

Risk-Sensitive Path Planning via CVaR Barrier Functions: Application to Bipedal Locomotion

Mohamadreza Ahmadi, Xiaobin Xiong, and Aaron D. Ames

Abstract—Enforcing safety of robotic systems in the presence of stochastic uncertainty is a challenging problem. Traditionally, researchers have proposed safety in the statistical mean as a safety measure in this case. However, ensuring safety in the statistical mean is only reasonable if robot safe behavior in the large number of runs is of interest, which precludes the use of mean safety in practical scenarios. In this paper, we propose a risk sensitive notion of safety called conditional-value-at-risk (CVaR) safety, which is concerned with safe performance in the worst case realizations. We introduce CVaR barrier functions as a tool to enforce CVaR safety and propose conditions for their Boolean compositions. Given a legacy controller, we show that we can design a *minimally interfering* CVaR safe controller via solving difference convex programs. We elucidate the proposed method by applying it to a bipedal locomotion case study.

I. INTRODUCTION

With the rise of robotic systems being deployed in real-world settings, the associated risk that stems from uncertain and unforeseen circumstances is correspondingly on the rise. There are several inherent sources of uncertainty in robotics systems, such as modeling uncertainty, sensor range and resolution limitations, highly dynamic and uncertain environments, noise and wear-and-tear in robot actuation [1], that lead to higher risk in robot deployment.

Mathematically speaking, risk can be quantified in numerous ways, such as chance constraints [2], [3]. However, applications in autonomy and robotics require more “nuanced assessments of risk” [4]. Artzner *et. al.* [5] characterized a set of natural properties that are desirable for a risk measure, called a coherent risk measure, and have obtained widespread acceptance in finance and operations research, among other fields. An important example of a coherent risk measure is the conditional value-at-risk (CVaR) that has received significant attention in decision making problems, such as Markov decision processes (MDPs) [6], [7], [8], [9]. For stochastic discrete-time dynamical systems, a model predictive control technique with coherent risk objectives was proposed in [10], wherein the authors also proposed Lyapunov condition for risk-sensitive exponential stability. Moreover, a method based on stochastic reachability analysis was proposed in [11] to estimate a CVaR-safe set of initial conditions via the solution to an MDP.

In this work, we use a special class of barrier function as a tool for enforcing risk-sensitive safety. Control barrier functions have been proposed in [12] and have been used for designing safe controllers (in the absence of a legacy

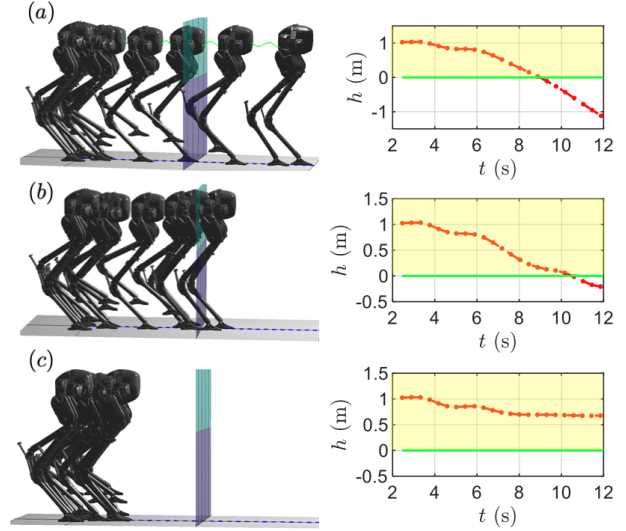


Fig. 1. Risk-sensitive obstacle avoidance using CVaR barrier functions (robot behavior and barrier function evolution). The shaded yellow area denotes safe regions. (a) safety violation with no barrier function; (b) safety violation with risk-neutral (statistical mean) barrier function; (c) safe behavior with CVaR barrier function. Plots on the right side show the values of the barrier functions.

controller, i.e., a desired controller that may be unsafe) and safety filters (in the presence of a legacy controller) for continuous-time dynamical systems, such as bipedal robots [13] and trucks [14], with guaranteed robustness [15], [16]. For discrete-time systems, discrete-time barrier functions have been formulated in [17], [18] and applied to the multi-robot coordination problem [19]. Recently, for a class of stochastic (Ito) differential equations, safety in probability and statistical mean was studied in [20], [21] via stochastic barrier functions.

In this paper, we go beyond the conventional notions of safety in probability and statistical mean for discrete-time systems subject to stochastic uncertainty. To this end, we define safety in the risk-sensitive CVaR sense, which is concerned with safety in the worst possible scenarios. We then propose CVaR barrier functions as a tool to enforce CVaR safety and formulate conditions for their Boolean compositions. We propose a computational method based on difference convex programs (DCPs) to synthesize CVaR safe controllers for stochastic linear discrete-time systems. These CVaR safe controllers are designed such that they minimally interfere with a given robot legacy controller. We show the efficacy of our proposed method on collision avoidance scenarios involving a bipedal robot subject to modeling uncertainty (see Figure 1).

The rest of the paper is organized as follows. In the

The authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125. email: {mrahmadi, xxiong, ames}@caltech.edu.

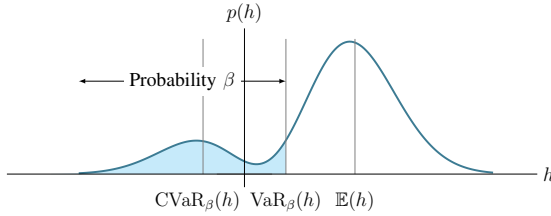


Fig. 2. Comparison of the mean, VaR, and CVaR for a given confidence $\beta \in (0, 1)$. The axes denote the values of the stochastic variable h and its pdf $p(h)$. The shaded area denotes the $\% \beta$ of the area under $p(h)$. If $h \geq 0$ represents a safe behavior, using $\mathbb{E}(h)$ as a performance measure is misleading (note that $\mathbb{E}(h)$ is positive). VaR gives the value of h at the β -tail of the distribution. But, it ignores the values of h with probability below β . CVaR is the average of the values of VaR with probability less than β (average of the worst-case values of h in the β tail of the distribution). Note that CVaR_β is negative indicating unsafe behavior.

next section, we introduce CVaR safety and formulate CVaR barrier functions as a tool to synthesize risk-sensitive safe controllers. In Section III, we discuss our bipedal locomotion case study. In Section IV, we present the obtained results from high fidelity simulations. Finally, in Section V, we conclude the paper and give directions for future research.

Notation: We denote by \mathbb{R}^n the n -dimensional Euclidean space and $\mathbb{N}_{\geq 0}$ the set of non-negative integers. For a finite set \mathcal{A} , we denote by $|\mathcal{A}|$ the number of elements of \mathcal{A} . For a probability space $(\mathcal{X}, \mathcal{F}, \mathbb{P})$ and a constant $p \in [1, \infty)$, $\mathcal{L}_p(\mathcal{X}, \mathcal{F}, \mathbb{P})$ denotes the vector space of real valued random variables X for which $\mathbb{E}|X|^p < \infty$. The Boolean operators are denoted by \neg (negation), \vee (conjunction), and \wedge (disjunction). For a risk measure ρ , we denote ρ^t to show the function composition of ρ with itself t times.

II. CVAR BARRIER FUNCTIONS FOR RISK-SENSITIVE PATH PLANNING

In this section, we formulate the risk-sensitive safety problem and propose a solution based on a special class of barrier functions. We begin by defining our risk measure of interest called CVaR.

A. Conditional Value-at-Risk

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, $\mathcal{H} = \mathcal{L}_p(\Omega, \mathcal{F}, \mathbb{P})$, $p \in [0, \infty)$, and let $h \in \mathcal{H}$ be a stochastic variable for which higher values are of interest (for example, greater values of h indicate safer performance). For a given confidence level $\beta \in (0, 1)$, value-at-risk (VaR_β) denotes the β -quantile value of a stochastic variable $h \in \mathcal{H}$ described as

$$\text{VaR}_\beta(h) = \sup_{\zeta \in \mathbb{R}} \{\zeta \mid \mathbb{P}(h \leq \zeta) \leq \beta\}.$$

Unfortunately, working with VaR for non-normal stochastic variables is numerically unstable, optimizing models involving VaR are intractable in high dimensions, and VaR ignores the values of h with probability less than β [22].

In contrast, CVaR overcomes the shortcomings of VaR. CVaR with confidence level $\beta \in (0, 1)$ denoted CVaR_β measures the expected loss in the β -tail given that the particular threshold VaR_β has been crossed, i.e., $\text{CVaR}_\beta(h) = \mathbb{E}[h \mid h \leq \text{VaR}_\beta(h)]$. An optimization formulation for CVaR

was proposed in [22] that we use in this paper. That is, CVaR_β is given by

$$\text{CVaR}_\beta(h) := \inf_{\zeta \in \mathbb{R}} \mathbb{E} \left[\zeta + \frac{(h - \zeta)_+}{\beta} \right], \quad (1)$$

where $(\cdot)_+ = \max\{\cdot, 0\}$. A value of $\beta \rightarrow 1$ corresponds to a risk-neutral case, i.e., $\text{CVaR}_1(h) = \mathbb{E}(h)$; whereas, a value of $\beta \rightarrow 0$ is rather a risk-averse case, i.e., $\text{CVaR}_0(h) = \text{VaR}_0(h)$ [23]. Figure 2 illustrates these notions for an example h variable with distribution $p(h)$.

Unlike VaR, CVaR is coherence is a coherent risk measure [24], [25], which satisfies the following properties.

Definition 1 (Coherent Risk Measure): We call a risk measures $\rho : \mathcal{H} \rightarrow \mathbb{R}$ a coherent risk measure, if it satisfies the following conditions

- **Convexity:** $\rho(\lambda h + (1 - \lambda)h') \leq \lambda \rho(h) + (1 - \lambda)\rho(h')$, for all $\lambda \in (0, 1)$ and for all $h, h' \in \mathcal{H}$;
- **Monotonicity:** If $h \leq h'$ then $\rho(h) \leq \rho(h')$ for all $h, h' \in \mathcal{H}$;
- **Translational Invariance:** $\rho(h + c) = \rho(h) + c$ for all $h \in \mathcal{H}$ and $c \in \mathbb{R}$;
- **Positive Homogeneity:** $\rho(\beta h) = \beta \rho(h)$ for all $h \in \mathcal{H}$ and $\beta \geq 0$.

In fact, we use the nice mathematical properties of CVaR given in Definition 1 in the proofs of our main results in Section II-C.

B. CVaR Safety

We assume the robot dynamics of interest is described by a discrete-time stochastic system given by

$$x^{t+1} = f(x^t, u^t, w^t), \quad x^0 = x_0, \quad (2)$$

where $t \in \mathbb{N}_{\geq 0}$ denotes the time index, $x \in \mathcal{X} \subset \mathbb{R}^n$ is the state, $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, $w \in \mathcal{W}$ is the stochastic uncertainty/disturbance, and the function $f : \mathbb{R}^n \times \mathcal{U} \times \mathcal{W} \rightarrow \mathbb{R}^n$. We assume that the initial condition x_0 is deterministic and that $|\mathcal{W}|$ is finite, i.e., $\mathcal{W} = \{w_1, \dots, w_{|\mathcal{W}|}\}$. At every time-step t , for a state-control pair (x^t, u^t) , the process disturbance w^t is drawn from set \mathcal{W} according to the probability density function $p(w) = [p(w_1), \dots, p(w_{|\mathcal{W}|})]^T$, where $p(w_i) := \mathbb{P}(w^t = w_i)$, $i = 1, 2, \dots, |\mathcal{W}|$. Note that the probability mass function for the process disturbance is time-invariant, and that the process disturbance is independent of the process history and of the state-control pair (x^t, u^t) .

We are interested in studying the properties of the solutions to (2) with respect to the compact set \mathcal{S} described as

$$\mathcal{S} := \{x \in \mathcal{X} \mid h(x) \geq 0\}, \quad (3a)$$

$$\text{Int}(\mathcal{S}) := \{x \in \mathcal{X} \mid h(x) > 0\}, \quad (3b)$$

$$\partial \mathcal{S} := \{x \in \mathcal{X} \mid h(x) = 0\}, \quad (3c)$$

where $h : \mathcal{X} \rightarrow \mathbb{R}$ is a continuous function. For instance, \mathcal{S} can represent robot constraints, e.g. joint limits, safe exploration region, and etc.

In the presence of stochastic uncertainty w , assuring almost sure (with probability one) invariance or safety may not be feasible. Moreover, enforcing safety in expectation is only meaningful if the law of large numbers can be invoked and we are interested in the long term performance, independent of the realization fluctuations. In this work, instead, we propose safety in a dynamic coherent risk measure, namely, CVaR, sense with conditional expectation (risk-neutral case) as an special case $\beta \rightarrow 1$.

Definition 2 (CVaR Safety): Given a safe set \mathcal{S} as given in (3) and a confidence level $\beta \in (0, 1)$, we call the solutions to (2) starting at $x_0 \in \mathcal{S}$ CVaR safe if and only if

$$\text{CVaR}_\beta^t(h(x^t)) \geq 0, \quad \forall t \geq 0. \quad (4)$$

Note that CVaR_β^t a dynamic time-consistent risk measure [26, Definition 3], i.e., if for some two realizations w and w' , $h_w(x^\theta) \geq h_{w'}(x^\theta)$ at some future time θ , and $h_w(x^t) = h_{w'}(x^t)$ for time $t \in (\tau, \theta)$, then $h_w(x^t) \not\leq h_{w'}(x^t)$ for $t < \tau$. The time consistency property ensures that contradictory evaluations of safety risk at different points in time does not happen. In other words, if one realization of w incurs higher safety risk at some point in time, then it is a riskier in terms of safety at any prior point in time.

C. CVaR Barrier Functions

In order to check and enforce CVaR safety, we define CVaR barrier functions.

Definition 3 (CVaR Barrier Function): For the discrete-time system (2) and a confidence level $\beta \in (0, 1)$, the continuous function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is a CVaR barrier function for the set \mathcal{S} as defined in (3), if there exists a constant $\alpha \in (0, 1)$ such that

$$\text{CVaR}_\beta(h(x^{t+1})) \geq \alpha h(x^t), \quad \forall x^t \in \mathcal{X}. \quad (5)$$

In the next result, we demonstrate that the existence of a CVaR barrier function indeed implies CVaR safety.

Theorem 1: Consider the discrete-time system (2) and the set \mathcal{S} as described in (3). Let $\beta \in (0, 1)$ be a given confidence level. Then, \mathcal{S} is CVaR safe, if there exists an CVaR barrier function as defined in Definition 3.

Proof: The proof is carried out by induction and using the properties of CVaR as a coherent risk measure as outlined in Definition 1. If (5) holds, for $t = 0$, we have

$$\text{CVaR}_\beta(h(x^1)) \geq \alpha h(x_0). \quad (6)$$

Similarly, for $t = 1$, we have

$$\text{CVaR}_\beta(h(x^2)) \geq \alpha h(x_1). \quad (7)$$

Since CVaR_β is monotone (because it is a coherent risk measure), composing both sides of (7) with CVaR_β does not change the inequality and we obtain

$$\text{CVaR}_\beta^2(h(x^2)) \geq \text{CVaR}_\beta(\alpha h(x_1)). \quad (8)$$

Since $\alpha \in (0, 1)$, from positive homogeneity property of CVaR, we obtain $\text{CVaR}_\beta(\alpha h(x_1)) = \alpha \text{CVaR}_\beta(h(x_1))$. Thus, (8) simplifies to $\text{CVaR}_\beta^2(h(x^2)) \geq \alpha \text{CVaR}_\beta(h(x_1))$.

Then, using inequality (6), we have

$$\text{CVaR}_\beta^2 \geq \alpha \text{CVaR}_\beta(h(x^1)) \geq \alpha^2 h(x_0).$$

Therefore, by induction, at time t , we can show that

$$\text{CVaR}_\beta^t(h(x^t)) \geq \alpha^t h(x_0). \quad (9)$$

If $x_0 \in \mathcal{S}$, from the definition of the set \mathcal{S} , we have $h(x_0) \geq 0$. Since $\alpha \in (0, 1)$, then we can infer that (4) holds. Thus, the system is CVaR $_\beta$ -safe. \blacksquare

In many practical robotics path planning scenarios, we encounter multiple obstacles and safe sets composed of Boolean compositions of several barrier functions [27], [19], [28]. Next, we propose conditions for checking Boolean compositions of CVaR barrier functions.

Proposition 1: Let $\mathcal{S}_i = \{x \in \mathbb{R}^n \mid h_i(x) \geq 0\}$, $i = 1, \dots, k$ denote a family of safe sets with the boundaries and interior defined analogous to \mathcal{S} in (3). Consider the discrete-time system (2). If there exist a $\alpha \in (0, 1)$ such that

$$\text{CVaR}_\beta \left(\min_{i=1, \dots, k} h_i(x^{t+1}) \right) \geq \alpha \min_{i=1, \dots, k} h_i(x^t) \quad (10)$$

then the set $\{x \in \mathbb{R}^n \mid \bigwedge_{i=1, \dots, k} (h_i(x) \geq 0)\}$ is CVaR safe. Similarly, if there exist a $\alpha \in (0, 1)$ such that

$$\text{CVaR}_\beta \left(\max_{i=1, \dots, k} h_i(x^{t+1}) \right) \geq \alpha \max_{i=1, \dots, k} h_i(x^t) \quad (11)$$

then the set $\{x \in \mathbb{R}^n \mid \bigvee_{i=1, \dots, k} (h_i(x) \geq 0)\}$ is CVaR safe.

Proof: If (10) holds from the proof of Theorem 1, we can infer that

$$\text{CVaR}_\beta^t \left(\min_{i=1, \dots, k} h_i(x^t) \right) \geq \alpha^t \min_{i=1, \dots, k} h_i(x^0).$$

That is, if $x^0 \in \{x \in \mathbb{R}^n \mid \min_{i=1, \dots, k} h_i(x) \geq 0\}$, then $\text{CVaR}_\beta^t(\min_{i=1, \dots, k} h_i(x^t)) \geq 0$ for all $t \in \mathbb{N}_{\geq 0}$. Let $h_{i^*}(x^t)$ be the smallest among $h_i(x^t)$, $i = 1, 2, \dots, k$, i.e., it satisfies $h_j(x^t) \geq \dots \geq h_{i^*}(x^t)$, $\forall j \neq i^*$. Because CVaR is monotone (see Definition 1), the latter inequality implies $\text{CVaR}_\beta^t(h_j(x^t)) \geq \dots \geq \text{CVaR}_\beta^t(h_{i^*}(x^t))$, $\forall j \neq i^*$. Since $\text{CVaR}_\beta^t(\min_{i=1, \dots, k} h_i(x^t)) = \text{CVaR}_\beta^t(h_{i^*}(x^t)) \geq 0$ for all $t \in \mathbb{N}_{\geq 0}$, we have

$$\text{CVaR}_\beta^t(h_j(x^t)) \geq \dots \geq \text{CVaR}_\beta^t(h_{i^*}(x^t)) \geq 0, \quad j \neq i^*.$$

Thus, $\text{CVaR}_\beta^t(h_i(x)) \geq 0$ for all $i \in \{1, \dots, k\}$.

Similarly, if (11) holds, we can infer that

$$\text{CVaR}_\beta^t \left(\max_{i=1, \dots, k} h_i(x^t) \right) \geq \alpha^t \max_{i=1, \dots, k} h_i(x^0).$$

Hence, using similar arguments as the proof of the conjunction case, $\text{CVaR}_\beta^t(\max_{i=1, \dots, k} h_i(x^t)) \geq 0$ for all $t \in \mathbb{N}_{\geq 0}$. That is, there exists at least an $i \in \{1, \dots, k\}$ for which $\text{CVaR}_\beta^t(\max_{i=1, \dots, k} h_i(x^t)) \geq 0$. \blacksquare

The negation operator is trivial and can be shown by checking if $-h$ satisfies inequality (5).

In the next section, we demonstrate how a sequence $\{u^t\}_{t>0}$ can be designed such that system (2) becomes CVaR safe based on optimization techniques.

D. CVaR-Safe Controller Synthesis

Inspired by the quadratic programming formulations of conventional control barrier functions in the continuous-time case [12], we pose the controller synthesis problem as an optimization.

CVaR Control Barrier Function Optimization: At every time step t , given x^t , a set \mathcal{S} as described in (3), a confidence level $\beta \in (0, 1)$, a parameter $\alpha \in (0, 1)$, control upper bounds \bar{u} , lower bounds \underline{u} , and a legacy controller u_{legacy}^t , solve

$$u_*^t = \underset{u^t}{\operatorname{argmin}} (u^t - u_{\text{legacy}}^t)^T (u^t - u_{\text{legacy}}^t)$$

subject to

$$\underline{u} \leq u^t \leq \bar{u}, \quad (12a)$$

$$\text{CVaR}_\beta(h(f(x^t, u^t, w))) \geq \alpha h(x^t). \quad (12b)$$

Note that instantaneous controls u^t are the only variables in the optimization. The cost function $(u^t - u_{\text{legacy}}^t)^T (u^t - u_{\text{legacy}}^t) = \|u^t - u_{\text{legacy}}^t\|^2$ ensures that u^t remains as close as possible to the legacy controller u_{legacy}^t in the Euclidean 2-norm; hence, it guarantees the minimally interference.

For general nonlinear h , optimization problem (12) is a nonlinear program in the decision variable u^t (note that CVaR is a convex function in h since it is a coherent risk measure). Indeed, this was the case for optimization problems designed for synthesizing discrete control barrier functions for discrete-time systems even without stochastic uncertainty [18], as well. MATLAB functions such as `fmincon` can be used to solve the nonlinear program.

Next, we show that under some assumptions the search over CVaR safe controls u^t can be carried out by solving difference convex programs (DCPs).

For the remainder of this section, we restrict our attention to the case when system (2) is a linear system. That is,

$$f(x^t, u^t, w^t) = A(w^t)x^t + B(w^t)u^t + G(w^t),$$

where $A : \mathcal{W} \rightarrow \mathbb{R}^{n \times n}$, $B : \mathcal{W} \rightarrow \mathbb{R}^{n \times m}$ and $G : \mathcal{W} \rightarrow \mathbb{R}^n$.

For such systems, we assume the CVaR barrier function takes the form of a linear function

$$h(x^t) = Hx^t + l, \quad (13)$$

where $H \in \mathbb{R}^{1 \times n}$ and $l \in \mathbb{R}$. Then, the term $\text{CVaR}_\beta(h(f(x^t, u^t, w)))$ in constraints (12b) changes to

$$\text{CVaR}_\beta(HA(w)x^t + HB(w)u^t + HG(w) + l). \quad (14)$$

Since CVaR_β is a convex function, the above term is a convex function in u^t , i.e., the control variable.

Re-writing optimization problem (12) for linear discrete time systems with stochastic uncertainty and CVaR barrier function (13) gives the following optimization problem

$$u_*^t = \underset{u^t}{\operatorname{argmin}} (u^t - u_{\text{legacy}}^t)^T (u^t - u_{\text{legacy}}^t)$$

subject to

$$\underline{u} \leq u^t \leq \bar{u}, \quad (15a)$$

$$(14) \geq \alpha h(x^t), \quad (15b)$$

Substituting the expression for CVaR (1) in (14) for uncertainty w with finite $|\mathcal{W}|$ yields

$$\inf_{\zeta \in \mathbb{R}} \left\{ \zeta + \frac{1}{\beta} \sum_{i=1}^{|\mathcal{W}|} (HA(w_i)x^t + HB(w_i)u^t + HG(w_i) + l - \zeta)_+ p(w_i) \right\}, \quad (16)$$

which introduces the extra decision variable $\zeta \in \mathbb{R}$.

Hence, (15) can be rewritten in the standard DCP form

$$u_*^t = \underset{u^t, \zeta}{\operatorname{argmin}} q_0(u^t)$$

subject to

$$q_1(u^t) \leq 0 \text{ and } q_2(u^t) \leq 0, \quad (17a)$$

$$q_3 - \hat{q}_4(\zeta, u^t) \leq 0, \quad (17b)$$

where wherein $q_0(u^t) = (u^t - u_{\text{legacy}}^t)^T (u^t - u_{\text{legacy}}^t)$ is a convex (quadratic) function, $q_1(u^t) = \underline{u} - u^t$ is a convex (linear) function, $q_2(u^t) = u^t - \bar{u}$ is a convex (linear) function, and $q_3 = \alpha Hx^t + \alpha l$ is a convex (constant) function. The expression for $\hat{q}_4(\zeta, u^t)$ is given in (16) which is a convex function in u^t and ζ since $\hat{q}_4(\zeta, u^t)$ is convex in ζ [22, Theorem 1] because the function $(\cdot)_+$ is increasing and convex [29, Lemma A.1.].

DCPs like (17) arise in many applications, such as feature selection in machine learning [30] and inverse covariance estimation in statistics [31]. In order to solve DCPs, we use a variant of the convex-concave procedure [32], [33], wherein the concave terms are replaced by a convex upper bound and solved. In fact, the disciplined convex-concave programming (DCCP) [33] technique linearizes DCP problems into a (disciplined) convex program (carried out automatically via the DCCP package [33]), which is then converted into an equivalent cone program by replacing each function with its graph implementation. Then, the cone program can be solved readily by available convex programming solvers, such as CVX [34] or YALMIP [35].

We should point out that solving (17) via the DCCP method, finds the (local) saddle points to optimization problem (17). Nonetheless, every such local u^t guarantees CVaR safety.

III. APPLICATION TO BIPEDAL LOCOMOTION

In this section, we apply the risk-sensitive planning on a bipedal walking robot. We first present the dynamics model of the walking. Then, we briefly review a stepping approach for realizing and stabilizing walking behaviors, which produces a discrete linear dynamics for path planning. Finally, we apply the risk-sensitive planning controller on bipedal walking.

A. Hybrid Dynamics of Bipedal Walking

Bipedal walking is a hybrid dynamical phenomenon. The dynamics description changes as the contact changes. For instance, the dynamics with two feet on the ground is different than that with one foot on the ground. The state also undergoes a discrete change when the swing foot strikes

the ground. We consider the dynamics to be described by two domains, i.e., the single support phase (SSP) and the double support phase (DSP). The walking transits from the SSP to DSP when the swing foot strikes the ground. It transits from DSP to SSP when one of the feet lifts off from the ground. Mathematically, the hybrid dynamical system can be described by a combination of nonlinear continuous dynamics and discrete transitions:

$$\dot{x}_v = f_v(x) + g_v(x)\tau, \quad (18)$$

$$x_{v+1}^+ = \Delta_v(x_v^-), \quad (19)$$

where x is the state of the robot, v denotes for the index of the domain, τ is the actuation torque vector of the robot, $+/-$ denotes for the initial or final instant of the domain, and Δ_v represents for the transition between each domains.

B. Linear S2S Approximation for Step Planning

Planning and controlling of bipedal walking is a challenging problem, and there has been various related approaches [36] in the literature. In this paper, we apply the approach in [37] to realize walking via stepping, where a discrete linear dynamics is utilized for approximating the actual dynamics of the walking robot.

Consider the state x at the instant before the swing foot strikes the ground (i.e., the impact event). We denote it as the pre-impact state. Assuming an existing walking behavior, the pre-impact state x^{t+1} of the next step is a function of the pre-impact state x^t at the current step and the actuation τ during the step. Then, it is a discrete dynamical system at the step level:

$$x^{t+1} = \mathcal{P}(x^t, \tau(t)), \quad (20)$$

where $\tau(t)$ denotes the applied torques during the step, which is referred to as the step-to-step (S2S) dynamics of the walking [37].

The, the horizontal state of the robot is critical for walking; the horizontal position and velocity of the robot mainly describe the walking w.r.t the world. Let $\mathbf{x}_h = [c, p, v]^T$ denote the horizontal state, where c is the horizontal position of the center of mass (COM) of the robot relative to the inertia frame, p is the horizontal position of the COM relative to its stance foot, and v is the horizontal velocities of the COM. Thus the horizontal S2S dynamics can be denoted as:

$$\mathbf{x}_h^{t+1} = \mathcal{P}^h(x^t, \tau(t)). \quad (21)$$

Our previous work [37], [38] suggests that the S2S dynamics of the walking of the Hybrid-Linear Inverted Pendulum (H-LIP) [39] can be used to approximate the actual horizontal S2S dynamics of the walking of the robot. The S2S dynamics of the H-LIP [37] is:

$$\mathbf{x}_{\text{H-LIP}}^{t+1} = A\mathbf{x}_{\text{H-LIP}}^t + Bu_{\text{H-LIP}}^t \quad (22)$$

where $\mathbf{x}_{\text{H-LIP}}^{t+1} = [c_{\text{H-LIP}}, p_{\text{H-LIP}}, v_{\text{H-LIP}}]^T$ is the discrete pre-impact state of the H-LIP, and $u_{\text{H-LIP}}^t$ is the step size. The

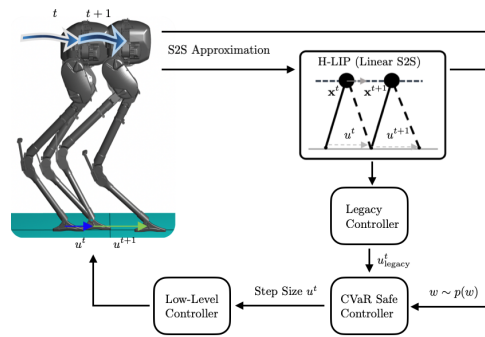


Fig. 3. A schematic diagram of our proposed risk-sensitive path planning method based on CVaR barrier functions for bipedal robots.

expressions of A, B can be found in [37]. By approximation, Eq. (21) can be rewritten as:

$$\mathbf{x}_h^{t+1} = A\mathbf{x}_h^t + Bu^t + w^t \quad (23)$$

$$w^t := \mathcal{P}^h(x^t, \tau(t)) - A\mathbf{x}_h^t - Bu^t. \quad (24)$$

where $w \in \mathcal{W}$ can be treated as the disturbance to the linear S2S dynamical system in Eq. (22). Applying the H-LIP based stepping [37]:

$$u = u_{\text{H-LIP}} + K(\mathbf{x}_h - \mathbf{x}_{\text{H-LIP}}) \quad (25)$$

on the robot yields the error $\mathbf{e} = \mathbf{x}_h - \mathbf{x}_{\text{H-LIP}}$ which evolves according to the error dynamics $\mathbf{e}^{t+1} = (A + BK)\mathbf{e}^t + w^t$. The error dynamics has an error (disturbance) invariant set E if $(A + BK)$ is stable. If $\mathbf{e}_k \in E$, then $\mathbf{e}^{t+1} \in E$. As a result, this stepping controller drives the robot to behavior approximately like the walking behavior of the H-LIP, the difference between which is bound by E .

For application of 3D bipedal walking, the H-LIP is applied in each plane of walking: the sagittal plane and the lateral plane. The H-LIP based planning provides the desired step sizes (step length and step width) for the robot, which become the desired outputs for the low-level controller to track [39].

C. Risk-Sensitive Path Planning

We apply the CVaR barrier function based risk-sensitive planning presented in Section II to the 3D bipedal walking as described in Figure 3. The uncertainty w is treated as stochastic uncertainty and as a risk factor that can lead to undesired behavior on the generated walking. For instance, in order to avoid an obstacle, one first generates a collision-free path on the H-LIP, and then apply the H-LIP based planning on the robot to create approximately similar walking. Due to the uncertainty w , the actual walking on the robot diverges with a bounded error (inside E) from the H-LIP walking, which potentially causes collision between the robot and the obstacle.

To circumvent this issue, we synthesize CVaR barrier functions based controllers to filter the H-LIP based stepping controller on the robot. The barrier functions are designed to represent the safe regions, where there are no obstacles. We are then interested in keeping the robot inside the safe (obstacle-free) regions during walking.

The uncertainty w is numerically calculated in simulation for a series of walking behaviors, which provided a polytopic set that bounds w . We took $|\mathcal{W}|$ random samples from the latter polytopic set. Since w is sparse in nature, we assumed a uniform distribution of w inside \mathcal{W} , i.e., $p(w) = 1/|\mathcal{W}|$. To design the risk-sensitive safe controllers, we then solve DCP (17), where A , B , and $G(w) = w$ are given by the approximated S2S dynamics (23).

IV. DEMONSTRATION

We apply the proposed approach in high-fidelity simulation on the underactuated bipedal robot Cassie [40]. The dynamics is integrated using Matlab ODE 45 function with event based function for detecting the domain transitions. The CVaR barrier function based optimization DCP (17) is solved in YALMIP using MOSEK solver at each step. The optimization typically takes $100 \sim 700$ steps under 10 seconds to solve on a laptop with the processor intel(R) Core(TM) i7-7700HQ@2.8GHz. The low-level controller on the robot is solved at 1kHz. The legacy controller used in our experiments is a model predictive controller.

Case 1: We consider a scenario, in which the robot is walking to follow a straight path. However, an obstacle is placed in the robot path. The results are shown in Fig. 1 (a) and 4. The legacy controller is not aware of this wall, which results in collision that in practice would cause collision and hardware failure.

Then, we apply a CVaR barrier function to filter the output of the legacy controller. The safe set is defined as

$$h(c_x) = p_x - c_x \geq 0,$$

where $p_x = 1$ is the position of the obstacle, c_x denotes the position of the robot in the forward direction. We first apply the CVaR barrier function with $\beta = 0.999$ (risk neutral) based risk-sensitive controller. The result is shown in Fig. 1 (b): the robot walks and stop at the location of the obstacle. However, due the stochastic uncertainty w , the risk-neutral path planning violates the safety requirement.

Then, we apply the CVaR barrier function with $\beta = 0.1$ (risk-averse case), which generates the walking shown in Fig. 1 (c). The legacy controller directs the robot forward, but the CVaR safe controller keeps the robot away from the obstacle.

Case 2: In this scenario, we consider the robot forward reference path. However, there is a wall at an angle, which does not completely prevent the robot from walking forward. The safe set is defined as

$$h(c_x, c_y) = c_y + k(c_x - p) \geq 0,$$

where k indicates the angle of the wall, p indicates the location of the wall in forward direction, and c_y is the position of the robot in the lateral plane. Here $k = -0.5$ and $p = 2$. Fig. 5 (a) shows the generated walking behavior. With the CVaR barrier function with $\beta = 0.5$, the robot keeps a distance from the wall and maintains its original forward walking behavior in its sagittal plane, which is similar to

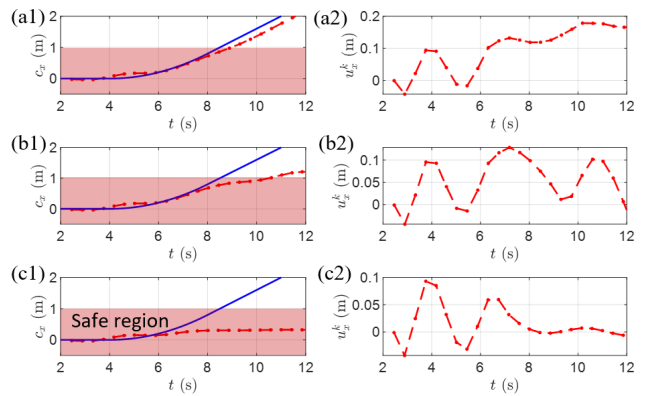


Fig. 4. *Case 1:* Trajectories of the positions (blue is the desired trajectory) and step length of the robot in the sagittal plane for (a) walking without CVaR barrier function, where (a2) shows the legacy controller input, (b1) walking with risk neutral CVaR barrier function with $\beta = 0.999$, where (b2) shows the output of the CVaR safe controller with $\beta = 0.999$, (c) walking with the risk-averse CVaR barrier function with $\beta = 0.1$, where (c2) shows the output of the CVaR safe controller with $\beta = 0.1$. The red shaded area indicates the safe region for the robot.

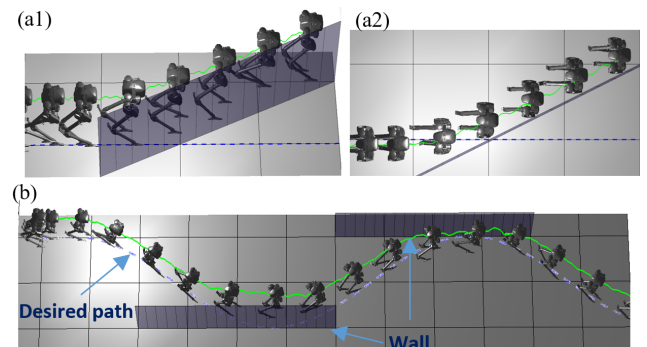


Fig. 5. The generated walking for *Case 2* (a1, a2) and *Case 3* (b).

the walking in Fig. 4 (a1). As a result, the robot also walks laterally as well to keep safe.

Case 3: We consider a scenario with multiple barrier functions. The robot is supposed to follow a sinusoidal path. We add two walls on its way. The safe set is then defined as $\min(h_1, h_2) \geq 0$, where

$$\begin{aligned} h_1(c_y) &= c_y + p_1 \geq 0, \\ h_2(c_y) &= -c_y + p_2 \geq 0, \end{aligned}$$

with $p_1 = 2$ and $p_2 = 0$. Fig. 5 (b) illustrates the walking with the CVaR barrier function $\beta = 0.5$, where the robot successfully avoided the collision with the walls.

V. CONCLUSION

We proposed a method based on CVaR barrier functions to verify and enforce CVaR safety for discrete-time stochastic systems. We proposed a computational method for synthesizing CVaR safe controllers in the case of linear dynamics. The method was applied to enforce risk-sensitive safety of a bipedal robot. Future work will extend the CVaR barrier functions to general coherent risk measures, continuous-time systems with stochastic uncertainty, and to applications involving cooperative robot-human teams and imperfect sensor measurements [41].

REFERENCES

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, Mass.: MIT Press, 2005.
- [2] M. Ono, M. Pavone, Y. Kuwata, and J. Balam, “Chance-constrained dynamic programming with application to risk-aware robotic space exploration,” *Autonomous Robots*, vol. 39, no. 4, pp. 555–571, 2015.
- [3] A. Wang, A. M. Jasour, and B. Williams, “Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty,” *IEEE Robotics and Automation Letters*, 2020.
- [4] A. Majumdar and M. Pavone, “How should a robot assess risk? towards an axiomatic theory of risk in robotics,” in *Robotics Research*. Springer, 2020, pp. 75–84.
- [5] P. Artzner, F. Delbaen, J. Eber, and D. Heath, “Coherent measures of risk,” *Mathematical finance*, vol. 9, no. 3, pp. 203–228, 1999.
- [6] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, “Risk-sensitive and robust decision-making: a cvar optimization approach,” in *Advances in Neural Information Processing Systems*, 2015, pp. 1522–1530.
- [7] Y. Chow and M. Ghavamzadeh, “Algorithms for cvar optimization in mdps,” in *Advances in neural information processing systems*, 2014, pp. 3509–3517.
- [8] L. Prashanth, “Policy gradients for cvar-constrained mdps,” in *International Conference on Algorithmic Learning Theory*. Springer, 2014, pp. 155–169.
- [9] N. Bäuerle and J. Ott, “Markov decision processes with average-value-at-risk criteria,” *Mathematical Methods of Operations Research*, vol. 74, no. 3, pp. 361–379, 2011.
- [10] S. Singh, Y. Chow, A. Majumdar, and M. Pavone, “A framework for time-consistent, risk-sensitive model predictive control: Theory and algorithms,” *IEEE Transactions on Automatic Control*, 2018.
- [11] M. P. Chapman, J. Lacotte, A. Tamar, D. Lee, K. M. Smith, V. Cheng, J. F. Fisac, S. Jha, M. Pavone, and C. J. Tomlin, “A risk-sensitive finite-time reachability approach for safety of stochastic dynamic systems,” in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2958–2963.
- [12] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [13] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, “3d dynamic walking on stepping stones with control barrier functions,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 827–834.
- [14] Y. Chen, A. Hereid, H. Peng, and J. Grizzle, “Enhancing the performance of a safe controller via supervised learning for truck lateral control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 10, 2019.
- [15] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, “Robustness of control barrier functions for safety critical control,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.
- [16] S. Kolathaya and A. D. Ames, “Input-to-state safety with control barrier functions,” *IEEE control systems letters*, vol. 3, no. 1, pp. 108–113, 2018.
- [17] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, “Safe policy synthesis in multi-agent pomdps via discrete-time barrier functions,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 4797–4803.
- [18] A. Agrawal and K. Sreenath, “Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation,” in *Robotics: Science and Systems*, 2017.
- [19] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, “Barrier functions for multiagent-pomdps with dtl specifications,” in *The 59th IEEE Conference on Decision and Control*, 2020.
- [20] A. Clark, “Control barrier functions for complete and incomplete information stochastic systems,” in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2928–2935.
- [21] C. Santoyo, M. Dutreix, and S. Coogan, “A barrier function approach to finite-time stochastic system verification and control,” *arXiv preprint arXiv:1909.05109*, 2019.
- [22] R. T. Rockafellar, S. Uryasev *et al.*, “Optimization of conditional value-at-risk,” *Journal of risk*, vol. 2, pp. 21–42, 2000.
- [23] R. T. Rockafellar and S. Uryasev, “Conditional value-at-risk for general loss distributions,” *Journal of banking & finance*, vol. 26, no. 7, pp. 1443–1471, 2002.
- [24] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.
- [25] H. Föllmer and A. Schied, *Stochastic finance: an introduction in discrete time*. Walter de Gruyter, 2011.
- [26] A. Ruszczyński, “Risk-averse dynamic programming for markov decision processes,” *Mathematical programming*, vol. 125, no. 2, pp. 235–261, 2010.
- [27] P. Glotfelter, J. Cortés, and M. Egerstedt, “Nonsmooth barrier functions with applications to multi-robot systems,” *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.
- [28] M. Ahmadi, A. Israel, and U. Topcu, “Safe controller synthesis for data-driven differential inclusions,” *IEEE Transactions on Automatic Control*, 2020.
- [29] J. T. Ott, *A Markov decision model for a surveillance application and risk-sensitive Markov decision processes*, 2010.
- [30] H. A. Le Thi, H. M. Le, T. P. Dinh *et al.*, “A de programming approach for feature selection in support vector machines learning,” *Advances in Data Analysis and Classification*, vol. 2, no. 3, pp. 259–278, 2008.
- [31] J. Thai, T. Hunter, A. K. Akametalu, C. J. Tomlin, and A. M. Bayen, “Inverse covariance estimation from data with missing values using the concave-convex procedure,” in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 5736–5742.
- [32] T. Lipp and S. Boyd, “Variations and extension of the convex–concave procedure,” *Optimization and Engineering*, vol. 17, no. 2, pp. 263–287, 2016.
- [33] X. Shen, S. Diamond, Y. Gu, and S. Boyd, “Disciplined convex-concave programming,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 1009–1014.
- [34] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.
- [35] J. Lofberg, “Yalmip: A toolbox for modeling and optimization in matlab,” in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE, 2004, pp. 284–289.
- [36] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, “Models, feedback control, and open problems of 3D bipedal robotic walking,” *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.
- [37] X. Xiong and A. D. Ames, “Dynamic and versatile humanoid walking via embedding 3d actuated slip model with hybrid lip based stepping,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6286–6293, 2020.
- [38] X. Xiong, J. Reher, and A. D. Ames, “Global position control on underactuated bipedal robots: A step-to-step dynamics approximation for step planning,” in *Review*, 2020.
- [39] X. Xiong and A. D. Ames, “Orbit Characterization, Stabilization and Composition of 3D Underactuated Bipedal Walking via Hybrid Passive Linear Inverted Pendulum Model,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [40] —, “Bipedal hopping: Reduced-order model embedding via optimization-based control,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3821–3828.
- [41] M. Ahmadi, M. Ono, M. D. Ingham, R. M. Murray, and A. D. Ames, “Risk-averse planning under uncertainty,” in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3305–3312.