $n = 1 \rightarrow N$ do 1 2 UpdateDataStorage(n-1, n); 3 SampleOnBoundary(n); 4 // $K_{1,n} \geq 1$ rounds for boundary conditions for $i=1 \rightarrow K_{1,n} \; \mathbf{do}$ 5 ConstructBoundary $(S_n, \overline{S}_n, J_n, \gamma_n, \Upsilon_n, J_n^{\gamma}, \mu_n, \Delta t_n)$; 6 // $K_{2,n} \geq 0$ rounds for the interior region for $i = 1 \rightarrow K_{2,n}$ do 7 $\texttt{ProcessInterior}(S_n, \overline{S}_n, J_n, \gamma_n, \Upsilon_n, J_n^{\gamma}, \mu_n, \kappa_n, \overline{\Delta t}_n);$ 8

Algorithm 2: ConstructBoundary $(S_n, \overline{S}_n, J_n, \gamma_n, \Upsilon_n, J_n^{\gamma}, \mu_n, \Delta t_n)$ 1 $z_s \leftarrow \text{Sample}(S)$; 2 $z_{near} \leftarrow \text{Nearest}(z_s, S_n, 1);$ 3 if $(x_e, u_e, \tau) \leftarrow \texttt{ExtBackwardsS}(z_{near}, z_s, T_0)$ then $z_e \leftarrow x_e(0);$ $ic = \tau g(z_e, u_e) + \alpha^{\tau} J_n(z_{near}, 1);$ 5 $ic^{\gamma} = \tau g(z_e, u_e) + \alpha^{\tau} J_n^{\gamma}(z_{near});$ 6 $(S_n, \overline{S}_n) \leftarrow (S_n \cup \{z_e\}, \overline{S}_n \cup \{(z_e, 1)\});$ 7 $\begin{array}{l} (J_n(z_e,1),\gamma_n(z_e),\Upsilon_n(z_e),J_n^{\gamma}(z_e),\mu_n(z_e,1),\Delta t_n(z_e)) \leftarrow \\ (ic,\gamma_n(z_{near}),\Upsilon_n(z_{near}),ic^{\gamma},u_e,\tau) ; \end{array}$ // Perform $L_n \geq 1$ updates for $i = 1 \rightarrow L_n$ do / Choose $\mathcal{K}_n = \Theta(|S_n|^{\theta}) < |S_n|$ states $Z_{update} \leftarrow \text{Nearest}(z_e, S_n \setminus \partial S_n, \mathcal{K}_n) \cup \{z_e\};$ for $z \in Z_{update}$ do 10 11 UpdateS $(z, S_n, J_n, \gamma_n, \Upsilon_n, J_n^{\gamma}, \mu_n, \Delta t_n)$;

The extended iMDP algorithm is presented in Algorithms 1-5. The algorithm incrementally refines two MDP sequences, namely $\{\mathcal{M}_n\}_{n=0}^{\infty}$ and $\{\overline{\mathcal{M}}_n\}_{n=0}^{\infty}$, and two holding time sequences, namely $\{\Delta t_n\}_{n=0}^{\infty}$ and $\{\overline{\Delta t}_n\}_{n=0}^{\infty}$, that consistently approximate the original system in Eq. (1) and the augmented system in Eq. (5) respectively. We associate with $z \in S_n$ a cost value $J_n(z, 1)$, a control $\mu_n(z, 1)$, a failure probability $\Upsilon_n(z)$ due to $\mu_n(\cdot, 1)$, a min-failure probability $\gamma_n(z)$, a cost-to-go value $J_n^{\gamma}(z)$ induced by the obtained min-failure policy. Similarly, we associate with $\overline{z} \in \overline{S}_n$ a cost value $J_n(\overline{z})$, a control $(\mu_n(\overline{z}), \kappa_n(\overline{z}))$.

As shown in Algorithm 1, initially, empty MDP models \mathcal{M}_0 and $\overline{\mathcal{M}}_0$ are created. The algorithm then executes N iterations in which it samples states on the pre-specified part of the boundary ∂D , constructs the un-specified part of ∂D and processes the interior of D. More specifically, at Line 3, 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 ∂S and $\partial S \times [0,1]$ to add to S_n and \overline{S}_n respectively. We also initialize appropriate cost values for these sampled states.

We conduct $K_{1,n}$ rounds to refine the MDP sequence $\{\mathcal{M}_n\}_{n=0}^{\infty}$ as done in the original iMDP algorithm using the procedure ConstructBoundary (Line 6). Thus, we can compute the cost function J_n and the associated failure

$$\begin{array}{c|c} \textbf{Algorithm 3:} \operatorname{ProcessInterior}(S_n, \overline{S}_n, J_n, \gamma_n, \Upsilon_n, J_n^{\gamma}, \mu_n, \kappa_n, \overline{\Delta t_n}) \\ \hline \mathbf{i} \ \overline{z}_s = (z_s, q_s) \leftarrow \operatorname{Sample}(\overline{S}); \\ \hline \overline{z}_{near} = (z_{near}, q_{near}) \leftarrow \operatorname{Nearest}(\overline{z}_s, \overline{S}_n, 1); \\ \hline \mathbf{i} \ \mathbf{i} (x_e, q_e, u_e, \tau) \leftarrow \operatorname{ExtBackwardsSM}(\overline{z}_{near}, \overline{z}_s, T_0) \ \textbf{then} \\ \hline \overline{z}_e \leftarrow (x_e(0), q_e); \\ \hline \mathbf{s} \ \mathbf{i} \ \mathbf{f} q_e < \gamma_n(z_{near}) \ \textbf{then} \\ \hline // \ \mathcal{C} \ \textbf{takes a large value} \\ \hline \mathbf{6} \ (\overline{S}_n, J_n(\overline{z}_e), \mu_n(\overline{z}_e), \kappa_n(\overline{z}_e), \overline{\Delta t_n}(\overline{z}_e)) \leftarrow \\ (\overline{S}_n \cup \{\overline{z}_e\}, \mathcal{C}, u_e, 0, \tau); \\ \hline \mathbf{7} \ \textbf{else} \\ \hline \mathbf{8} \ \mathbf{s} \ \mathbf{i} \ \mathbf{c} = \tau g(z_e, u_e) + \alpha^{\tau} J_n(\overline{z}_{near}); \\ \hline \mathbf{9} \ (\overline{S}_n, J_n(\overline{z}_e), \mu_n(\overline{z}_e), \kappa_n(\overline{z}_e), \overline{\Delta t_n}(\overline{z}_e)) \leftarrow \\ (\overline{S}_n \cup \{\overline{z}_e\}, c, u_e, 0, \tau); \\ // \ \operatorname{Perform} \ \overline{L}_n \ge 1 \ \mathrm{updates} \\ \hline \mathbf{for} \ \mathbf{i} = 1 \rightarrow \overline{L}_n \ \mathbf{do} \\ \hline // \ \operatorname{Choose} \ \overline{K}_n = \Theta(|\overline{S}_n|^{\theta}) < |\overline{S}_n| \ \mathrm{states} \\ \overline{Z}_{update} \leftarrow \operatorname{Nearest}(\overline{z}_e, \overline{S}_n \setminus \partial \overline{S}_n, \overline{K}_n) \cup \{\overline{z}_e\}; \\ \hline \mathbf{for} \ \overline{z} = (z, q) \in \overline{Z}_{update} \ \mathbf{do} \\ \hline \mathbf{3} \ updateSM(\overline{z}, \overline{S}_n, J_n, \gamma_n, \gamma_n, J_n^{\gamma}, \mu_n, \kappa_n, \overline{\Delta t}_n); \\ \end{array}$$

probability function Υ_n on $S_n \times \{1\}$. In the same procedure, we compute the min-failure probability function γ_n as well as the min-failure cost function J_n^{γ} on S_n . In other words, the algorithm effectively constructs approximate boundaries for D and approximate cost-to-go functions J_n on these approximate boundaries over iterations. To compute cost values for the interior D^o of D, we conduct $K_{2,n}$ rounds of the procedure ProcessInterior (Line 8) that similarly refines the MDP sequence $\{\overline{\mathcal{M}}_n\}_{n=0}^{\infty}$ in the augmented state space. We can choose the values of $K_{1,n}$ and $K_{2,n}$ so that we perform a large number of iterations to obtain stable boundary values before processing the interior domain when n is small. In the following discussion, we will present in detail the implementations of these procedures.

In Algorithm 2, we show the implementation of the procedure ConstructBoundary. We construct a finer MDP model \mathcal{M}_n based on the previous model as follows. A state $z_{\rm s}$, is sampled from the interior of the state space S (Line 1). The nearest state $z_{\rm near}$ to $z_{\rm s}$ (Line 2) in the previous model is used to construct an extended state $z_{\rm e}$ by using the procedure ExtendBackwardsS at Line 3. The extended states $z_{\rm e}$ and $(z_{\rm e}, 1)$ are added into S_n and \overline{S}_n respectively. The associated cost value $J_n(z_{\rm e}, 1)$, failure probability $\Upsilon_n(z_{\rm e})$, min-failure probability $\gamma_n(z_{\rm e})$ and control $\mu_n(z_{\rm e})$ are initialized at Line 8.

We then perform $L_n \geq 1$ updating rounds in each iteration (Lines 9-12). In particular, we construct the updateset Z_{update} consisting of $K_n = \Theta(|S_n|^{\theta})$ states and z_e where $|K_n| < |S_n|$. For each state z in Z_{update} , the procedure UpdateS as shown in Algorithm 4 implements the following Bellman update:

$$J_n(z,1) = \min_{v \in U_n(z)} \{ G_n(z,v) + \alpha^{\Delta t_n(z)} \mathbb{E}_{P_n}[J_{n-1}(y)|z,v] \}.$$

The details of the implementation are as follows. A set of U_n controls is constructed using the procedure ConstructControlsS where $|U_n| = \Theta(\log(|S_n|))$ at Line 2. For each $v \in U_n$, we construct the support Z_{near} and



	Algorithm 5: UpdateSM $(\overline{z}, \overline{S}_n, J_n, \gamma_n, \Upsilon_n, J_n^{\gamma}, \mu_n, \kappa_n, \overline{\Delta t}_n)$			
1	$\overline{\tau} \leftarrow \texttt{ComputeHoldingTime}(\overline{z}, \overline{S}_n , d_x + 1);$			
	// Sample or discover $\overline{M}_n = \Theta(\log(\overline{S}_n))$ controls			
2	$\overline{U}_n \leftarrow \texttt{ConstructControlsSM}(\overline{M}_n, \overline{z}, \overline{S}_n, \overline{\tau});$			
3	for $\overline{v}=(v,c)\in\overline{U}_n$ do			
4	$(\overline{Z}_{near}, \overline{P}_n) \leftarrow \texttt{ComputeTranProb}(\overline{z}, \overline{v}, \overline{\tau}, \overline{S}_n, \overline{f}, \overline{F});$			
5	$J \leftarrow \overline{\tau}g(z, v) + \alpha^{\overline{\tau}} \sum_{\overline{y} = (y, s) \in \overline{Z}_{near}} \overline{P}_n(\overline{y}) \left[1_{s = \gamma_n(y)} J_n^{\gamma}(y) + \right]$			
	$1_{\gamma_n(y) < s < \Upsilon_n(y)} J_n(\overline{y}) + 1_{s \ge \Upsilon_n(y)} J_n(y, 1)];$			
	// Improved cost			
6	if $J < J_n(\overline{z})$ then			
7	$\left[(J_n(\overline{z}), \mu_n(\overline{z}), \kappa_n(\overline{z}), \overline{\Delta t}_n(\overline{z})) \leftarrow (J, v, c, \tau); \right]$			

compute the transition probability $P_n(\cdot | z, v)$ consistently over Z_{near} from the procedure ComputeTranProb (Line 4). The cost values for the state z and controls in U_n are computed at Lines 5. We finally choose the best control in U_n that yields the smallest updated cost value (Line 7). Correspondingly, we improve the min-failure probability γ_n and its induced min-failure cost value J_n^{γ} in Lines 9-12.

Similarly, in Algorithm 3, we carry out the sampling and extending process in the augmented state space \overline{S} to refine the MDP sequence $\overline{\mathcal{M}}_n$ (Lines 1-3). In this procedure, if an extended node has a martingale state that is below the corresponding min-failure probability, we initialize the cost value for extended node with a very large constant Crepresenting $+\infty$ (see Lines 5-6). Otherwise, we initialize the extended node as seen in Lines 8-9. We then execute \overline{L}_n rounds (Lines 10-13) to update the cost-to-go J_n for states in the interior D^o of D using the procedure UpdateSM as shown in Algorithm 5. When a state $\overline{z} \in \overline{S}_n$ is updated in UpdateSM, we perform the following Bellman update:

$$J_n(\overline{z}) = \min_{(v,c)\in\overline{U}_n(z)} \{\overline{G}_n(z,v) + \alpha^{\overline{\Delta t}_n(z)} \mathbb{E}_{\overline{P}_n}[J_{n-1}(\overline{y})|\overline{z},(v,c)]\},\$$

where the control set \overline{U}_n is constructed by the procedure ConstructControlsSM, and the transition probability $\overline{P}_n(\cdot|\overline{z}, (v, c))$ consistently approximates the augmented dynamics in Eq. (5). To implement the above Bellman update at Line 5 in Algorithm 5, we make use of the characteristics presented in Section III-C where the notation 1_A is 1 if the event A occurs and 0 otherwise. That is, when the martingale

$$\label{eq:second} \begin{array}{|c|c|c|c|} \hline \textbf{Algorithm 6:} & \operatorname{Risk Constrained Policy}(\overline{z}=(z,q)\in\overline{S},n) \\ \hline \textbf{i} & z_{\operatorname{nearest}} \leftarrow \operatorname{Nearest}(z,S_n,1); \\ \hline \textbf{2} & \textbf{if} & q \geq \gamma_n(z_{\operatorname{nearest}}) \, \textbf{then} \\ \hline & // & \operatorname{Switch to} & a & \operatorname{deterministic control policy} \\ \hline \textbf{3} & \textbf{return} & (\varphi(\overline{z})=(\mu_n(z_{\operatorname{nearest}}),0),\Delta t_n(z_{\operatorname{nearest}})); \\ \hline \textbf{4} & \textbf{else} \\ \hline & \textbf{5} & (J_{min},v_{min},c_{min}) \leftarrow (+\infty,\emptyset,\emptyset); \\ \hline \textbf{6} & \overline{\tau} \leftarrow \operatorname{ComputeHoldingTime}(\overline{z},|\overline{S}_n|,d_x+1); \\ \hline // & \operatorname{Construct} & \overline{M}_n = \Theta(\log(|\overline{S}_n|)) \quad \operatorname{controls} \\ \hline \textbf{7} & \overline{U}_n \leftarrow \operatorname{ConstructControlsSM}(\overline{M}_n,\overline{z},\overline{S}_n,\overline{\tau}); \\ \hline \textbf{8} & \textbf{for} & \overline{v} = (v,c) \in \overline{U}_n \, \, \textbf{do} \\ \hline \textbf{9} & (\overline{Z}_{\operatorname{near}},\overline{P}_n) \leftarrow \operatorname{ComputeTranProb}(\overline{z},\overline{v},\overline{\tau},\overline{S}_n,\overline{f},\overline{F}); \\ \hline \textbf{0} & (\overline{Z}_{\operatorname{near}},\overline{P}_n) \leftarrow \operatorname{ComputeTranProb}(\overline{z},v,\overline{v},\overline{S}_n,\overline{f},\overline{F}); \\ \hline \textbf{1} & (J_{\min}v_{min},c_{min}) \leftarrow (J,v,c); ; \\ \hline \textbf{1} & (J_{min},v_{min},c_{min}) \leftarrow (J,v,c); ; \\ \hline \textbf{1} & (J_{min},v_{min},c_{min}),\overline{\tau}); \\ \hline \end{array} \right.$$

state s of a state $\overline{y} = (y, s)$ in the support \overline{Z}_{near} is at least $\Upsilon_n(y)$, we substitute $J_n(\overline{y})$ with $J_n(y, 1)$. Similarly, when the martingale state s is equal to $\gamma_n(y)$, we substitute $J_n(\overline{y})$ with $J_n^{\gamma}(y)$.

C. Feedback Control

At the n^{th} iteration, given a state $x \in S$ and a martingale component q, to find a policy control (v, c), we perform a Bellman update based on the approximated cost-to-go J_n for the augmented state (x, q). During the holding time Δt_n , 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.

Figure 2(a) visualizes how feedback policies look in the original and augmented state spaces. In the augmented state space \overline{S} , a feedback control policy is a deterministic Markov policy as a function of an augmented state (x,q). As the system actually evolves in the original state space S, and the martingale state q can be seen as a random parameter at each state x, the feedback control policy is a randomized policy.

Using the characteristics presented in Section III-C, we infer that when a certain condition meets, the system can start following a deterministic control policy. More precisely, we recall that for all $\eta \in [\Upsilon(z), 1]$, we have $J^*(z, \eta) =$ $J^*(z, 1)$. Thus, starting from any augmented state (z, η) where $\eta > \Upsilon(z)$, we can solve the problem as if the failure probability were 1.0 and use optimal control policies of the unconstrained problem from the state z. We illustrate this idea in Fig. 2(b). As we can see, when the martingale state along the trajectory is at least the corresponding value provided by Υ , the system starts following a deterministic control policy $\mu_n(\cdot, 1)$ of the unconstrained problem.

Algorithm 6 implements the above feedback policy. As shown in this algorithm, Line 3 returns a deterministic policy



Fig. 2. In Fig. 2(a), we show a feedback-controlled trajectory of $\mathcal{OPT1}$ and $\mathcal{OPT2}$. In the augmented state space \overline{S} , a feedback control policy is a deterministic Markov policy as a function of an augmented state (x, q). As the system actually evolves in the original state space S, and the martingale state q can be seen as a random parameter at each state x, the feedback control policy is a randomized policy. In Fig. 2(b), we show a modified feedbackcontrolled trajectory. We continue the illustration in Fig. 2(a). When the martingale state along the trajectory is at least the corresponding value provided by Υ , the system starts following a deterministic control policy $\mu_n(\cdot, 1)$ of the unconstrained problem.

of the unconstrained problem if the martingale state is large enough, and Lines 5-13 perform a Bellman update to find the best augmented control if otherwise. When the system starts using deterministic policies of the unconstrained problem, we can set the martingale state to 1.0 and set the optimal martingale control to 0 in the following control period.

D. Complexity

The time complexity per iteration of Algorithms 1-5 is $O(|\overline{S}_n|^{\theta}(\log |\overline{S}_n|)^2)$. The space complexity of the iMDP algorithm is $O(|\overline{S}_n|)$ where $|\overline{S}_n| = \Theta(n)$ 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, 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 $J_n^*(\cdot, 1)$. As a consequence, it follows that Υ_n converges to Υ uniformly in probability. Using the same proof, we conclude that $\gamma_n(\cdot)$ and $J_n^{\gamma}(\cdot)$ converges uniformly to $\gamma(\cdot)$

and $J^*(\cdot, \gamma)$ in probability respectively. Therefore, we have incrementally constructed the boundary values on ∂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 $\{\overline{\mathcal{M}}_n\}_{n=0}^{\infty}$. In the following theorem, we formally summarize the key convergence results of the extended iMDP algorithm.

Theorem 3 Let \mathcal{M}_n and $\overline{\mathcal{M}}_n$ be two MDPs with discrete states constructed in S and \overline{S} respectively, and let J_n : $\overline{S}_n \to \overline{\mathbb{R}}$ be the cost-to-go 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 random variables converge in probability:

- 1) $\operatorname{plim}_{n \to \infty} ||J_n(\cdot, 1) J^*(\cdot, 1)||_{S_n} = 0,$
- 2) $\operatorname{plim}_{n \to \infty}^{n \to \infty} ||\Upsilon_n \Upsilon||_{S_n} = 0,$
- 2) $\operatorname{plim}_{n \to \infty} || I_n I || S_n = 0,$ 3) $\operatorname{plim}_{n \to \infty} || \gamma_n \gamma || S_n = 0,$ 4) $\operatorname{plim}_{n \to \infty} || J_n^{\gamma} J^{\gamma} || S_n = 0,$ 5) $\operatorname{plim}_{n \to \infty} || J_n J^* || \overline{S_n} = 0.$

The first four events construct the boundary values on ∂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 function J_n to the optimal cost-to-go function J^* on the augmented state space \overline{S} .



(a) Policy on \mathcal{M}_{500} .



(b) Policy on \mathcal{M}_{1000} .



(e) Markov chain implied by \mathcal{M}_{500} .



(c) Policy on \mathcal{M}_{3000} .



(f) Markov chain implied by \mathcal{M}_{1000} .

Fig. 3. 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 a terminal cost, which is -1000 for the goal region (G) and 10 for the obstacle region (Γ), with a discount factor $\alpha = 0.9$. Figures 3(a)-3(c) depict anytime policies on the boundary $S \times 1.0$ over iterations. Figures 3(d)-3(f) show the Markov chains created by anytime policies on \mathcal{M}_n over iterations.



(d) Policy map induced by γ_{4000} .

(e) Value function J_{4000}^{γ} .

(f) Min-collision prob. γ_{4000} .

Fig. 4. Figures 4(a)-4(c) shows a policy map, cost value function and the associated collision probability function for the unconstrained problem after 4000 iterations. Similar, Figures 4(d)-4(f) show a policy map, the associated value function, and the min-collision probability function after 4000 iterations. These values provide the boundary values for the stochastic target problem. For the unconstrained problem, the policy map encourages the system to go through the narrow corridors with low cost-to-go values and high probabilities of collision. In contrast, the policy map from the min-collision probability problem encourages the system to detour around the obstacles with high cost-to-go values and low probabilities of collision.

VI. EXPERIMENTS

In the following experiments, we used a computer with a 2.0-GHz Intel Core 2 Duo T6400 processor and 4 GB



(d) Markov chain implied by $\overline{\mathcal{M}}_{200}$.

(e) Markov chain implied by $\overline{\mathcal{M}}_{500}$.

(f) Markov chain implied by $\overline{\mathcal{M}}_{1000}$.

Fig. 5. Figures 5(a)-5(c) and Figures 5(d)-5(f) show the corresponding anytime policies and the associated Markov chains on $\overline{\mathcal{M}}_n$ respectively. In Fig. 5(c), we show the top-down view of a policy for states in $\overline{\mathcal{M}}_{3000} \setminus \mathcal{M}_{3000}$. We observe that the system will try to avoid the narrow corridors when the risk tolerance is low. We can also observe that the structures of the Markov chains quickly cover the state spaces S and \overline{S} with connected random graphs.



Fig. 6. Examples of incremental value functions over iterations. Figure 6(a)-6(c) show the approximate cost-to-go functions J_n when the probability threshold η_0 is 1.0 for n = 200, 2000 and 4000. Figures 6(d)-6(f) present the approximate cost-to-go function J_{4000} in $\overline{\mathcal{M}}_{4000}$ for augmented states where their martingale components are 0.1, 0.5 and 0.9 respectively. The plot shows that the lower the martingale state is, the higher the cost value is – which is consistent with the characteristics in Section III-C.

of RAM. We controlled a system with stochastic single

integrator dynamics to a goal region with free ending time in



probability 49.27%, average cost -125.20.

(b) Min-collision trajectories: simulated collision probability 0%, average cost -17.85.

Fig. 7. Examples of trajectories from policies of the unconstrained problem (Fig. 7(a)) and the min-collision probability problem (Fig. 7(b)). In the unconstrained problem, the system takes risk to go through one of the narrow corridors to reach the goal. In contrast, in the min-collision probability problem, the system detours around the obstacles to reach the goal. While there are about 49.27% of 2000 trajectories (plotted in red) that collide with the obstacles for the former, we observe no collision out of 2000 trajectories for the latter.



(a) An example of controlled trajectories using boundary values.

(b) Failure ratios for the first N trajectories ($N \leq 5000$) with different η .

Fig. 8. In Fig. 8(a), we show an example of controlled trajectories using boundary values. The system starts from (6.5, -3) with the failure-probability threshold $\eta = 0.4$. The martingale state varies along controlled trajectories as a random parameter in a randomized control policy. When the martingale state is above Υ , the system follows a deterministic control policy obtained from the unconstrained problem. In Fig. 8(b), we show failure ratios for the first N trajectories $(1 \le N \le 5000)$ starting from (6.5, -3) with different values of η . As seen in Fig. 8(b), the algorithm is able to keep the failure ratio in 5000 executions around 0.40 as dictated by the choice of $\eta = 0.40$ at time 0. Other failure ratios follow very closely the values of η , which indicates that the iMDP algorithm is able to provide solutions that are probabilistically sound.

a cluttered environment. The dynamics is given by dx(t) = u(t)dt + Fdw(t) where $x(t) \in \mathbb{R}^2$, $u(t) \in \mathbb{R}^2$, and $F = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$. 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 G at (8, 8), which is measured as the integral of square of control magnitude, and a terminal cost, which is -1000 for the goal region G and 10 for the obstacle region Γ , with a discount factor $\alpha = 0.9$. The maximum velocity of the system in the x and y directions is one. At the beginning, the system starts from (6.5, -3). 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 over iterations in Fig. 3. In particular, Figs. 3(a)-3(c) depict anytime policies on the boundary $S \times 1.0$ after 500, 1000, and 3000 iterations. Figures 3(d)-3(f) show the 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 (asymptotically almost-surely) connected to cover the state space S. As in the original version of iMDP, it



Fig. 9. Trajectories after 5000 iterations starting from (6.5, -3). In Figs. 9(a)-9(c) and Figs. 9(g)-9(i), we show 50 trajectories resulting from a policy induced by J_{4000} with different collision-probability thresholds ($\eta = 0.01, 0.05, 0.10, 0.20, 0.30, 0.40$). In Figs. 9(d)-9(f) and Figs. 9(j)-9(l), we show 5000 corresponding trajectories in the original state space S with *simulated collision probabilities* and *average costs* in their captions. Trajectories that reach the goal region are plotted in blue, and trajectories that hit obstacles are plotted in red.

is worth noting that the structures of these Markov chains can be constructed on-demand during the execution of the algorithm.

The sequence of approximating MDPs on S provides boundary values for the stochastic target problem as shown in Fig. 4. In particular, Figs. 4(a)-4(c) shows a policy map, cost value function $J_{4000,1.0}$ and the associated collision probability function Υ_{4000} for the unconstrained problem after 4000 iterations. Similarly, Figs. 4(d)-4(f) show a policy map, the associated value function J^{γ}_{4000} , and the mincollision probability function γ_{4000} after 4000 iterations. As we can see, for the unconstrained problem, the policy map encourages the system to go through the narrow corridors with low cost-to-go values and high probabilities of collision. In contrast, the policy map from the min-collision probability problem encourages the system to detour around the obstacles with high cost-to-go values and low probabilities of collision.

We now show how the extended iMDP algorithm constructs the sequence of approximating MDPs on the augmented state space \overline{S} . Figures 5(a)-5(c) show the corresponding anytime policies in \overline{S} over iterations. In Fig. 5(c), we show the top-down view of a policy for states in $\overline{\mathcal{M}}_{3000} \setminus \mathcal{M}_{3000}$. Compared to Fig 3(c), we observe that the system will try to avoid the narrow corridors when the risk tolerance is low. In Figs. 5(d)-5(f), we show the 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 \overline{S} with (asymptotically almostsurely) connected random graphs.

We then examine how the algorithm computes the value functions for the interior D^o of the reformulated stochastic target problem in comparison with the value function of the unconstrained problem in Fig. 6. Figure 6(a)-6(c) show approximate cost-to-go J_n when the probability threshold η_0 is 1.0 for n = 200, 2000 and 4000. We recall that the value functions in these figures form the boundary conditions on $S \times 1$, which is a subset of ∂D . In the interior D^o , Figs. 6(d)-6(f) 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 η . To do this, we first show how the policies of the unconstrained problem and the min-collision probability problem perform in Fig. 7. As we can see, in the unconstrained problem, the system takes risk to go through one of the narrow corridors to reach the goal. In contrast, in the min-collision probability problem, the system detour around the obstacles to reach the goal. While there are about 49.27% of 2000 trajectories (plotted in red) that collide with the obstacles for the former, we observe no collision out of 2000 trajectories for the latter. From the characteristics presented in Section III-C and illustrated in Fig. 2(b), from the starting state (6.5, -3), for any initial collision probability threshold η above 0.4927, we can execute the deterministic policy of the unconstrained problem.

In Fig. 8(a), we provide an example of controlled trajectories that are illustrated in Fig. 2(b) when the system starts from (6.5, -3) with the failure probability threshold $\eta = 0.4$. In this figure, the min-collision probability function γ_{4000} is plotted in blue, and the collision probability function Υ_{4000} is plotted in green. Starting from the augmented state (6.5, -3, 0.40), the martingale state varies along controlled trajectories as a random parameter in a randomized control policy. When the martingale state is above Υ_{4000} , the system follows a deterministic control policy obtained from the unconstrained problem.

Similarly, in Fig. 9, we show controlled trajectories for different values of η (0.01, 0.05, 0.10, 0.20, 0.30, 0.40). In

 TABLE I

 FAILURE RATIOS AND AVERAGE COSTS FOR FIG. 8(B).

η	Failure Ratio	Average Cost
1.00	0.4927	-125.20
0.40	0.4014	-115.49
0.30	0.2819	-76.80
0.20	0.1560	-65.81
0.10	0.1024	-58.00
0.05	0.0420	-42.53
0.01	0.0084	-19.42
0.001	0.0000	-18.86

Figs. 9(a)-9(c) and Figs. 9(g)-9(i), we show 50 trajectories resulting from a policy induced by J_{4000} with different initial collision probability thresholds. In Figs. 9(d)-9(f) and Figs. 9(j)-9(l), we show 5000 corresponding trajectories in the original state space S with reported *simulated collision probabilities* and *average costs* in their captions. Trajectories that reach the goal region are plotted in blue, and trajectories that hit obstacles are plotted in red. These simulated collision probabilities and average costs are shown in Table I. As we can see, the lower the threshold is, the higher the average cost is as we expect. When $\eta = 0.01$, the average cost is -19.42 and when $\eta = 1.0$, the average cost is -125.20.

More importantly, the simulated collision probabilities follow very closely the values of η chosen at time 0. In Fig. 8(b), we plot these simulated probabilities for the first N trajectories where $N \in [1, 5000]$ to show that the algorithm fully respects the bounded failure probability. Thus, this observation indicates that the extended iMDP algorithm is able to manage the risk tolerance along trajectories in different executions to minimize the 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 subject to bounded failure probabilities for initial states. We present here the martingale approach that diffuses the probability constraint into a martingale. The martingale stands for the level of risk tolerance that is contingent on available information over time. The approach transforms the probability-constrained problem into an equivalent stochastic target problem with the augmented state and control spaces. The boundary conditions for the transformed problem is, however, unspecified. The extended iMDP algorithm incrementally computes the boundary values and any-time feedback control policies for the transformed problem using asynchronous value iterations. The returned policies can be considered as randomized policies in the original state space. Effectively, the extended iMDP algorithm provides probabilistically-sound and asymptoticallyoptimal control policies for the class of stochastic control problems with bounded failure-probability constraints.

The future extension of the work is broad. We intend 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

- 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, "An optimal one-way multigrid algorithm for discrete-time stochastic control," *IEEE Transactions on Automatic Control*, vol. AC-36, pp. 898–914, 1991.
- [9] R. Munos, A. Moore, and S. Singh, "Variable resolution discretization in optimal control," in *Machine Learning*, 2001, pp. 291–323.
- [10] L. Grüne, "An adaptive grid scheme for the discrete hamilton-jacobibellman equation," *Numerische Mathematik*, vol. 75, pp. 319–337, 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.
- [15] V. A. Huynh, S. Karaman, and E. Frazzoli, "An incremental samplingbased algorithm for stochastic optimal control," in *ICRA*, 2012, pp. 2865–2872.
- [16] ——, "An incremental sampling-based algorithm for stochastic optimal control," arXiv:1202.5544v1 [cs.RO], 2012.
- [17] H. J. Kushner and P. G. Dupuis, Numerical Methods for Stochastic Control Problems in Continuous Time (Stochastic Modelling and Applied Probability). Springer, Dec. 2000.
- [18] H. J. Kushner and H. Joseph, Probability methods for approximations in stochastic control and for elliptic equations. Academic Press New York, 1977, vol. 129.
- [19] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Iowa State University, Ames, IA, Tech. Rep. 98-11, Oct. 1998.
- [20] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, 2010.
- [21] A. G. Banerjee, M. Ono, N. Roy, and B. C. Williams, "Regressionbased LP solver for chance-constrained finite horizon optimal control with nonconvex constraints," in *Proceedings of the American Control Conference*, San Francisco, CA, 2011.

- [22] Y. L. Chow and M. Pavone, "Stochastic optimal control with dynamic, time-consistent risk constraints," in *American Control Conference* (ACC), 2012. IEEE, 2012. Submitted.
- [23] D. E. Kirk, Optimal Control Theory: An Introduction. Dover Publications, Apr. 2004.
- [24] P. Kosmol and M. Pavon, "Lagrange approach to the optimal control of diffusions," *Acta Applicandae Mathematicae*, vol. 32, pp. 101–122, 1993, 10.1007/BF00998149.
- [25] —, "Solving optimal control problems by means of general lagrange functionals," *Automatica*, vol. 37, no. 6, pp. 907 – 913, 2001.
- [26] L. Blackmore, H. Li, and B. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *in Proceedings of the American Control Conference*, 2006.
- [27] M. Ono and B. C. Williams, "Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint," in *CDC*, 2008, pp. 3427–3432.
- [28] B. Luders, M. Kothari, and J. P. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in AIAA Guidance, Navigation, and Control Conference (GNC), Toronto, Canada, August 2010, (AIAA-2010-8160).
- [29] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in AIAA Guidance, Navigation, and Control Conference (GNC), Boston, MA, August 2013.
- [30] B. D. Luders, S. Karaman, E. Frazzoli, and J. P. How, "Bounds on tracking error using closed-loop rapidly-exploring random trees," in *American Control Conference (ACC)*, 2010. IEEE, 2010, pp. 5406– 5412.
- [31] R. C. Chen and G. L. Blankenship, "Dynamic programming equations for discounted constrained stochastic control," *Automatic Control, IEEE Transactions on*, vol. 49, no. 5, pp. 699–709, 2004.
- [32] A. Piunovskiy, "Dynamic programming in constrained markov decision processes," *Control and Cybernetics*, vol. 35, no. 3, p. 645, 2006.
- [33] S. Mannor and J. Tsitsiklis, "Mean-variance optimization in markov decision processes," arXiv preprint arXiv:1104.5601, 2011.
- [34] P. Huang, D. A. Iancu, M. Petrik, and D. Subramanian, "The price of dynamic inconsistency for distortion risk measures," *arXiv preprint* arXiv:1106.6102, 2011.
- [35] A. Ruszczyński and A. Shapiro, "Optimization of risk measures," in Probabilistic and randomized methods for design under uncertainty. Springer, 2006, pp. 119–157.
- [36] —, "Conditional risk mappings," *Mathematics of Operations Research*, vol. 31, no. 3, pp. 544–561, 2006.
- [37] B. Rudloff, A. Street, and D. Valladao, "Time consistency and risk averse dynamic decision models: Interpretation and practical consequences," *Internal Research Reports*, vol. 17, 2011.
- [38] B. Bouchard and N. Touzi, "Weak dynamic programming principle for viscosity solutions," *SIAM Journal on Control and Optimization*, vol. 49, no. 3, pp. 948–962, 2011.
- [39] B. Bouchard, R. Elie, and C. Imbert, "Optimal control under stochastic target constraints," *SIAM Journal on Control and Optimization*, vol. 48, no. 5, pp. 3501–3531, 2010.
- [40] B. Bouchard, R. Elie, and N. Touzi, "Stochastic target problems with controlled loss," *SIAM Journal on Control and Optimization*, vol. 48, no. 5, pp. 3123–3150, 2009.
- [41] N. Touzi and A. Tourin, Optimal stochastic control, stochastic target problems, and backward SDE. Springer, 2013, vol. 29.
- [42] B. Bouchard and M. Nutz, "Weak dynamic programming for generalized state constraints," *SIAM Journal on Control and Optimization*, vol. 50, no. 6, pp. 3344–3373, 2012.
- [43] H. M. Soner and N. Touzi, "Dynamic programming for stochastic target problems and geometric flows," *Journal of the European Mathematical Society*, vol. 4, no. 3, pp. 201–236–236, Sept. 2002.
- [44] B. Bouchard and T. N. Vu, "The obstacle version of the geometric dynamic programming principle: Application to the pricing of american options under constraints," *Applied Mathematics and Optimization*, vol. 61, no. 2, pp. 235–265, 2010.
- [45] V. A. Huynh and E. Frazzoli, "Probabilistically-sound and asymptotically-optimal algorithm for stochastic control with trajectory constraints," in *Decision and Control (CDC)*, 2012 IEEE 51st Annual Conference on. IEEE, 2012, pp. 1486–1493.
- [46] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus* (*Graduate Texts in Mathematics*), 2nd ed. Springer, Aug. 1991.
- [47] D. Lamberton and B. Lapeyre, *Introduction to stochastic calculus applied to finance*. Chapman & Hall, 2008.

[48] B. Oksendal, Stochastic differential equations (3rd ed.): an introduction with applications. New York, NY, USA: Springer-Verlag New York, Inc., 1992.