

Probabilistic Guarantees for Nonlinear Safety-Critical Optimal Control

Prithvi Akella*, Wyatt Ubellacker*, and Aaron D. Ames¹

Abstract—Leveraging recent developments in black-box risk-aware verification, we provide three algorithms that generate probabilistic guarantees on (1) optimality of solutions, (2) recursive feasibility, and (3) maximum controller runtimes for general nonlinear safety-critical finite-time optimal controllers. These methods forego the usual (perhaps) restrictive assumptions required for typical theoretical guarantees, *e.g.* terminal set calculation for recursive feasibility in Nonlinear Model Predictive Control, or convexification of optimal controllers to ensure optimality. Furthermore, we show that these methods can directly be applied to hardware systems to generate controller guarantees on their respective systems.

I. INTRODUCTION

From Kalman till date, the pursuit of theoretical guarantees for optimal controllers has fascinated the controls and robotics communities alike [1]–[4]. This fascination arises as optimal controllers provide a natural way of expressing and segmenting disparate control objectives, as can be easily seen in works regarding model predictive control (MPC) [5]–[7], control barrier functions [8]–[10], and optimal path planning [11]–[13], among others. However, optimization problems becoming central to controller synthesis resulted in newer problems such as determining whether solutions exist, *e.g.* recursive feasibility in MPC, determining the efficiency with which solutions can be identified to inform control loop rates, and determining the optimality of identified solutions in non-convex optimization settings.

Recent years have seen tremendous strides in answering these questions, but areas of improvement still exist. For example, advances in Nonlinear MPC still require assumptions on the existence of control invariant terminal sets and stabilizing controllers for recursive feasibility, though identification of such items for general nonlinear systems remains a difficult problem [14]–[18]. In general, determination of solution optimality for MPC problems is equivalent to solving the Hamilton-Jacobi-Bellman equation which is known to be difficult [19]. For path-planning problems, RRT* and other, sampling-based methods are known to be probabilistically complete, *i.e.* they will produce the optimal solution given an infinite runtime, though sample-complexity results for sub-optimal solutions are few [12], [20], [21]. Finally, there are similarly few theoretical results on the time complexity of these controllers on hardware systems, as such an analysis is heavily dependent on the specific hardware.

This work was supported by the AFOSR Test and Evaluation Program, grant FA9550-19-1-0302

*Both authors contributed equally.

¹All authors are with the California Institute of Technology {pakella, wubellac, ames}@caltech.edu

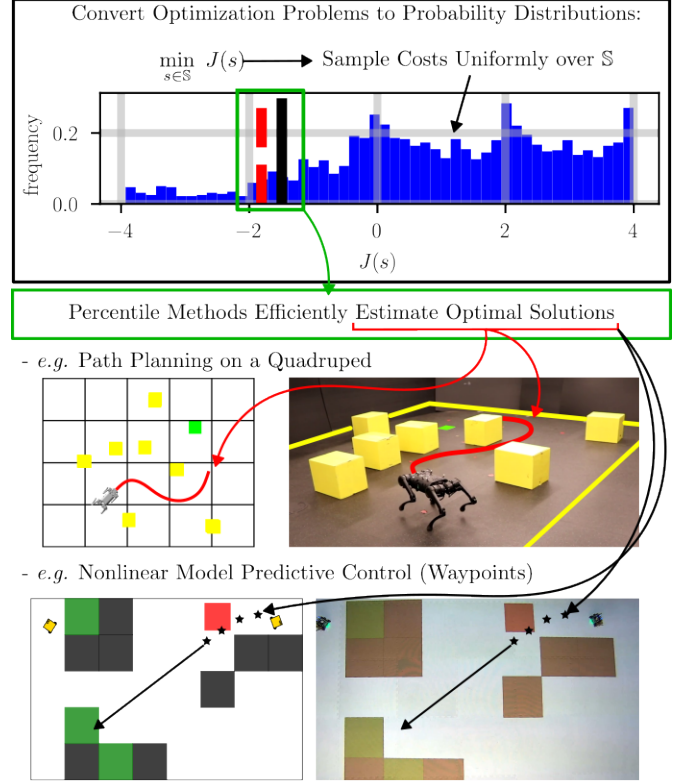


Fig. 1: Finite-time optimal controllers and their guarantees can be expressed as optimization problems. We provide probabilistic guarantees on solutions to these problems using novel results in black-box risk-aware verification.

Our Contribution: Here, the authors believe recent results in black-box risk-aware verification might prove useful in generating theoretical statements on recursive feasibility, provable sub-optimality of results, and time complexity of the associated controllers on hardware systems, without the need for restrictive assumptions. Our results are threefold.

- We provide theoretical guarantees on the provable sub-optimality of percentile-based optimization procedures [22] on producing input sequences for general, finite-time optimal control problems.
- We provide an algorithm for determining the probability with which a black-box controller is successively feasible on existing system hardware.
- We provide an algorithm to determine a probabilistic upper bound on hardware-specific controller runtimes.

Structure: To start, Section II motivates and formally states the problems under study in this paper, and the introduction to Section III provides the general theorem employed throughout. Then, Section III-A details our algorithm that

provides probabilistic guarantees on the optimality of outputted solutions to nonlinear safety-critical finite-time optimal control problems. Likewise, Section III-B details our algorithm that provides probabilistic guarantees on successive feasibility for the same type of optimal controllers. Finally, Section III-C details our algorithm that provides probabilistic guarantees on maximum controller runtimes. Lastly, we portray all our theoretical results on hardware, as described for the quadrupedal example in Section IV-A and for the Robotarium in Section IV-B [23].

II. GENERAL MOTIVATION AND PROBLEM STATEMENTS

We assume the existence of a nonlinear discrete-time system whose dynamics f are (potentially) unknown:

$$x_{k+1} = f(x_k, u_k, d), \quad x \in \mathcal{X}, \quad u \in \mathcal{U}, \quad d \in \mathcal{D}. \quad (1)$$

Here, $\mathcal{X} \subseteq \mathbb{R}^n$ is the state space, $\mathcal{U} \subseteq \mathbb{R}^m$ is the input space, and $\mathcal{D} \subseteq \mathbb{R}^p$ is the space of variable objects in our environment that we can control, *e.g.* center locations of obstacles and goals for path-planning examples, variable wind-speeds for a drone, *etc.* Provided this dynamics information, a cost J , state constraints, and input constraints, one could construct a Nonlinear Model Predictive Controller of the following form (with $j \in [0, 1, \dots, H-1]$):

$$\begin{aligned} \mathbf{u}^* = & \underset{\mathbf{u}=(u^0, u^1, \dots, u^{H-1}) \in \mathcal{U}^H}{\operatorname{argmin}} & J(\mathbf{u}, x_k, d), & \text{(NMPC)} \\ & \text{subject to} & x_k^{j+1} = f(x_k^j, u^j, d), \\ & & x_k^0 = x_k, \\ & & x_k^{j+1} \in \mathcal{X}_k^{j+1}, \\ & & u^j \in \mathcal{U}. \end{aligned}$$

For the analysis to follow, however, we note that the general NMPC problem posed in (NMPC) can be posed as the following Finite-Time Optimal Control Problem.

$$\begin{aligned} & \underset{\mathbf{u}=(u^0, u^1, \dots, u^{H-1}) \in \mathcal{U}^H}{\operatorname{argmin}} & J(\mathbf{u}, x_k, d), & \text{(FTOCP)} \\ & \text{subject to} & \mathbf{u} \in \mathbb{U}(x_k, d) \subseteq \mathcal{U}^H. \end{aligned}$$

Here, J is a bounded (perhaps) nonlinear cost function, and \mathbb{U} is a set-valued function outputting a constraint space for input sequences that (potentially) depends on the initial system and environment states (x_k, d) , respectively. Specific examples following this general form will be provided in Sections IV-A and IV-B. Finally, $H > 0$ is the horizon length for the finite-time optimal control problem. Then, the three problem statements predicated on this optimal controller (FTOCP) follow.

Problem 1. *Develop a procedure to identify input sequences \mathbf{u} that are in the $100(1 - \epsilon)\%$ -ile for some $\epsilon \in (0, 1]$ with respect to solving (FTOCP).*

Problem 2. *Develop a procedure to determine whether (FTOCP) is recursively feasible.*

Problem 3. *Develop a procedure to upper bound maximum controller runtimes for optimal controllers of the form in (FTOCP).*

III. PROBABILISTIC GUARANTEES

To make progress on the aforementioned problem statements — each will be addressed in a separate subsection to follow — we will first state a general result combining existing results on black-box risk-aware verification. To that end, consider the following optimization problem:

$$\min_{s \in \mathbb{S}} J(s), \quad (2)$$

subject to the following assumption:

Assumption 1. The decision space \mathbb{S} is a set with bounded volume, *i.e.* $\int_{\mathbb{S}} 1 \, ds = V_{\mathbb{S}} < \infty$ or \mathbb{S} has a finite number of elements. Furthermore, the cost function J is bounded over \mathbb{S} , *i.e.* $\exists m, M \in \mathbb{R}$, s. t. $m \leq J(s) \leq M$, $\forall s \in \mathbb{S}$.

This assumption permits us to define the functions \mathcal{V}, F corresponding to the volume fraction occupied by a subset A of \mathbb{S} and the set of strictly better decisions for a provided decision $s' \in \mathbb{S}$, respectively:

$$\mathcal{V}(A) = \frac{\int_A 1 \, ds}{\int_{\mathbb{S}} 1 \, ds}, \quad (3)$$

$$F(s') = \{s \in \mathbb{S} \mid J(s) < J(s')\}. \quad (4)$$

Naturally then, for a given decision $s' \in \mathbb{S}$, were $\mathcal{V}(F(s')) \leq \epsilon$ for some $\epsilon \in (0, 1]$, *i.e.* s' is such that the volume fraction of strictly better decisions is no more than ϵ , then s' would be in the $100(1 - \epsilon)\%$ -ile with respect to minimizing J . Likewise, the associated minimum cost of such a decision $J(s')$ should also be a probabilistic lower bound on achievable costs. Both of these notions are expressed formally in the theorem below, which combines similar results from [22], [24].

Theorem 1. *Let $\{(s_i, J(s_i))\}_{i=1}^N$ be a set of N decisions and costs for decisions s_i sampled via $\mathbb{U}[\mathbb{S}]$, with ζ_N^* the minimum sampled cost and s_N^* the (perhaps) non-unique decision with minimum cost. Then $\forall \epsilon \in (0, 1]$, the probability of sampling a decision whose cost is at-least ζ_N^* is at minimum $1 - \epsilon$ with confidence $1 - (1 - \epsilon)^N$, *i.e.**

$$\mathbb{P}_{\mathbb{U}[\mathbb{S}]}^N [\mathbb{P}_{\mathbb{U}[\mathbb{S}]} [J(s) \geq \zeta_N^*] \geq 1 - \epsilon] \geq 1 - (1 - \epsilon)^N. \quad (5)$$

Furthermore, $\forall \epsilon \in (0, 1]$, s_N^ is in the $100(1 - \epsilon)\%$ -ile with minimum confidence $1 - (1 - \epsilon)^N$, *i.e.**

$$\mathbb{P}_{\mathbb{U}[\mathbb{S}]}^N [\mathcal{V}(F(s_N^*)) \leq \epsilon] \geq 1 - (1 - \epsilon)^N. \quad (6)$$

Proof: This is a direct application of Theorem 7 in [22] and Theorem 2 in [24]. ■

To clarify then, this is the central result on probabilistic optimality — derived from existing results on black-box risk-aware verification — that we will exploit in the remainder of the paper to address the three aforementioned questions. Our efforts regarding the first problem statement will follow.

A. Percentile-Based Input Selection

Problem 1 references the development of an efficient method to solve (FTOCP). To that end, we aim to take a percentile method that exploits equation (6) in Theorem 1.

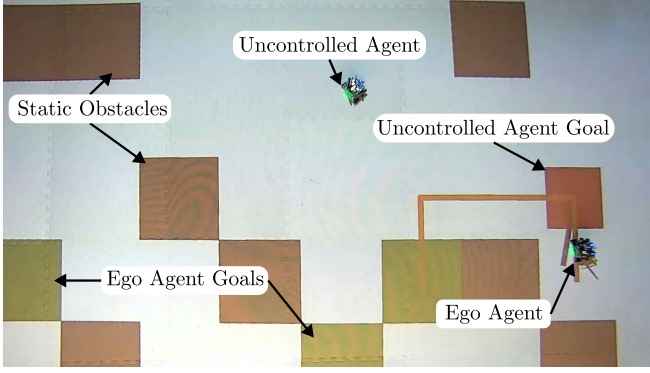


Fig. 2: Experimental setup for Robotarium reach-avoid tests.

As a result, our corollary in this vein stems directly from Theorem 1, though we will make one clarifying assumption.

Assumption 2. Let J and \mathbb{U} be as per (FTOCP), let \mathcal{V} be as per (3) with respect to the decision space $\mathbb{U}(x_k, d)$, and let F be as per (4) with respect to this cost J and $\mathbb{U}(x_k, d)$. Furthermore, let J be bounded over $\mathbb{U}(x_k, d)$, and let $\mathbb{U}(x_k, d)$ be a set of bounded volume (or finitely many elements if a discrete set) for any choice of $(x_k, d) \in \mathcal{X} \times \mathcal{D}$ (these sets defined in (1)). Finally, let $\{(\mathbf{u}_i, J(\mathbf{u}_i, x_k, d))\}_{i=1}^N$ be a set of N uniformly sampled sequences \mathbf{u}_i from $\mathbb{U}(x_k, d)$ with their corresponding costs, and let \mathbf{u}_N^* be the (potentially) non-unique sequence with minimum sampled cost.

Corollary 1. Let Assumption 2 hold and let $\epsilon \in (0, 1]$. Then, \mathbf{u}_N^* is in the $100(1 - \epsilon)\%$ -ile with respect to minimizing J at the current system and environment state (x_k, d) with minimum confidence $1 - (1 - \epsilon)^N$, i.e.,

$$\mathbb{P}_{\mathbb{U}(x_k, d)}^N [\mathcal{V}(F(\mathbf{u}_N^*)) \leq \epsilon] \geq 1 - (1 - \epsilon)^N.$$

Proof: Use equation (6) in Theorem 1. ■

In short, Corollary 1 tells us that if we have a finite-time optimal control problem of the form in (FTOCP), where for some system and environment state (x_k, d) , the cost function J is bounded over a bounded decision space $\mathbb{U}(x_k, d)$, then we can take a percentile approach to identify input sequences that are better than a large fraction of the space of all feasible input sequences. Notably, this statement is made independent of the convexity, or lack thereof, of (FTOCP), making it especially useful for non-convex MPC. Furthermore, as is done in Section IV-A to follow, one can further optimize over the outputted percentile solution \mathbf{u}_N^* via gradient descent — should gradient information be available. The resulting solution then retains the same confidence on existing within the same percentile, while also being efficient to calculate. This does introduce new questions, however. Namely, will a percentile solution always exist, and how efficient is the calculation of these sequences on a given hardware? These questions will be answered in the sections to follow.

B. Determining Recursive Feasibility

Problem 2 references the development of an algorithm to efficiently determine the recursive feasibility of (FTOCP). To ease the statement of the theoretical results to follow,

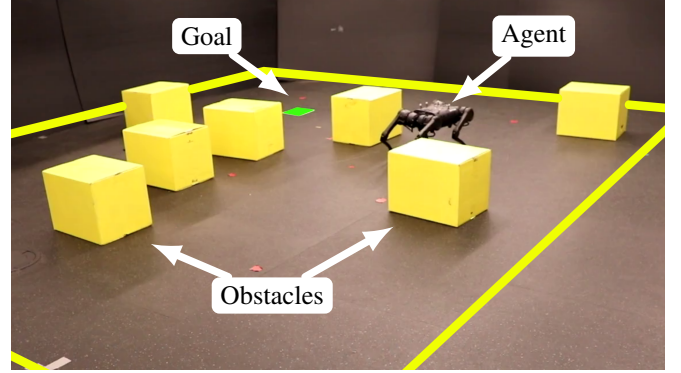


Fig. 3: Experimental setup for Quadruped reach-avoid tests.

we indicate via $|\mathbb{U}(x_k, d)|$ the “size” of the constraint space $\mathbb{U}(x_k, d)$ for (FTOCP), with $|\emptyset| = 0$. Additionally, we will assume that there exists some controller U that either utilizes the aforementioned percentile method in Section III-A or some other technique to produce (potentially approximate) solutions to (FTOCP), i.e.

$$\exists U : \mathcal{X} \times \mathcal{D} \rightarrow \mathcal{U} \text{ s.t. } U(x, d) = u \in \mathcal{U} \quad (7)$$

Furthermore, we will indicate via the following notation, the evolution of our system under this controller U , provided an initial system and environment state:

$$x^+[x, d] = f(x, U(x, d), d).$$

This allows us to formally define recursive feasibility.

Definition 1. An optimal controller of the form in (FTOCP) is *recursively feasible* if and only if for all system and environment states, the feasible space for (FTOCP) is non-empty for successive timesteps, i.e. $\forall (x, d) \in \mathcal{X} \times \mathcal{D}, |\mathbb{U}(x, d)| > 0 \implies |\mathbb{U}(x^+[x, d], d)| > 0$.

As motivated earlier, we can express recursive feasibility determination as an optimization problem. Specifically, let our cost function C be as follows:

$$\begin{aligned} \mathbb{T}(x, d) &= |\mathbb{U}(x, d)| > 0 \text{ and } |\mathbb{U}(x^+[x, d], d)| > 0, \\ C(x, d) &= \begin{cases} 1 & \text{if } \mathbb{T}(x, d) = \text{True}, \\ 0 & \text{else.} \end{cases} \end{aligned} \quad (8)$$

We can generate a minimization problem provided this cost function C over the joint state space $\mathcal{X} \times \mathcal{D}$:

$$\min_{x \in \mathcal{X}, d \in \mathcal{D}} C(x, d). \quad (9)$$

If the solution to (9) were positive, then (FTOCP) is recursively feasible. Likewise, if the solution were negative, then there exists a counterexample. As a result, not only can we express recursive feasibility determination as an optimization problem, but this problem is also of the same form as in (2), permitting a probabilistic solution approach as expressed in the following assumption and corollary.

Assumption 3. Let C be as per (8), let \mathcal{X}, \mathcal{D} be as per (1) and also be spaces of bounded volume, let $\{C(x_i, d_i)\}_{i=1}^N$ be a set of N cost evaluations of decision tuples (x_i, d_i)

sampled independently via $U[\mathcal{X} \times \mathcal{D}] \triangleq \mu$, let ζ_N^* be the minimum cost evaluation, and let $\epsilon \in [0, 1]$.

Corollary 2. *Let Assumption 3 hold. Then if $\zeta_N^* = 1$, (FTOCP) is successively feasible with minimum probability $1 - \epsilon$ and with minimum confidence $1 - (1 - \epsilon)^N$.*

Proof: Equation (5) in Theorem 1 tells us that

$$\mathbb{P}_\mu^N [\mathbb{P}_\mu [C(x, d) \geq \zeta_N^*] \geq 1 - \epsilon] \geq 1 - (1 - \epsilon)^N.$$

By definition of C in (8), if $\zeta_N^* = 1$, then with minimum probability $1 - \epsilon$, $|\mathbb{U}(x, d)| > 0 \implies |\mathbb{U}(x^+[x, d], d)| > 0$. In other words, with minimum probability $1 - \epsilon$, if (FTOCP) were feasible at the prior time step, then it will also be feasible at the next time step, *i.e.* successively feasible. ■

In other words, Corollary 2 tells us that to probabilistically determine whether a given finite-time optimal control problem is successively feasible, it is sufficient to identify at least one input in the constraint space for successive optimization problems starting at N randomly sampled state pairs (x, d) . Determining at least one such input could be achieved by querying the corresponding controller U or some other method desired by the practitioner. Notably, this does not guarantee recursive feasibility as that would correspond to the optimal value of (9) being positive. However, with arbitrarily high probability, we can provide guarantees that even hardware controllers will be successively feasible for sampled state pairs $(x, d) \in \mathcal{X} \times \mathcal{D}$, which is the underlying requirement for recursive feasibility as per Definition 1.

C. Determining Hardware-Specific Controller Runtimes

Lastly, Problem 3 references the development of an algorithm to efficiently identify maximum controller runtimes on existing system hardware. To address this from a probabilistic perspective, we will first define some notation. To start, we will use the same controller U as per equation (7). We also denote via T a timing function that outputs the evaluation time for querying the controller U at a given state pair (x, d) , *i.e.* $T : \mathcal{X} \times \mathcal{D} \rightarrow \mathbb{R}_{++}$. Then we can nominally express maximum controller runtime determination as an optimization problem:

$$\max_{x \in \mathcal{X}, d \in \mathcal{D}} T(x, d). \quad (10)$$

Under the fairness assumption that the controller does have a bounded runtime, however, identification of a probabilistic maximum runtime is solvable via probabilistic optimization procedures as outlined by Theorem 1. In a similar fashion as prior, we will state a clarifying assumption and the formal corollary statement will follow.

Assumption 4. Let T be as per (10), let \mathcal{X}, \mathcal{D} be as per (1) and be of bounded volume, let $\{T(x_i, d_i)\}_{i=1}^N$ be a set of N controller runtimes for state pairs (x_i, d_i) sampled independently via $U[\mathcal{X} \times \mathcal{D}] \triangleq \mu$, let ζ_N^* be the maximum runtime, and let $\epsilon \in [0, 1]$.

Corollary 3. *Let Assumption 4 hold. Then, the probability of sampling a state pair whose controller runtime is at most*

ζ_N^* *is at-least* $1 - \epsilon$ *with confidence* $1 - (1 - \epsilon)^N$, *i.e.*

$$\mathbb{P}_\mu^N [\mathbb{P}_\mu [T(x, d) \leq \zeta_N^*] \geq 1 - \epsilon] \geq 1 - (1 - \epsilon)^N.$$

Proof: Consider (10) expressed as a minimization. Under the same assumptions, equation (5) in Theorem 1 states that

$$\mathbb{P}_\mu^N [\mathbb{P}_\mu [-T(x, d) \geq -\zeta_N^*] \geq 1 - \epsilon] \geq 1 - (1 - \epsilon)^N,$$

and flipping the innermost inequality provides the result. ■ In short then, Corollary 3 tells us that probabilistic determination of maximum controller runtimes stems easily by recording controller runtimes for N randomly sampled scenarios identified through N randomly sampled system and environment state pairs (x, d) from $\mathcal{X} \times \mathcal{D}$.

IV. EXPERIMENTAL DEMONSTRATIONS

To demonstrate the contributions of our work, we applied the aforementioned methods to two reach-avoid navigation examples: 1) an A1 Unitree Quadruped [25] in a field of static obstacles, and 2) a Robotarium [23] scenario with the controlled agent subject to both static obstacles and an additional uncontrolled, dynamic agent.

A. Quadrupedal Walking

Reach Avoid Navigation Task: In the quadruped example, the agent is tasked to reach a specific goal location (green) while avoiding static obstacles (yellow) within a 5m by 4m space—the agent and obstacles move and can be placed continuously within this space. The set of all environments \mathcal{D} corresponds to the set of all setups, including goals, robot starting locations, and obstacles, that satisfy the aforementioned conditions while allowing for at-least one feasible path to the goal. Figure 3 depicts an example setup, with Figure 8 showing multiple examples of viable environments in \mathcal{D} .

FT-OCP formulation: We formulated quadrupedal navigation as an optimal control problem of the form in (FTOCP). We consider as states, the position of the robot within a bounded rectangle $\mathcal{X} = [0, 5] \times [0, 4]$. Individual inputs are discrete changes in position with bounded magnitude, with corresponding H -length input sequence \mathbf{u} a finite horizon of positional waypoints. Mathematically, the state-dependent subset of permissible sequences $\mathcal{U}_p^H(x)$ is as follows, with $j \in [0, 1, \dots, H - 2]$:

$$\mathcal{U}_p^H(x) = \left\{ \mathbf{u} \in \mathcal{U}^H \mid \begin{array}{l} \|u^0 - x\| \leq 0.03, \text{ and } \\ \|u^{j+1} - u^j\| \leq 0.03. \end{array} \right\}$$

$\mathbb{U}(x_k, d)$ then further constrains \mathbf{u} to remain within a feasible set of states via a discrete barrier-like condition. To define that feasible state set, for D obstacle positions let $d = [d_1^T, d_2^T, \dots, d_D^T]^T \in \mathbb{R}^{2 \times D}$. Then with a collision radius r , the feasible state set is:

$$\mathcal{F}(d) = \{x \in \mathcal{X} \mid \|x - d_j\| \geq r \ \forall j = 1, \dots, D\}.$$

Then we can define the overall constrained input space $\mathbb{U}(x, d)$ as follows, with $x^0 = x$, $x^{j+1} = f(x^j, u^j, d)$, and $\forall \ell \in 0, 1, \dots, H$:

$$\mathbb{U}(x, d) = \{\mathbf{u} \in \mathcal{U}_p^H(x) \mid x^\ell \in \mathcal{F}(d)\}. \quad (11)$$

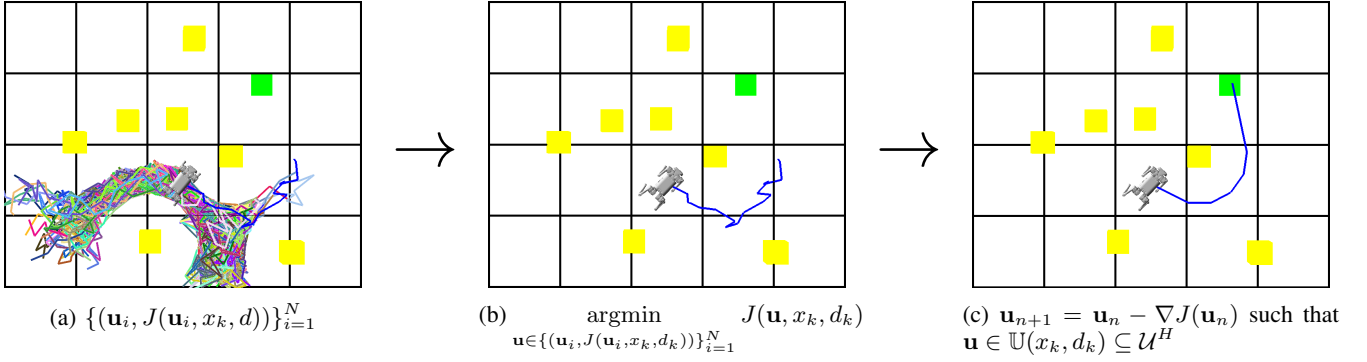


Fig. 4: Solving the FT-OCF for the quadruped reach-avoid experiment. (a) generates uniformly random feasible input sequence samples. (b) selects the best sample according to cost function $J(\mathbf{u}, x_k, d_k)$. Finally, (c) leverages the differentiability of J to further improve the choice of \mathbf{u} via constrained gradient descent.

Here, the discrete-time dynamics are simply $f(x, u, d) = x + u$. Finally, with goal state x_d , we have our cost function J as follows, again with $x^0 = x$ and $x^{j+1} = f(x^j, u^j, d)$:

$$J(\mathbf{u}, x, d) = 10\|x^H - x_d\| + \sum_{i=0}^{H-1} \|x^{i+1} - x^i\|. \quad (12)$$

This cost simultaneously rewards the final waypoint when closer to the goal and a shorter overall path length. As a result, the overall finite-time optimal control problem is:

$$\begin{aligned} \mathbf{u}^* = \underset{\mathbf{u} \in \mathcal{U}^H}{\operatorname{argmin}} \quad & J(\mathbf{u}, x_k, d) \text{ as per (12),} \\ \text{subject to} \quad & \mathbf{u} \in \mathbb{U}(x_k, d) \text{ as per (11).} \end{aligned} \quad (13)$$

Solving the FT-OCF: To solve (13), we employ the procedure described in Section III-A. We directly sample the input space \mathcal{U}^H and employ rejection sampling to generate samples $\mathbf{u} \in \mathbb{U}(x_k, d)$, until we collect 1000 such samples. From this collection of samples, we choose the minimum cost sample by evaluating $J(\mathbf{u}, x_k, d)$. This sample meets our guarantees as described in Corollary 1. However, we recognize that our cost function is differentiable in \mathbf{u} , and we can employ constrained gradient descent [26] to further improve the solution. This process is illustrated in Figure 4.

Experiments and Results: Tests were performed for both random and curated obstacle locations, with care taken to reject samples without a feasible path to the goal. The quadruped was given a random start position and orientation, and a fixed goal, x_d . (13) was solved using a Python implementation of the above procedure at ~ 1.5 Hz, taking x_k to be the position of the quadruped as measured by an Optitrack motion capture system. An IDQP-based walking controller [27] tracked the computed plan, with tangent angles along the plan used as desired quadruped heading.

By Corollary 1, choosing the best out of 1000 uniformly chosen waypoint sequences implies that the best sequence \mathbf{u}_N^* should be in the 99%-ile with 99.995% confidence. This is indeed the case as can be seen in the data portrayed at the top of Figure 6, corroborating Corollary 1. Both Corollaries 2 and 3 were also corroborated by recording successive feasibility and controller runtimes for 1000 randomized instances

of the percentile method applied to (13). In all cases, the controller was successively feasible, and the maximum controller runtime was 0.92 seconds. Comparing against another 5000 random samples affirms that the reported maximum runtime exceeded the 99%-ile cutoff, while the controller was successively feasible in all instances as well. The data for runtimes is shown on the bottom in Figure 6. Qualitatively speaking, however, the proposed procedure produces a valid, collision-free plan in all tested scenarios. This plan ultimately leads to the quadruped reaching the desired goal in many scenarios. However, some obstacle placements lead to local minima that cannot be escaped, as this is a finite-time method. Increasing the horizon H allows for success in these conditions, but requires a trade-off in execution time. These results are elucidated in the supplemental video.

B. Multi-Agent Verification

Figure 2 depicts the reach-avoid scenario for the Robotarium [23] agents which can be modeled as unicycle systems, i.e. with $x_k \in \mathcal{X}$, $u_k \in \mathcal{U}$:

$$x_{k+1} = x_k + (\Delta t = 0.033) \underbrace{\begin{bmatrix} \cos(x_k[3]) & 0 \\ \sin(x_k[3]) & 0 \\ 0 & 1 \end{bmatrix} u_k}_{f(x_k, u_k, d)}. \quad (14)$$

Here, $\mathcal{X} = [-1.6, 1.6] \times [-1.2, 1.2] \times [0, 2\pi]$ and $\mathcal{U} = [-0.2, 0.2] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$. Additionally, each agent comes equipped with a Lyapunov controller U that steers the agent to a provided waypoint $w \in \mathcal{W}$:

$$U : \mathcal{X} \times \mathcal{D} \times \mathcal{W} \triangleq [-1.6, 1.6] \times [-1.2, 1.2] \rightarrow \mathcal{U}.$$

The environment space \mathcal{D} consists of the grid locations of 8 static obstacles on an 8×5 grid overlaid on the state space \mathcal{X} , the cells of 3 goals on the same grid, the starting position in \mathcal{X} of another, un-controlled moving agent that is at-least 0.3 meters away from the ego agent of interest, and the un-controlled agent's goal cell on the same grid. No static obstacles are allowed to overlap with any of the goals, though the un-controlled agent's goal may overlap with at least one of the goals of the ego agent, and the setup of

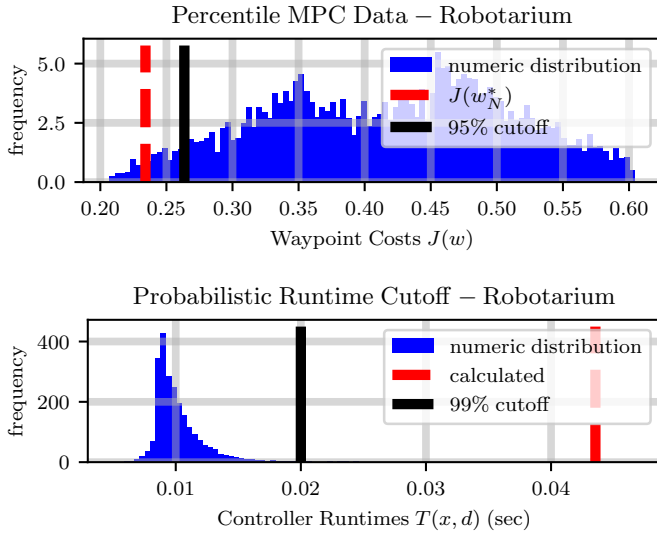


Fig. 5: Robotarium Hardware data when (top) taking a percentile method to solving (NMPC-B), and (bottom) calculating a probabilistic cutoff on maximum controller runtime. In both cases, the red lines corresponding to (top) the identified waypoint and (bottom) the reported maximum controller runtime are to the left and right, respectively, of their corresponding, true probabilistic cutoffs. In other words, the identified values satisfy their corresponding probabilistic statements, affirming Corollaries 1 and 3. Numeric distributions were calculated by evaluating 5000 random samples.

static obstacles must always allow for there to exist at least one path to one of the ego agent’s goals. Figure 7 shows multiple examples of environment setups within \mathcal{D} .

NMPC Formulation: Based on the setup of static obstacles and goal locations on the grid, we define a function $S : \mathcal{W} \rightarrow \mathbb{R}_+$ that outputs the length of the shortest feasible path to a goal from a provided planar waypoint. Should no feasible path exist from a waypoint $w \in \mathcal{W}$, $S(w) = 100$ to indicate infeasibility. Inspired by discrete control barrier function theory [28], we define a control barrier function h which accounts for both the ego agent state x_a and the un-controlled agent state x_o (with $P = [I_{2 \times 2} \quad \mathbf{0}_{2 \times 1}]$):

$$h(x_a, x_o) = \begin{cases} -5 & \text{in static obstacle cell,} \\ \|P(x_a - x_o)\| - 0.18 & \text{else.} \end{cases}$$

Then, provided $h(x_a, x_o) \geq 0$, the ego agent hasn’t crashed into a static obstacle and is maintaining at least a distance of 0.18 m from the un-controlled agent.

This permits us to define an NMPC problem as follows with the dynamics f as per (14) and $\forall j \in [1, 2, 3, 4, 5]$:

$$\begin{aligned} w_k^* &= \operatorname{argmin}_{w \in \mathcal{W}} S(w), & (\text{NMPC-A}) \\ \text{subject to} & \quad x_k^j = f(x_k^{j-1}, u^{j-1}, d), & (\text{a}) \\ & \quad x_k^0 = x_k, & (\text{b}) \\ & \quad h(x_{k,a}^j, x_o) \geq 0 & (\text{c}) \\ & \quad u^{j-1} = U(x_k^{j-1}, d, w), & (\text{d}) \\ & \quad 0.05 \leq \|w - x_k\| \leq 0.2. \end{aligned}$$

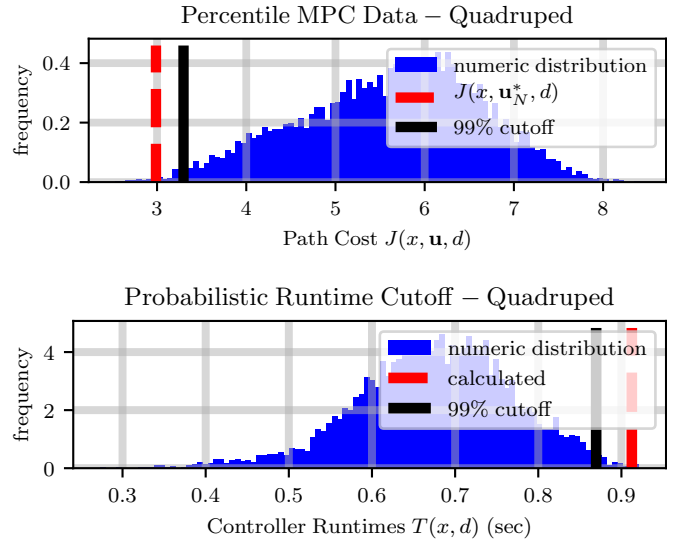


Fig. 6: Quadraped Hardware data when (top) taking a percentile method to solve (13), and (bottom) calculating a probabilistic cutoff on maximum controller runtime. In both cases, the red lines corresponding to (top) the identified path and (bottom) the reported maximum controller runtime are to the left and right, respectively, of their corresponding, true probabilistic cutoffs. This affirms Corollaries 1 and 3 insofar as the identified values satisfy their corresponding probabilistic statements. Numeric distributions were calculated by evaluating 5000 random samples.

To ease sampling then, we will consider an augmented cost J that outputs 100 whenever a waypoint w fails to satisfy constraints (a)-(d) in (NMPC-A). Then we define the NMPC problem to-be-solved as follows:

$$\begin{aligned} w_k^* &= \operatorname{argmin}_{w \in \mathcal{W}} J(w), & (\text{NMPC-B}) \\ \text{subject to} & \quad 0.05 \leq \|w - x_k\| \leq 0.2. \end{aligned}$$

Results: By Corollary 1, if we wish to take a percentile approach to determine a waypoint w_N^* in the 95%-ile with 99.4% confidence we need to evaluate $N = 100$ uniformly chosen waypoints from the constraint space for (NMPC-B). Figure 5 shows the cost of the outputted waypoint sequence compared against 5000 randomly sampled values, and as can be seen, the outputted waypoint w_N^* is indeed in the 95%-ile, confirming Corollary 1. Calculating this controller’s runtime in 460 randomly sampled initial state and environment scenarios yielded a probabilistic maximum $\zeta_N^* = 0.043$ seconds. According to Corollary 3, this maximum runtime should be an upper bound on the true, 99% cutoff on controller runtimes with confidence 99% — and as can be seen in Figure 5, ζ_N^* exceeds the true value. Finally, to corroborate Corollary 2, we evaluated the recursive feasibility cost function C as per (8) in each of the same 460 randomly sampled scenarios from prior. In each scenario, the percentile controller was successively feasible, indicating that with 99% probability the controller will be successively feasible. Evaluating the same cost for 5000 more uniformly chosen samples resulted in the controller being successively feasible

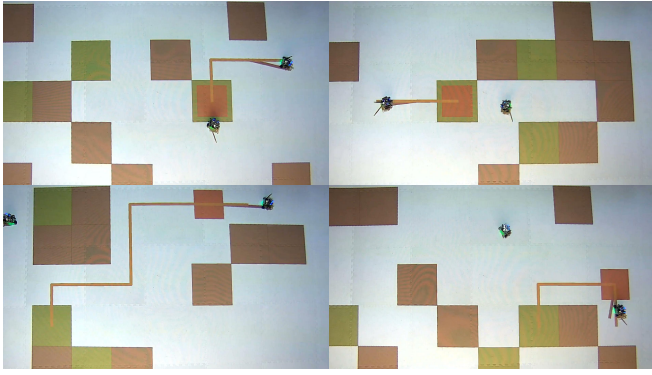


Fig. 7: Experimental depictions of the randomized environments \mathcal{D} for the Robotarium as described in Section IV-B. The black squares correspond to static obstacles, the green squares correspond to goals for the ego-agent whose shortest path from its starting cell is shown in orange, and the red squares correspond to the un-controlled agent's goal.

each time, corroborating Corollary 2.

V. CONCLUSION

Based on existing work in black-box risk-aware verification, we provided probabilistic guarantees for percentile approaches to solving finite-time optimal control problems, recursive feasibility of such approaches, and bounds on maximum controller runtimes. In future work, the authors plan to explore how the generated probabilistic guarantees can be applied in other scenarios, *e.g.* probabilistic planning procedures. Secondly, we aim to bound the optimality gap between our percentile solutions and the global optimum.

REFERENCES

- [1] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.
- [2] R. E. Kalman *et al.*, "Contributions to the theory of optimal control," *Bol. soc. mat. mexicana*, vol. 5, no. 2, pp. 102–119, 1960.
- [3] A. Locatelli and S. Sieniutycz, "Optimal control: An introduction," *Appl. Mech. Rev.*, vol. 55, no. 3, pp. B48–B49, 2002.
- [4] S. P. Sethi and S. P. Sethi, *What is optimal control theory?* Springer, 2019.
- [5] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.
- [6] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE control systems magazine*, vol. 20, no. 3, pp. 38–52, 2000.
- [7] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [8] 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.
- [9] 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.
- [10] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8352–8358.
- [11] P. Raja and S. Pugazhenthii, "Optimal path planning of mobile robots: A review," *International journal of physical sciences*, vol. 7, no. 9, pp. 1314–1320, 2012.
- [12] I. Noreen, A. Khan, and Z. Habib, "Optimal path planning using rrt* based approaches: a survey and future directions," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016.

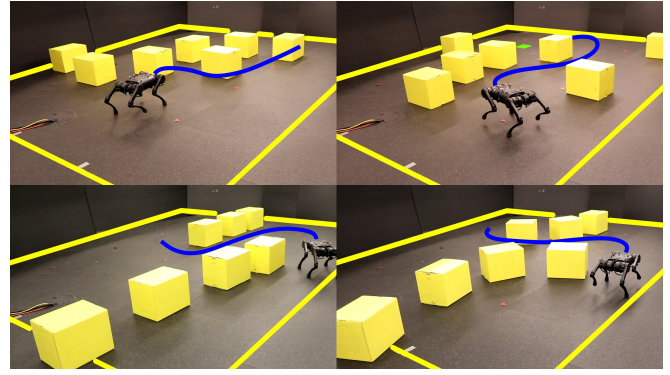


Fig. 8: Depictions of the randomized environments \mathcal{D} for the Quadruped experiments as described in Section IV-A. Yellow boxes are static obstacles, and the goal is shown in green (not visible in all images). The computed plan is depicted in blue.

- [13] B. Riviere, W. Hönig, Y. Yue, and S.-J. Chung, "Glas: Global-to-local safe autonomy synthesis for multi-robot motion planning with end-to-end learning," *IEEE robotics and automation letters*, vol. 5, no. 3, pp. 4249–4256, 2020.
- [14] M. Maiworm, T. Bähge, and R. Findeisen, "Scenario-based model predictive control: Recursive feasibility and stability," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 50–56, 2015.
- [15] W. Esterhuizen, K. Worthmann, and S. Streif, "Recursive feasibility of continuous-time model predictive control without stabilising constraints," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 265–270, 2020.
- [16] X. Fang and W.-H. Chen, "Model predictive control with preview: recursive feasibility and stability," *IEEE Control Systems Letters*, vol. 6, pp. 2647–2652, 2022.
- [17] S. Yu, X. Li, H. Chen, and F. Allgöwer, "Nonlinear model predictive control for path following problems," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 8, pp. 1168–1182, 2015.
- [18] S. Lucia, S. Subramanian, D. Limon, and S. Engell, "Stability properties of multi-stage nonlinear model predictive control," *Systems & Control Letters*, vol. 143, p. 104743, 2020.
- [19] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2004.
- [20] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *Ieee access*, vol. 2, pp. 56–77, 2014.
- [21] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1478–1483.
- [22] P. Akella, A. Dixit, M. Ahmadi, J. W. Burdick, and A. D. Ames, "Sample-based bounds for coherent risk measures: Applications to policy synthesis and verification," *arXiv preprint arXiv:2204.09833*, 2022.
- [23] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.
- [24] P. Akella, M. Ahmadi, and A. D. Ames, "A scenario approach to risk-aware safety-critical system verification," *arXiv preprint arXiv:2203.02595*, 2022.
- [25] U. Robotics. (2021) Unitree A1 Quadruped. [Online]. Available: <https://www.unitree.com/products/a1>
- [26] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, pp. 2004–2005, 2003.
- [27] W. Uellacker and A. D. Ames, "Robust locomotion on legged robots through planning on motion primitive graphs," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, accepted.
- [28] 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*, vol. 13. Cambridge, MA, USA, 2017.