An Optimistic Approach to Cost-Aware Predictive Control ⋆

Michael Enqi Cao [a], Matthieu Bloch [a], Samuel Coogan [a]

[a] *School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332*

Abstract

We consider continuous-time systems subject to a priori unknown state-dependent disturbance inputs. Given a target goal region, our first approach consists of a control scheme that avoids unsafe regions of the state space and observes the disturbance behavior until the goal is reachable with high probability. We leverage collected observations and the mixed monotonicity property of dynamical systems to efficiently obtain high-probability overapproximations of the system's reachable sets. These overapproximations improve as more observations are collected. For our second approach, we consider the problem of minimizing cost while navigating towards the goal region and modify our previous formulation to allow for the estimated confidence bounds on the disturbance to be adjusted based on what would reduce the overall cost. We explicitly consider the additional cost incurred through exploration and develop a formulation wherein the amount of exploration performed can be directly tuned. We show theoretical results confirming that this confidence bound modification strategy outperforms the previously developed strategy on a simplified system. We demonstrate the first approach on an example of a motorboat navigating a river, then showcase a Monte Carlo simulation comparison of both approaches on a planar multirotor navigating towards a goal region through an unknown wind field.

Key words: analysis of systems with uncertainties; optimal controller synthesis for systems with uncertainties; robust estimation.

1 Introduction

When the dynamics of a controlled system are not fully known, a common approach is to apply control actions to explore and observe the behavior of the system and adjust the control strategy as new information is collected. However, for systems with safety constraints that restrict allowable regions of the state space, the process of collecting observations must be designed to avoid unsafe behavior. Additionally, if there exists a separate objective that must be fulfilled, a balance must be struck between collecting observations of the unknown dynamics and progressing towards the objective.

Often, the system must optimize for some cost while in operation. For example, minimizing the magnitude of the input signal or the amount of power consumed by the system may be desirable. As collecting observations of the unknown behavior generally increases this cost, a natural strategy is to collect the minimum number of observations needed to guarantee safety and objective fulfillment, then immediately drive the system to the goal. However, as the disturbance signal is state-based, it may be that unexplored areas of the state space would have incurred a lower disturbance and a lower overall cost. Consequently, a key challenge arises in determining the optimal tradeoff between exploring the state space and exploiting the current least-cost path to the goal.

As an example, consider a planar quadrotor operating in an unknown wind field (see Figures 1-2), where the quadrotor must fly to the goal area while minimizing energy usage. Given some observations of the wind field at lower altitudes, it is possible to calculate a control strategy that arrives at the goal with high probability. Since the wind is observed to be blowing against the quadrotor at this altitude, the overall cost (i.e. the energy usage) is likely to be quite high. However, the wind at higher altitudes is less strong, and even blowing in the direction of the goal. If the quadrotor were to discover this, it could achieve lower energy usage. Exploring the windfield is, however, not without risk as the quadrotor needs to spend energy to fly higher and collect these observations. Thus, the objective is to design a control scheme

⋆ This paper was not presented at any IFAC meeting. Corresponding author M. E. Cao.

Email addresses: mcao34@gatech.edu (Michael Enqi Cao), matthieu.bloch@gatech.edu (Matthieu Bloch), sam.coogan@gatech.edu (Samuel Coogan).

that quantifies the risk and expected reward of exploration, and determines whether it is worth exploring to collect more observations.

In this work, we consider systems in continuous-time subject to state-dependent unknown components that enter the dynamics nonlinearly. We leverage the mixed monotonicity property of dynamical systems (see [11] for an overview) and utilize previous results [9] to obtain hyperrectangular overapproximations of the reachable sets of the system that hold with high probability. These overapproximations are obtained by computing a single trajectory of an appropriately constructed *embedding system* that is an ordinary differential equation with twice the dimension of the original system.

Comparing to existing approaches, we consider continuous time systems with nonlinear disturbances and we use reachability techniques that are computationally efficient and scalable, as demonstrated on a multirotor case study with six states. Moreover, this approach avoids excessive conservatism that often occurs when linearizing the dynamics and outerbounding the linearization error using the Lipschitz constant of the dynamics [1]. We also explicitly consider the goal of reaching a target region of the state-space while avoiding an unsafe region. We pose our algorithm for safe exploration and goal reaching as a continuous-time model predictive control problem.

We next formulate a control scheme that considers potentially lower-cost trajectories, and provide an analysis of a novel strategy for selecting when to pursue said trajectories. Specifically, we allow the Model Predictive Control (MPC) solver to *directly* adjust the probabilistic bounds on the disturbance so that it can consider lower-cost potential trajectories. We offer theoretical results that show the advantages of this novel strategy on a simplified system, and empirical results demonstrating performance on a more complex system. Complete results for general nonlinear systems are left as future work.

Parts of this work have previously appeared in conference proceedings [10], which focused on developing the base MPC scheme outlined in Section 6. We expand upon this work by providing a formal proof of the safety of the MPC scheme as well as an additional case study. Moreover, a major focus of the present work is to modify the control scheme to allow for speculation on lower-cost trajectories, which was not considered in our prior work, supported with a theoretical analysis and numerical studies.

The rest of this paper is organized as follows: in Section 2, we cover related work, before formally providing useful notation in Section 3 and defining our problem in Section 4. We then provide a brief overview of the tools we use in Section 5, before detailing the resulting control formulation in Section 6. We then outline the modifications made to enable speculation on lower-cost trajectories in Section 7, and perform an experimental analysis of each in Section 8. We conclude the paper in Section 9.

2 Related Work

Our work is most closely related to [18], which presents a discrete time MPC formulation that provides high probability safety guarantees in the presence of uncertain dynamics. The paper [18] also uses Gaussian Processes (GPs) to estimate the unknown dynamics, and then high probability ellipsoidal overapproximations of reachable sets are computed by combining these estimations with a linearization of the dynamics, where the error is bounded using Lipschitz constants. We draw from the problem setup proposed in [18] and consider a nonlinear dynamical system whose dynamics are not fully known. As in [18], we estimate the unknown component using Gaussian Process (GP) regression. Exploration of the state space is allowed so long as a feasible return trajectory is available that returns the system to a known safe set.

Similarly, [2] provides safety guarantees on reinforcement learning for robotic applications by learning the system's unknown dynamics using GPs, then employing Hamilton-Jacobi-Isaacs (HJI) reachability analysis to iteratively update the safety set of the system. The authors in [5] also provide safety guarantees (defined in terms of stability guarantees) on model-based reinforcement learning using Lyapunov-based stability verification. Additionally, [20] proposes an exploration/exploitation reachability-based control framework utilizing Bayesian meta-learning to learn the entirety of the dynamical model, while [21] uses Lipschitz interpolation to calculate reachable sets towards the same end. Alternatively, [30] proposes a Bayesian MPC algorithm wherein the model predictive controller optimizes directly on the parametric model derived from collected samples to enable the exploration/exploitation tradeoff, though safety constraints are not explicitly considered.

The paper [7] presents an adaptive MPC framework under state-dependent uncertainty. Safety is guaranteed by approximating the graph of the uncertainty via envelopes defined by quadratic constraints. A set of convex optimization problems is solved to guarantee robust constraint satisfaction for all possible values of system uncertainty. However, a key assumption of [7] is that the uncertainty is additive and globally Lipschitz with a known Lipschitz constant.

The work [5] presents a learning algorithm that explicitly considers safety defined in terms of Lyapunov stability guarantees, [13] proposes a general safety framework based on Hamilton-Jacobi reachability methods, [12, 15, 16] synthesize control barrier functions online to guarantee safety, and [17] achieves safety by estimating

the Lipschitz constant of the disturbance. Other works, such as [3, 8, 27], explore learning and updating safety sets in an online manner. For MPC-based approaches, proposed frameworks are robust to uncertainty by, e.g., assuming a known Lipschitz constant [7], assuming the uncertainty is parametric [19], or applying MPC to iterative learning control [25].

In the realm of optimizing control strategies in the face of uncertainty, [26] presents a gradient descent algorithm that simultaneously learns and optimizes for the partially unknown dynamics of a discrete-time system, [6] derives a set of sampling point selection strategies that result in data-efficient learning of an unknown Gaussian Process system, and [29] leverages a Neural Control Contraction Metric to ensure safety of a system while exploring and observing state-dependent uncertainties.

In our previous work [9], we derive high probability bounds on the unknown disturbance behavior by modeling it as a GP. In turn, these bounds enable us to calculate overapproximations of the reachable sets of the system that hold with high probability. We then use these reachable set overapproximations in an MPC formulation to compute a control strategy that is safe with high probability [10]. We accomplish this by requiring that the reachable set overapproximations never intersect the unsafe regions of the state space. This results in a control strategy that is not only safe with high probability, but is also able to ensure fulfillment of objectives with high probability.

Our approach differs from these existing works in several key aspects. Specifically, we leverage mixed monotonicity from nonlinear systems theory to calculate the high probability reachable set overapproximations. This allows us to work in continuous time, incorporate the uncertainty nonlinearly, avoid excess conservatism that results from linearizing the dynamics and outerbounding the linearization error, and provide tighter bounds than outerbounding the uncertainty through Lipschitz constants. Additionally, this enables our method to scale to systems of higher dimension, something which has traditionally been a challenge for other reachability-based techniques, though this is an active area of research as shown in [14, 22] where warm-starting is employed to speed up computation times in HJI-based methods.

The outlined strategy also bears a resemblance to traditional closed-loop MPC, but there are a few key distinctions. Chiefly, the computed strategy is still open-loop. Adjusting the confidence bounds in the described manner acts as a proxy to closing the loop around the disturbance input, as it is implicit in this strategy that an observation of the disturbance behavior will be collected and the control actions will be recomputed at the next time step. As a result, the strategy gains some of the forward-looking benefits of closed-loop MPC while avoiding the need for a computationally complex dynamic programming solution.

The main novelty of this work is that we explicitly consider the tradeoff between safety and performance. Most of the existing literature focuses on providing safety guarantees during the learning process, assuming that the system has enough time and resources to reach the optimal policy. By contrast, we present a formulation wherein the desired probability of safety is a tunable parameter, allowing for the balance between safety and performance, given limited time and resources, to be fully customizable.

3 Notation

Let (x, y) denote the vector concatenation of $x, y \in \mathbb{R}^n$, i.e., $(x, y) := [x^T \ y^T]^T \in \mathbb{R}^{2n}$. Additionally, $\preceq$ denotes the componentwise vector order, i.e., $x \preceq y$ if and only if $x_i \leq y_i$ for all $i \in \{1, ..., n\}$ where vector components are indexed via subscript.

Given $x, y \in \mathbb{R}^n$ such that $x \preceq y$, we denote the hyperrectangle defined by the endpoints x and y using the notation $[x, y] := \{z \in \mathbb{R}^n \mid x \preceq z \text{ and } z \preceq y\}$. Also, given $a = (x, y) \in \mathbb{R}^{2n}$ with $x \preceq y$, $[\![a]\!]$ denotes the hyperrectangle formed by the first and last n components of a, i.e., $[\![a]\!] := [x, y]$. Finally, let $\preceq_{\text{SE}}$ denote the *southeast order* on $\mathbb{R}^{2n}$ defined by $(x, x') \preceq_{\text{SE}} (y, y')$ if and only if $x \preceq y$ and $y' \preceq x'$. In particular, observe that when $x \preceq x'$ and $y \preceq y'$,

$$(x, x') \preceq_{\text{SE}} (y, y') \iff [y, y'] \subseteq [x, x']. \tag{1}$$

4 Problem Setup

We consider the continuous-time nonlinear dynamical system

$$\dot{x} = f(x, u, w) \tag{2}$$

with f continuously differentiable where $x \in \mathbb{R}^n$ is the system state, $u \in \mathcal{U} \subset \mathbb{R}^m$ is the input constrained to take values in $\mathcal{U}$, and $w \in \mathbb{R}^p$ is an unknown, state-dependent component of the dynamics so that $w_i = g_i(x)$ where g_i is unknown. Throughout, we assume the input constraint set has the form $\mathcal{U} = [\underline{u}, \overline{u}]$ for some $\underline{u}, \overline{u} \in \mathbb{R}^m, \underline{u} \preceq \overline{u}$, that is, $\mathcal{U}$ is a hyperrectangle defined by corners $\underline{u}$ and $\overline{u}$.

Given a feedback control strategy $u = \pi(t, x)$, we denote by $\phi(t, x_0, \pi)$ the resulting true closed-loop state trajectory of (2) when $w = g(x)$ and the system is initialized at x_0 at time 0. If π is time-invariant, we write $\pi(x)$ instead. Additionally, given some $X_0 \subseteq \mathbb{R}^n$, the *$T$-horizon reachable set from X_0* for (2) is the set of states reachable over the time horizon T from any initial condition

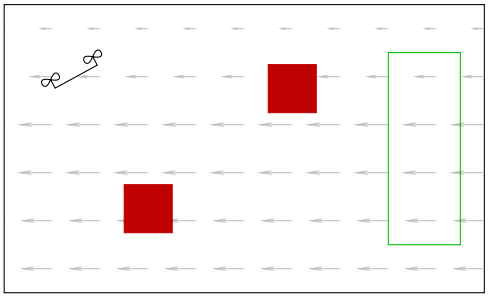

Fig. 1. An illustrative example system that fits the problem setting. A planar multirotor must fly to the goal region (green) while avoiding obstacles in midair (red). There is also a wind force acting on the multirotor which varies based on its location and is unknown a priori. Observations of this force can be collected, and the objective is to guarantee a safe path to the goal, potentially while minimizing the energy spent.

$x_0 \in X_0$ and is denoted

$$R(T, X_0, \pi) = \{\phi(T, x_0, \pi) \mid x_0 \in X_0\}. \tag{3}$$

Our objective is to steer the system to a goal region with minimal cost while avoiding any unsafe regions. For example, given a planar multirotor operating in a wind field as in Figure 1, the objective is to avoid crashing into the mid-air obstacles while trying to reach the other end of the state space, while potentially minimizing the amount of power used. We formalize this objective in the following problems and assumption.

Problem 1. Consider a system as in (2) with specified initial condition $x_0 \in \mathcal{X}_{\text{safe}}$ and input constraints $\mathcal{U}$. Given a goal region $\mathcal{X}_{\text{goal}} \subset \mathbb{R}^n$, the objective is to compute a feedback control strategy $u = \pi(t, x)$ that reaches the goal while avoiding the unsafe region $\mathcal{X}_{\text{unsafe}}$, *i.e.*,

$$\forall t \geq 0, \phi(t, x_0, \pi) \in (\mathcal{X}_{\text{unsafe}})^{\complement} \tag{4}$$
$$\exists T > 0 \text{ s.t. } \phi(T, x_0, \pi) \in \mathcal{X}_{\text{goal}}. \tag{5}$$

For this problem, we make the following assumption.

Assumption 1. There exists a known subset of the state space $\mathcal{X}_{\text{unsafe}} \subset \mathbb{R}^n$ that must be avoided. Additionally, there exists a known safe set $\mathcal{X}_{\text{safe}} \subset \mathbb{R}^n$ and corresponding time-invariant safety controller π_{safe} with $\pi_{\text{safe}}(x) \in \mathcal{U}$ for all $x \in \mathbb{R}^n$ such that, if the system is initialized in $\mathcal{X}_{\text{safe}}$, it avoids $\mathcal{X}_{\text{unsafe}}$, i.e.

$$\phi(t, x_0, \pi_{\text{safe}}) \in (\mathcal{X}_{\text{unsafe}})^{\complement} \quad \forall t \geq 0, \quad \forall x_0 \in \mathcal{X}_{\text{safe}}. \tag{6}$$

In general, we are interested in scenarios in which $\mathcal{X}_{\text{goal}}$ does not intersect $\mathcal{X}_{\text{safe}}$ so that we cannot achieve our objective by remaining within $\mathcal{X}_{\text{safe}}$. Thus, while the safety controller π_{safe} achieves (4), it generally will not achieve (5). Consequently, a separate control strategy that can navigate the unknown disturbance behavior is required to drive the system into the goal region.

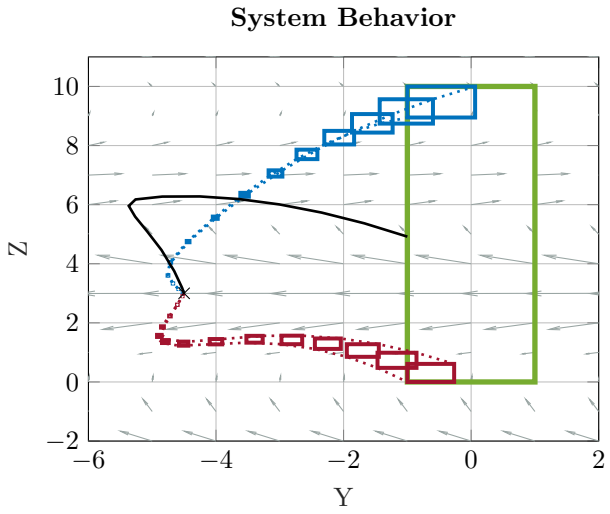

Fig. 2. For systems in which there is a cost to be minimized (i.e., energy consumption), such as the planar multirotor operating in an unknown wind field shown above, the disturbance behavior in unexplored areas of the state space may incur a lower overall cost. As shown, the system tries to reach the goal region (green rectangle) while minimizing the energy spent. The multirotor only has a few observations of the wind around its starting location, thus considering the worst-case behavior of the disturbance (Pessimism) results in the red trajectory. However, allowing the multirotor to adjust the estimated worst-case bounds (Optimism) allows the multirotor to explore, resulting in the blue trajectory. In this case, it is advantageous to be Optimistic, as the blue trajectory is closer to the calculated optimal trajectory in black. These trajectories were generated from an execution of the second case study in Section 8.

Our proposed control approach is to incrementally make progress towards the goal while learning the unknown component of the dynamics and ensuring the system is always able to safely return to $\mathcal{X}_{\text{safe}}$ if needed, until it can be guaranteed that the system can safely reach the goal. The safe return and path to the goal are ensured via a nonlinear MPC scheme that directly optimizes for an open-loop control input strategy in both cases and incorporates uncertainty from the unknown component $g(x)$ to produce probabilistically safe reachable sets of the dynamics via the mixed monotonicity property of dynamical systems. While moving towards the goal, the system is able to collect information about its dynamics and reduce the uncertainty in its estimate of $g(x)$, allowing it more freedom to safely explore.

We also consider that the system must often also optimize for some cost $J(x, u)$ (e.g. fuel consumption) while in operation. Thus, a strategy must be developed to de-

termine when exploration, which generally increases this cost, is worth the potential long-term gains while fulfilling the objective.

Problem 2. Consider a system as in (2) with specified initial condition x_0 and input constraints $\mathcal{U}$. Given a goal region $\mathcal{X}_{\text{goal}} \subset \mathbb{R}^n$, the objective is to compute a feedback control strategy $u = \pi(t, x)$ that reaches the goal with lower expected cost than the nominal strategy that solves Problem 1.

This problem setup assumes that it is possible for exploration to lower the incurred cost; thus, we only consider scenarios in which the unknown disturbance behavior has at least an indirect effect on the overall cost. Additionally, as we are only considering cost incurred, we no longer assume the existence of unsafe sets in this problem, which also removes the need for a safety set.

An example of this setup can be found in Figure 2, wherein a planar multirotor is attempting to fly to the goal with minimal energy expenditure. The nominal strategy that solves Problem 1 produces the red trajectory, whereas the trajectory that would be optimal if wind disturbance were exactly known is in black. The objective of Problem 2 is, given current knowledge of the disturbance behavior, to produce strategies akin to the blue trajectory, which is closer to the optimal trajectory than the nominal red trajectory. We return to this planar multirotor system with a case study in Section 8.

5 High Probability Reachable Sets

In this section, we provide an overview of mixed monotonicity and how it enables efficient calculation of reachable set overapproximations. We then illustrate how the introduction of Gaussian Processes (GPs) leads to overapproximations of reachable sets that hold with high probability. In subsequent sections, we will use these reachable sets to formulate our proposed solutions to Problems 1 and 2.

5.1 Mixed Monotonicity

The system (2) is *mixed monotone with respect to a decomposition function* δ if δ satisfies the following:

(1) For all x and all w, $\delta(x, u, w, x, w) = f(x, u, w)$;
(2) For all i, $j \in \{1, \cdots, n\}$, $i \neq j$, $\frac{\partial \delta_i}{\partial x_j}(x, u, w, \widehat{x}, \widehat{w}) \geq 0$ for all $x, \widehat{x}, u, w, \widehat{w}$;
(3) For all i, $j \in \{1, \cdots, n\}$, $\frac{\partial \delta_i}{\partial \widehat{x}_j}(x, u, w, \widehat{x}, \widehat{w}) \leq 0$ for all $x, \widehat{x}, u, w, \widehat{w}$;
(4) For all $i \in \{1, \cdots, n\}$ and all $k \in \{1, \cdots, p\}$, $\frac{\partial \delta_i}{\partial w_k}(x, u, w, \widehat{x}, \widehat{w}) \geq 0$ and $\frac{\partial \delta_i}{\partial \widehat{w}_k}(x, u, w, \widehat{x}, \widehat{w}) \leq 0$ for all $x, \widehat{x}, u, w, \widehat{w}$.

For any system (2), it is known that there always exists a decomposition function δ satisfying the above conditions, although one may not be readily available in closed form [1]. In general, finding a decomposition function, especially one that produces tight overapproximations, is problem-specific, although automated techniques exist for computing some classes of decomposition functions. In the case studies of Section 8, we demonstrate how a decomposition function is obtained in closed form for particular systems.

From a decomposition function, we then construct an *embedding system* with state $(x, \widehat{x}) \in \mathbb{R}^n \times \mathbb{R}^n$, input $u \in \mathbb{R}^m$, and disturbance $(w, \widehat{w}) \in \mathbb{R}^p \times \mathbb{R}^p$ as

$$\begin{bmatrix} \dot{x} \\ \dot{\widehat{x}} \end{bmatrix} = \varepsilon(x, u, w, \widehat{x}, \widehat{w}) := \begin{bmatrix} \delta(x, u, w, \widehat{x}, \widehat{w}) \\ \delta(\widehat{x}, u, \widehat{w}, x, w) \end{bmatrix}. \quad (7)$$

Denote the state of (7) at time t when initialized at $(\underline{x}_0, \overline{x}_0)$ under some input signal $u : [0, \infty) \to \mathbb{R}^m$, and disturbance signal $(w, \widehat{w}) : [0, \infty) \to \mathbb{R}^p \times \mathbb{R}^p$ by $\Phi^{\varepsilon}(t; (\underline{x}_0, \overline{x}_0), u, (w, \widehat{w}))$. The fundamental result of mixed monotone systems theory is that (7) is a monotone control system as defined in [4] with respect to the southeast order on state and disturbance; that is, given $a, a' \in \mathbb{R}^n \times \mathbb{R}^n$, $b : [0, \infty) \to \mathbb{R}^m$ and $c, c' : [0, \infty) \to \mathbb{R}^p \times \mathbb{R}^p$ such that $a \preceq_{\text{SE}} a'$ and $c(t) \preceq_{\text{SE}} c'(t)$ for all $t \geq 0$, then for all $t \geq 0$,

$$\Phi^{\varepsilon}(t; a, b, c) \preceq_{\text{SE}} \Phi^{\varepsilon}(t; a', b, c'). \quad (8)$$

An important implication of this result is that, provided that the system is initialized within $X_0 \subseteq [\underline{x}_0, \overline{x}_0]$, and the disturbance signal is overapproximated by $[w, \widehat{w}]$, then the hyperrectangle defined by $[\![\Phi^{\varepsilon}(t; (\underline{x}_0, \overline{x}_0), u, (w, \widehat{w}))]\!]$ overapproximates the true reachable set of (2), *i.e.*

$$R(T, X_0, u) \subseteq \widehat{R}(T, X_0, u) := [\![\Phi^{\varepsilon}(T; (\underline{x}_0, \overline{x}_0), u, (w, \widehat{w}))]\!]. \quad (9)$$

5.2 Gaussian Processes and High Probability Reachable Sets

If there exist known bounding functions $\underline{\gamma}_i(x, \widehat{x})$ and $\overline{\gamma}_i(x, \widehat{x})$, $\underline{\gamma}_i, \overline{\gamma}_i : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, for all $i \in \{1, \ldots, p\}$ such that

$$\underline{\gamma}_i(\underline{x}, \overline{x}) \leq g_i(x) \leq \overline{\gamma}_i(\underline{x}, \overline{x}) \ \forall x \in [\underline{x}, \overline{x}] \quad (10)$$

for all $\underline{x}, \overline{x} \in \mathbb{R}^n$ with $\underline{x} \preceq \overline{x}$, then these functions can be inserted into the previously described embedding system to produce valid reachable set overapproximations. This is achieved by taking the embedding system (7) and inserting $\underline{\gamma}(x, \widehat{x}), \overline{\gamma}(x, \widehat{x})$ in place of $w, \widehat{w}$ to produce a new

embedding system as follows:

$$\begin{bmatrix}\dot{x}\\ \dot{\widehat{x}}\end{bmatrix} = e(x,\widehat{x},u) := \begin{bmatrix}\delta(x,u,\underline{\gamma}(x,\widehat{x}),\widehat{x},\overline{\gamma}(x,\widehat{x}))\\ \delta(\widehat{x},u,\overline{\gamma}(x,\widehat{x}),x,\underline{\gamma}(x,\widehat{x}))\end{bmatrix}. \quad (11)$$

Modeling the unknown functions $g_i(x)$ as GPs enables us to formulate bounding functions that fulfill (10) with probability $1-\eta, \eta \in (0,1]$. Given noisy observations $\{y_j\}_{j=1}^t$ of the GP at corresponding points $\{x_j\}_{j=1}^t$, the surrogate functions of interest to approximate g_i are

$$\forall i \in \{1,\cdots,p\} \quad \begin{cases}\overline{g}_i^{(t)}(x) := \mu_t(x) + \sqrt{\beta_t}\sigma_t(x)\\ \underline{g}_i^{(t)}(x) := \mu_t(x) - \sqrt{\beta_t}\sigma_t(x)\end{cases} \quad (12)$$

where β_t is defined as in [9, Theorem 7] to meet the probability constraint $1-\eta$, $\mu_t(\cdot)$ is the posterior mean, and $\sigma_t(\cdot)$ is the posterior variance, computed according to the standard GP updates [24]:

$$\mu_t(x) := k_t(x)^T(K_t+\sigma^2 I)^{-1}y \quad (13)$$
$$k_t(x,x') := k(x,x') - k_t(x)^T(K_t+\sigma^2 I)^{-1}k_t(x') \quad (14)$$
$$\sigma_t^2(x) := k_t(x,x) \quad (15)$$

where $k_t(x) := (k(x_1,x),\cdots,k(x_t,x))$ and $K_t = [k_t(x_i,x_j)]$. We make the mild technical assumption [28] that the states x are confined to a compact subset $\mathcal{D} \subset \mathbb{R}^n$ included in a hypercube of edge size r, and that there exist constants $a, b > 0$ such that

$$\forall i \in \{1,\cdots,n\} \forall j \in \{1,\cdots,p\} \quad \Pr\left(\sup_{x\in\mathcal{D}}\left|\frac{\partial g_j}{\partial x_i}\right| > L\right) \le ae^{-L^2/b^2}. \quad (16)$$

Then, by adapting the proof of [28, Theorem 2], we pick $\eta \in (0,1)$ and set

$$\beta_t := 2\log\left(\frac{pt^2\pi^2}{3\eta}\right) + 2n\log\left(t^2nbr\sqrt{\log\left(\frac{2pna}{\eta}\right)}\right). \quad (17)$$

At every step t of the GP update, define a uniform discretization $\mathcal{D}_t$ of the hypercube containing $\mathcal{D}$ with size τ_t^n where $\tau_t := nt^2br\sqrt{\log\left(\frac{2npa}{\eta}\right)}$. For every $x \in \mathcal{D}$, define

$$x^{(t,-)} := \sup\{y \in \mathcal{D}_t \mid y \preceq x\}, \quad (18)$$
$$x^{(t,+)} := \inf\{y \in \mathcal{D}_t \mid x \preceq y\}. \quad (19)$$

Finally, for all $i \in \{1,\cdots,p\}$ and all $t \ge 1$, we define $\forall \underline{x} \preceq \overline{x}$,

$$\underline{\gamma}_i^{(t)}(\underline{x},\overline{x}) := \min_{x\in[\underline{x}^{(t,-)},\overline{x}^{(t,+)}]\cap\mathcal{D}_t} \underline{g}_i^{(t)}(x) - \frac{1}{t^2}, \quad (20)$$

$$\overline{\gamma}_i^{(t)}(\underline{x},\overline{x}) := \max_{x\in[\underline{x}^{(t,-)},\overline{x}^{(t,+)}]\cap\mathcal{D}_t} \overline{g}_i^{(t)}(x) + \frac{1}{t^2}. \quad (21)$$

These functions fulfill (10) for all $\underline{x}, \overline{x}$ with probability at least $1-\eta$, and thus given $\Phi^e(T;(\underline{x}_0,\overline{x}_0),u)$, which is the resulting state trajectory of (11), it holds that

$$P(R(T,X_0,u) \subseteq \widehat{R}(T,X_0,u) := [\![\Phi^e(T;(\underline{x}_0,\overline{x}_0),u)]\!]) \ge 1-\eta. \quad (22)$$

In other words, the hyperrectangular sets calculated by the embedding system (11) using these functions overapproximate the true reachable set of the system with probability at least $1-\eta$.

6 A Safe Control Algorithm

We now outline our first main contribution in developing a control scheme that is safe with high probability by leveraging the mixed monotonicity property of dynamical systems to calculate high-probability reachable sets. We insert these reachable sets into an MPC formulation, which allows for the controller to tune the amount of exploration outside the safe set while ensuring a feasible return to the safe set through the terminal set constraint. Thus, we obtain a formulation that solves for a control strategy that always has a path back to safety. We first prove that this formulation indeed satisfies Problem 1. While safe, this strategy is conservative and might incur high cost, as elaborated below. We therefore consider this nominal safe strategy to be a *pessimistic* solution to Problem 2. In subsequent sections, we modify this formulation to form the proposed optimistic strategy.

To create a numerically tractable algorithm, we consider sampled control inputs with control update timestep h such that at each step $k = t/h$, the resulting sampled embedding system dynamics are

$$\begin{bmatrix}x[k+1]\\ \widehat{x}[k+1]\end{bmatrix} = \Phi^e(h;(x[k],\widehat{x}[k]),\pi_k) \quad (23)$$

where π_k is the controller applied from time kh to $(k+1)h$, and the disturbance bound functions $\underline{\gamma},\overline{\gamma}$ are resolved by sampling over the state hyperrectangle and taking the relevant min or max. Below, we assume π_k is a zero-order hold policy so that $\pi_k(t,x) \equiv u_k$ for some $u_k \in \mathcal{U}$ to be designed by an MPC scheme. Thus, taking $\widehat{R}_k = [x[k],\widehat{x}[k]]$ overapproximates the reachable set

of (2) at time $t = kh$ with high probability (i.e. with probability at least $1 - \eta$).

We then use these overapproximations and formulate the following MPC scheme that satisfies the safety condition (4):

$$\underset{\Pi=\{u_0,\ldots,u_D\}}{\text{minimize}} \quad J_{k,\text{obj}}(\widehat{R}_0, ..., \widehat{R}_D) \tag{24}$$

subject to:

$$\begin{aligned}
&(23),\ x[0] = \widehat{x}[0] \text{ given}, u_d \in \mathcal{U} && \forall d \in \{0, \ldots, D-1\} \\
&\widehat{R}_d = [x[d], \widehat{x}[d]],\ \widehat{R}_D \subset \mathcal{X}_{\text{obj}} && \forall d \in \{0, \ldots, D\} \\
&[x(t), \widehat{x}(t)] \subset (\mathcal{X}_{\text{unsafe}})^{\complement} && \forall t \in [0, T]
\end{aligned}$$

where $\text{obj} \in \{\text{goal}, \text{safe}\}$.

The control strategy incorporating the above MPC scheme is outlined in Algorithm 1. This strategy optimizes for desired behavior based on the cost functions $J_{k,\text{goal}}$ and $J_{k,\text{safe}}$ that are designed to prioritize goal-reaching and exploration, respectively. If (24) is feasible when obj = goal, then Π contains control inputs for each timestep that altogether are guaranteed with high probability to drive the system into $\mathcal{X}_{\text{goal}}$ while avoiding $\mathcal{X}_{\text{unsafe}}$. Thus, the entire resulting control strategy Π is executed immediately and the algorithm terminates.

Otherwise, the algorithm attempts to solve (24) with obj = safe. If this MPC problem is feasible, Π contains a set of control inputs that explores the state space while guaranteeing with high probability that the system will avoid $\mathcal{X}_{\text{unsafe}}$ and return to $\mathcal{X}_{\text{safe}}$. Thus, the algorithm saves the entire strategy as Π_k. If the problem is not feasible, the algorithm copies the unexecuted actions from the previous saved strategy Π_{k-1} and appends the safety action π_{safe}. The previously saved strategy Π_{k-1} must either end in $\mathcal{X}_{\text{safe}}$ or be the result of applying π_{safe} for all time after starting in $\mathcal{X}_{\text{safe}}$, thus Π_k is guaranteed to be safe with high probability. The algorithm then executes the first action saved in Π_k, and restarts at trying to solve (24) with obj = goal.

The system may be initially unable to reach the goal, as high uncertainty on the bounds of the unknown behavior may prevent the final reachable set $\widehat{R}_K$ from being contained in $\mathcal{X}_{\text{goal}}$. However, as observations are collected and the bounds $\underline{\gamma}, \overline{\gamma}$ tighten, the reachable set overapproximations also tighten, allowing for finer control over the system and thus allowing for exploration further outside of $\mathcal{X}_{\text{safe}}$ or enabling the system to reach $\mathcal{X}_{\text{goal}}$.

Finally, we note that as we are discretizing a continuous-time system, it is possible for the system to enter $\mathcal{X}_{\text{unsafe}}$ between timesteps, and thus violate the safety condition requiring $[x(t), \widehat{x}(t)] \subset (\mathcal{X}_{\text{unsafe}})^{\complement}$ for all $t \in [0, T]$ in (24). Thus, in practice, we also check this condition at a large number of time instances between the sampling times. Moving forward, we assume sufficient sampling to prevent violation between timesteps, and we note the number of instances sampled where relevant.

Algorithm 1 Resulting Control Scheme

Data: Safety controller π_{safe}, embedding system (11) sampled as (23), bounding functions $\underline{\gamma}, \overline{\gamma}$
$\Pi_0 \leftarrow \{\pi_{\text{safe}}, ..., \pi_{\text{safe}}\}$;
for $k = 0, 1, ...$ **do**
  $(\textit{feasible}, \Pi) \leftarrow$ solve MPC (24), obj = goal;
  if feasible **then**
    apply $u = \Pi$ to system;
    break;
  $(\textit{feasible}, \Pi) \leftarrow$ solve MPC (24), obj = safe;
  if feasible **then**
    $\Pi_k \leftarrow \Pi$;
  else
    $\Pi_k \leftarrow \{\Pi_{k-1,1:D-1}, \pi_{\text{safe}}\}$;
  $x_{k+1} \leftarrow$ apply $u(t) = \Pi_{k,0}(x(t))$ to (2) until $t = (k+1)h$;
  collect observation and update $\underline{\gamma}, \overline{\gamma}$

We then have the following guarantee of safety for all timesteps, even if the MPC problems are infeasible.

Theorem 1. Given a system (2) under the assumptions made in Problem 1 with $x_0 \in \mathcal{X}_{\text{safe}}$, the control strategy resulting from Algorithm 1 is safe with probability at least $1 - \eta$.

The proof of this theorem is built on the following Lemmas.

Lemma 1. Given a system (2) under the assumptions made in Problem 1 at timestep $k = 0$ with $x_0 \in \mathcal{X}_{\text{safe}}$, the control strategy resulting from Algorithm 1 is safe with probability at least $1 - \eta$.

PROOF. Given the constraints of the MPC problem, the produced reachable sets $[x(t), \widehat{x}(t)]$ do not intersect any portion of $\mathcal{X}_{\text{unsafe}}$. As these reachable sets hold with probability at least $1 - \eta$ per (22), it follows that the overall strategy Π that produces these reachable sets is safe with probability at least $1 - \eta$.

Thus, if the MPC problem (24) is feasible for either obj = safe or obj = goal, the resulting strategy is safe with probability at least $1 - \eta$.

If the MPC problem is infeasible in both cases, Algorithm 1 produces a control strategy that only consists of applying π_{safe}. As the system is initialized in $\mathcal{X}_{\text{safe}}$, this guarantees the system's safety per the problem setup. ■

Lemma 2. Given a system (2) under the assumptions made in Problem 1, and a control strategy computed

by Algorithm 1 in the previous timestep $k-1$ which is safe with probability at least $1-\eta$, the control strategy resulting from Algorithm 1 at timestep k is safe with probability at least $1-\eta$.

PROOF. Again, if the MPC problem (24) is feasible for either obj = safe or obj = goal, the resulting strategy is safe with probability at least $1-\eta$.

If the MPC problem is infeasible in both cases, Algorithm 1 produces a control strategy that appends π_{safe} to the strategy computed in the previous timestep.

We then consider the potential origins of the previous strategy. If the previous strategy was originally computed by solving (24) with obj = safe, by the constraints of the MPC problem, applying this strategy results in the system avoiding $\mathcal{X}_{\text{unsafe}}$ and ending in $\mathcal{X}_{\text{safe}}$ with probability at least $1-\eta$. Thus, appending π_{safe} to the end of this strategy preserves this safety via the problem setup. It is not possible for the previous strategy to have resulted from solving (24) with obj = goal, as Algorithm 1 immediately executes the entire strategy if that problem is feasible.

If the previous strategy is a result of the MPC problem being infeasible in both cases, then that previous strategy will have, at some point, entered $\mathcal{X}_{\text{safe}}$ and then started only executing π_{safe}. Thus, appending π_{safe} to this strategy preserves safety via the problem setup. ■

With the previous Lemmas now proven, we now formally prove Theorem 1.

PROOF. Proof of Theorem 1: At timestep $k=0$, the control strategy produced by Algorithm 1 is safe with probability at least $1-\eta$ via Lemma 1, and must be safe with probability at least $1-\eta$ for every timestep afterward via Lemma 2. Thus, the control strategy produced by Algorithm 1 is safe for all timesteps with probability at least $1-\eta$. ■

Thus, this control scheme solves Problem 1. We emphasize that the key features of Algorithm 2 that make it computationally tractable and always safe are: 1) the sequential solving of a goal-reaching strategy and, failing that, a safe strategy; 2) the persistent safety property that there always exists a known safe control action from the previous step that can be executed next if needed; 3) the efficient computation of high probability reachable sets using mixed monotone systems theory.

7 An Optimistic Control Algorithm

Our second main contribution is to modify the control scheme outlined in the previous section with the goal of solving Problem 2. The key insight is that the values of β are typically fixed to guarantee the probability of bounding the disturbance behavior defined by $1-\eta$. As outlined in Section 6, this allows for safety guarantees that hold with high probability. However, these high probability guarantees often result in conservative control actions that leads to suboptimal cost minimization.

Thus, adjusting these bounds allows us to calculate alternative trajectories that hold with lower probability. In other words, we allow the system more freedom to explore cost-minimizing paths by reducing the required amount of conservatism in estimating the disturbance behavior.

We modify the embedding system to accept the desired level of confidence as an input, resulting in

$$\begin{bmatrix}\dot{x}\\ \dot{\widehat{x}}\end{bmatrix} = e(x,\widehat{x},u,\underline{\beta},\overline{\beta}) := \begin{bmatrix}\delta(x,u,\underline{\gamma}_{\underline{\beta}}(x,\widehat{x}),\widehat{x},\overline{\gamma}_{\overline{\beta}}(x,\widehat{x}))\\ \delta(\widehat{x},u,\overline{\gamma}_{\overline{\beta}}(x,\widehat{x}),x,\underline{\gamma}_{\underline{\beta}}(x,\widehat{x}))\end{bmatrix} \tag{25}$$

where $\underline{\gamma}_{\underline{\beta}}$ and $\overline{\gamma}_{\overline{\beta}}$ represent the bounding functions (20) and (21) with their values of β_t set to $\underline{\beta}$ and $\overline{\beta}$, respectively.

We then sample this new embedding system with timestep h such that at each step $k = t/h$,

$$\begin{bmatrix}x[k+1]\\ \widehat{x}[k+1]\end{bmatrix} = \Phi^e(h;(x[k],\widehat{x}[k]),\pi_k,\underline{\beta},\overline{\beta}), \tag{26}$$

where π_k is again the zero-order hold controller applied from time kh to $(k+1)h$. Below, we assume π_k is a constant policy $\pi_k(t,x) \equiv u_k$ for some $u_k \in \mathcal{U}$ to be designed by an MPC scheme. Thus, taking $\widehat{R}_k = [x[k],\widehat{x}[k]]$ overapproximates the reachable set of (2) at time $t = kh$ with high probability. These discretized reachable sets

are then included in the MPC as follows:

$$\underset{\Pi,\underline{\beta},\overline{\beta}}{\text{minimize}} \quad J_k(\widehat{R},\Pi) \tag{27}$$

subject to:

$$(26),\ x[0]=\widehat{x}[0] \text{ given},\ u_d\in\mathcal{U} \quad \forall d\in\{0,\dots,D-1\}$$
$$\widehat{R}_d=[x[d],\widehat{x}[d]] \quad \forall d\in\{0,\dots,D\}$$
$$\widehat{R}_D\subset\mathcal{X}_{\text{obj}}$$
$$\underline{\beta},\overline{\beta}\in[\beta_{\text{MIN}},\beta_{\text{MAX}}],$$
$$P(\underline{\gamma}_{\underline{\beta}}\preceq g(x)\preceq\overline{\gamma}_{\overline{\beta}})\geq 1-\eta_o$$

where $\mathcal{X}_{\text{obj}}$ is the goal hyperrectangle, and $J_{k,\text{obj}}$ is the desired cost function. The value of β_{MAX} is the value of $\sqrt{\beta_t}$ where β_t is defined as in [9, Theorem 7], which results in bounding functions that encapsulate the disturbance behavior with probability at least $1-\eta$, where η represents the same value as in Section 6. The value of β_{MIN} where $\beta_{\text{MIN}}\in[0,\beta_{\text{MAX}}]$ is a user-defined value. Additionally, $1-\eta_o$ where $1-\eta_o\in[0,1-\eta]$ is a user-defined value that determines the minimum probability desired for the bounds to encapsulate the disturbance behavior. This chance constraint is imposed by converting the chosen $\underline{\beta},\overline{\beta}$ using the associated z-score to the resulting probability value based on the Gaussian distribution. Thus, when this problem is feasible, it produces the lowest-cost set of inputs that is guaranteed to drive the system into the objective with probability at least $1-\eta_o$.

We then note that the nominal strategy of fixing $\underline{\beta},\overline{\beta}=\beta_{\text{MIN}}=\beta_{\text{MAX}}$ recovers the control strategy that solves Problem 1. While this solution provides feasible trajectories, it effectively assumes the worst-case disturbance behavior and as a result may be overly conservative and thus incur more cost than is necessary. Thus, moving forward we refer to this strategy as Pessimistic. This strategy is outlined in Algorithm 2.

Algorithm 2 Pessimistic Control Strategy

1: **Data:** Embedding system (25) sampled as (26), bounding functions $\underline{\gamma}_{\underline{\beta}},\overline{\gamma}_{\overline{\beta}}$
2: **for** $k=0,1,\dots$ **do**
3: $(\Pi,\underline{\beta},\overline{\beta})\leftarrow$ solve MPC (27), $\beta_{\text{MIN}}=\beta_{\text{MAX}}$;
4: $\Pi_k\leftarrow\Pi$;
5: $x_{k+1}\leftarrow$ apply $u(t)=\Pi_{k,0}(x(t))$ to (2) until $t=(k+1)h$;
6: collect observation and update $\underline{\gamma}_{\underline{\beta}},\overline{\gamma}_{\overline{\beta}}$

Our proposed strategy is to allow the MPC to modify the bounds by setting $\beta_{\text{MIN}}=0$ and selecting a probability η_o such that $1-\eta_o\leq 1-\eta$, thereby expanding the search space of feasible trajectories available to the solver. While allowing the calculated bounds to shrink means that the resulting bounds have a lower probability of encapsulating the disturbance, the resulting exploration and new observations allow the system to take advantage of the disturbance behavior in areas that further decrease the cost incurred. Selecting $1-\eta_o>0$ additionally preserves some of the robustness of Pessimism, by accounting for the fact that new observations are going to be collected and allowing the MPC scheme to "look ahead" despite being open-loop. As the trajectories produced are essentially speculation, we refer to this strategy moving forward as Optimistic. This strategy is outlined in Algorithm 3.

Algorithm 3 Optimistic Control Strategy

1: **Data:** Embedding system (25) sampled as (26), bounding functions $\underline{\gamma}_{\underline{\beta}},\overline{\gamma}_{\overline{\beta}}$
2: **for** $k=0,1,\dots$ **do**
3: $(\Pi,\underline{\beta},\overline{\beta})\leftarrow$ solve MPC (27), $\beta_{\text{MIN}}=0$;
4: $\Pi_k\leftarrow\Pi$;
5: $x_{k+1}\leftarrow$ apply $u(t)=\Pi_{k,0}(x(t))$ to (2) until $t=(k+1)h$;
6: collect observation and update $\underline{\gamma}_{\underline{\beta}},\overline{\gamma}_{\overline{\beta}}$

An example of the different trajectories produced by each strategy is shown in Figure 2 for the planar multirotor system described in Section 8. The system has a few observations of the disturbance in the range $Z\in[0,4]$. The red (Pessimistic) trajectory assumes worst-case bounds and thus prioritizes exploitation of the known disturbance behavior, while the blue (Optimistic) trajectory assumes tighter confidence bounds, though they may not correctly encapsulate the disturbance behavior. Solving the optimal control problem for the true disturbance behavior results in the black trajectory.

In the next sections, we provide a proof that the Optimistic strategy incurs lower expected cost than the Pessimistic strategy for a simplified system, as well as empirical results showing the same on a higher-dimensional quadrotor system.

7.1 Theoretical Results from a Simplified Setting

The above algorithm applies to general nonlinear systems in continuous-time and, as we demonstrate in the following sections, its benefits are supported empirically. However, its generality prevents provable guarantees. In this subsection, we explore a simplified setting under which theoretical guarantees are available. These results provide a degree of justification to the empirical successes that follow in Section 8.

Consider the discrete-time system

$$x[k+1]=x[k]+u[k]+w \tag{28}$$

with state x, input u, and disturbance $w=g(x)$ for some unknown function $g(x)$, from which observations can be drawn. We task the controller with driving the system

into a goal region $X_G \in [\underline{x}_G, \overline{x}_G]$ within two timesteps while minimizing the total input used, i.e. we want to solve

$$\begin{aligned} &\underset{u[0],u[1]}{\text{minimize}} \quad |u[0]| + |u[1]| \\ &\text{subject to:} \\ &\quad (28),\ x[0] \text{ given}, \\ &\quad x[1] \in X_G \ || \ x[2] \in X_G. \end{aligned} \tag{29}$$

At $k \in \{0, 1\}$, we assume that we can noiselessly observe $g(x[k])$. Following the approach of Sections 6-7, we form the associated embedding system of (28) and insert the appropriate confidence bounds on $g(x)$, resulting in

$$\begin{bmatrix} x[k+1] \\ \widehat{x}[k+1] \end{bmatrix} = \begin{bmatrix} x[k] + u[k] + \underline{\gamma}_{\underline{\beta}}(x[k], \widehat{x}[k]) \\ \widehat{x}[k] + u[k] + \overline{\gamma}_{\overline{\beta}}(x[k], \widehat{x}[k]) \end{bmatrix} \tag{30}$$

Thus, as a proxy to (29), we solve

$$\begin{aligned} &\underset{u[0],u[1],\underline{\beta},\overline{\beta}}{\text{minimize}} \quad |u[0]| + |u[1]| \\ &\text{subject to:} \\ &\quad (30),\ x[0] = \widehat{x}[0] \text{ given}, \\ &\quad [x[1], \widehat{x}[1]] \subseteq X_G \ \text{ or } \ [x[2], \widehat{x}[2]] \subseteq X_G \\ &\quad \underline{\beta}, \overline{\beta} \in [\beta_{\text{MIN}}, \beta_{\text{MAX}}]. \end{aligned} \tag{31}$$

Note that, in this case, we impose no minimum probability requirement; as there are only two timesteps, the advantage of "looking ahead" is minimal.

Theorem 2. Given the optimization problem (29), the Pessimistic strategy of solving (31) with $\underline{\beta} = \overline{\beta} = \beta_{\text{MAX}}$ incurs greater expected cost than the Optimistic strategy of solving (31) allowing $\underline{\beta}, \overline{\beta} \in [0, \beta_{\text{MAX}}]$.

PROOF. We note that for this system, as we impose no minimum probability requirement, an option that is always available to the Optimistic strategy is to set $\underline{\beta} = \overline{\beta} = 0$, effectively modeling the disturbance directly by the mean of the GP produced by the available observations. Moving forward, we assume that the Optimistic strategy always exercises this option, as any trajectory that is feasible for the Optimistic strategy when $\underline{\beta}, \overline{\beta} \neq 0$ is also feasible when $\underline{\beta} = \overline{\beta} = 0$. Generally speaking, it is safe to assume that the Optimistic strategy reduces the bounds to the minimum probability requirement, as there is never a disadvantage in doing so.

Denote by $u_O^j[k], u_P^j[k]$ the control input proposed by the Optimistic and Pessimistic strategies, respectively, for timestep k calculated at timestep j, and denote by $u_O[k], u_P[k]$ the actual inputs applied by each strategy at timestep k. Similarly, we utilize $x_{\{O,P\}}^j[k], \widehat{x}_{\{O,P\}}^j[k]$ to denote the proposed reachable set hyperrectangle endpoints calculated by each strategy at timestep j for timestep k, and $x_{\{O,P\}}[k]$ to denote the actual state encountered by each strategy at timestep k. Finally, we utilize $J_{\{O,P\}}^j$ to denote the cost of the proposed control actions of each strategy calculated at timestep j, and $J_{\{O,P\}}$ to denote the actual incurred cost of each strategy.

First, we note that, given a feasible Pessimistic strategy $u_P^0[0], u_P^0[1]$ with resulting reachable set approximations $[x_P^0[1], \widehat{x}_P^0[1]], [x_P^0[2], \widehat{x}_P^0[2]]$, it must hold that

$$\begin{aligned} (|u_P^0[0]| + |u_P^0[1]|) - (|u_O^0[0]| + |u_O^0[1]|) \\ \geq \max\{\sigma(x_P^0[1])\beta_{\text{MAX}}, u_P^0[1]\}, \end{aligned} \tag{32}$$

or, equivalently,

$$J_P^0 - J_O^0 \geq \max\{\sigma(x_P^0[1])\beta_{\text{MAX}}, |u_P^0[1]|\} \tag{33}$$

as the Optimistic strategy can simply take the feasible control actions chosen by the Pessimistic strategy, and trim the excess input proposed by that strategy needed to overcome the uncertainty that Pessimism works with. For example, if $\mu(x_P^0[1])$ is such that

$$x_P^0[1] + \mu(x_P^0[1]) \leq \underline{x}_G, \tag{34}$$

the Pessimistic strategy proposes

$$u_P^0[1] = \underline{x}_G - (x_P^0[1] + \mu(x_P^0[1]) - \sigma(x_P^0[1])\beta_{\text{MAX}}) \tag{35}$$

while the Optimistic strategy may propose

$$u_O^0[1] = \underline{x}_G - (x_P^0[1] + \mu(x_P^0[1])). \tag{36}$$

We then note that, per the problem setup, it holds that

$$u_{\{O,P\}}^1[1] = u_{\{O,P\}}[1] \tag{37}$$

as at timestep $k = 1$, the strategies have perfect knowledge of $g(x)$ at their current state and thus knows exactly the value of $u[1]$ needed to arrive at the goal. Similarly, as the strategies have perfect knowledge of $g(x[0])$ at timestep $k = 0$, it holds that

$$x_{\{O,P\}}^0[1] = x_{\{O,P\}}[1] \tag{38}$$

Thus, the actual incurred cost of any given strategy is equivalent to

$$J_{\{O,P\}} = |u_{\{O,P\}}^0[0]| + |u_{\{O,P\}}^1[1]|. \tag{39}$$

We then note that, as the Optimistic strategy estimates the disturbance using $\mu(x)$ only, it holds that

$$E(u_O^0[1] - u_O^1[1]) = 0 \tag{40}$$

which consequently means that

$$E(J_O) = J_O^0 \leq J_P^0 - \max\{\sigma(x_P^0[1])\beta_{\text{MAX}}, |u_P^0[1]|\} \tag{41}$$

Finally, we note that the Pessimistic strategy has a limit to how much it can improve upon observing $g(x_P[1])$ if it has correctly bounded the disturbance behavior (i.e. $g(x_P[1]) \in [\mu(x_P[1]) - \sigma(x_P^0[1])\beta_{\text{MAX}}, \mu(x_P[1]) + \sigma(x_P^0[1])\beta_{\text{MAX}}]$), which is $E(|u_P^0[1]| - |u_P^1[1]|) \leq \max\{\sigma(x_P^0[1])\beta_{\text{MAX}}, |u_P^0[1]|\}$. If the Pessimistic strategy has incorrectly bounded the disturbance behavior, then $E(|u_P^0[1]| - |u_P^1[1]|) = 0$, as the Pessimistic bounds are formulated by taking the same deviation from the mean towards either side.

Thus, for the Pessimistic strategy,

$$E(|u_P^0[1]| - |u_P^1[1]|) \leq \max\{\sigma(x_P^0[1])\beta_{\text{MAX}}, |u_P^0[1]|\}. \tag{42}$$

As a result,

$$E(J_P) \geq J_P^0 - \max\{\sigma(x_P^0[1])\beta_{\text{MAX}}, |u_P^0[1]|\}. \tag{43}$$

Combining (41) and (43) gives

$$E(J_O) \leq E(J_P). \tag{44}$$

■

Thus, the Optimistic Strategy outlined solves Problem 2.

While this theoretical result is only proven for the system (28), in the next section we provide empirical results that show that the Optimistic strategy outperforms Pessimism on a higher-dimensional system, suggesting that this result is applicable to general systems.

8 Case Study Results

In this section we provide two case studies. In the first case study, we demonstrate the safety and goal-reaching capabilities of Algorithm 1 on a motorboat crossing a river. Additional examples are presented in [10], where we demonstrate this algorithm on a planar multirotor in a wind field and an autonomous vehicle on an icy road. In the second case study, we return to the planar multirotor operating in a wind field and perform a Monte Carlo simulation comparing the Optimistic and Pessimistic strategies outlined in Section 6. [1]

8.1 Boat on a River: A Demonstration of Algorithm 1

We consider a case study of a motorboat crossing a river where the exact flow behavior of the river is unknown. The position of the boat is (x, y), the forward velocity of the boat as v, the yaw angle as ψ. There exist two inputs: thrust produced by the motor u_1 and rudder position u_2. The resulting dynamics (adapted from [23]) are

$$\begin{aligned} \dot{v} &= -v + u_1 \\ \dot{\psi} &= 0.15 v u_2 \\ \dot{x} &= v\cos\psi + g_x(x) \\ \dot{y} &= -v\sin\psi - f_y + g_y(y) \end{aligned} \tag{45}$$

where f_y is the known average flow of the river in the negative y direction and g_x and g_y denote the disturbed flow of the river in the x and y directions, respectively.

The associated decomposition function takes the form

$$\delta(x, u, w, \widehat{x}, \widehat{w}) = \begin{bmatrix} d^v & d^\psi & d^x & d^y \end{bmatrix}^T \tag{46}$$

$$\begin{aligned} d^v &= -v + u_1 \\ d^\psi &= \begin{cases} 0.15\min\{vu_2, \widehat{v}u_2\}, & \widehat{v} \geq v \\ 0.15\max\{vu_2, \widehat{v}u_2\}, & \widehat{v} \leq v \end{cases} \\ d^x &= d^{b_1 b_2}\left(\begin{bmatrix} v \\ d^{\cos}(\psi, \widehat{\psi}) \end{bmatrix}, \begin{bmatrix} \widehat{v} \\ d^{\cos}(\widehat{\psi}, \psi) \end{bmatrix} \right) + w_x \\ d^y &= -d^{b_1 b_2}\left(\begin{bmatrix} \widehat{v} \\ d^{\sin}(\widehat{\psi}, \psi) \end{bmatrix}, \begin{bmatrix} v \\ d^{\sin}(\psi, \widehat{\psi}) \end{bmatrix} \right) - f_y + w_y \end{aligned}$$

where $d^{b_1 b_2}$ is defined as

$$d^{b_1 b_2}(b, \widehat{b}) = \begin{cases} \min\{b_1 b_2, \widehat{b}_1 b_2, b_1 \widehat{b}_2, \widehat{b}_1 \widehat{b}_2\}, & \text{if } b \preceq \widehat{b} \\ \max\{b_1 b_2, \widehat{b}_1 b_2, b_1 \widehat{b}_2, \widehat{b}_1 \widehat{b}_2\}, & \text{if } \widehat{b} \preceq b. \end{cases} \tag{47}$$

and $d^{\sin}, d^{\cos}$ are the known tight decomposition functions for sin and cos, respectively (see [9, Equations 74–75]).

We define the safety set $\mathcal{X}_{\text{safe}}$ as the hyperrectangle of states $[(-5000, -6\pi, -2, -5), (5000, 6\pi, 1, 15)]$ and task the boat with crossing the river, i.e., reaching the goal set $[(-5000, -6\pi, 9, -5), (5000, 6\pi, 1, 12)]$. We also want the system to avoid several rocks that exist within the river,

[1] A code repository for reproducing each of these case studies is available at `https://github.com/gtfactslab/Cao_OptimisticControl`.

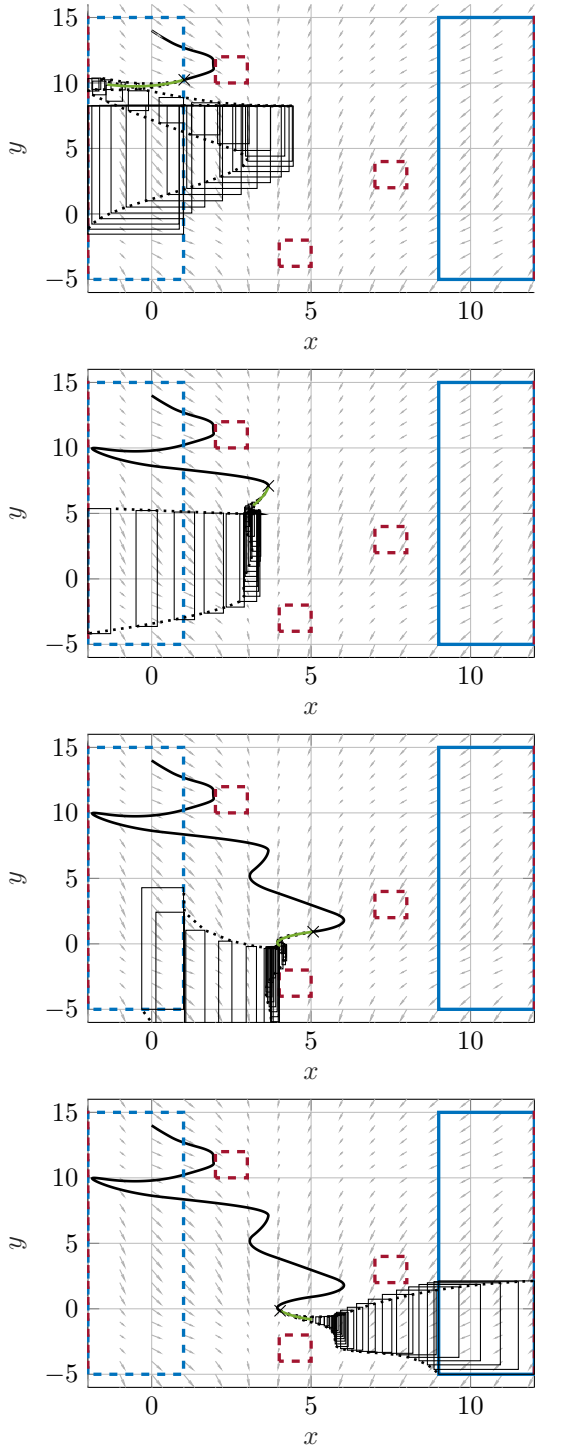

Fig. 3. Execution of the river motorboat case study. The arrows denote the river flow acting on the system. Additionally, the set of states where $x < -2$ and $x > 12$ are considered part of $\mathcal{X}_{\text{unsafe}}$.

denoted by the red dashed hyperrectangles in Figure 3, as well as the riverbank, which we denote by the sets of states where $x < -2$ and $x > 12$. We check that five intermediate reachable sets between each timestep satisfy $[x(t), \hat{x}(t)] \subset (\mathcal{X}_{\text{unsafe}})^{\complement}$, and solve (24) for $D = 5$ timesteps.

Our cost functions for each MPC scheme are as follows. We define our cost function for reaching the goal as

$$J_{k,\text{goal}}(\widehat{R}_0, ..., \widehat{R}_D) = \sum_{d=0}^{D} ||C_d - C_{\text{goal}}||_2 \tag{48}$$

where C_d and C_{goal} denote the center points of the reachable and goal hyperrectangles, respectively, and we define our cost function for exploration as

$$\begin{aligned} J_{k,\text{safe}}(\widehat{R}_0, ..., \widehat{R}_D) = \\ - \sum_{d=0}^{D} \left(\sigma_x(x_d) - ||x_d - x_{\text{goal}}|| e^{-\sigma_x(x_d)} \right) \\ - 0.5 \sum_{d=0}^{D} \left(\sigma_y(y_d) - ||y_d - y_{\text{goal}}|| e^{-\sigma_y(y_d)} \right) \end{aligned} \tag{49}$$

where $x_d, x_{\text{goal}}, y_d, y_{\text{goal}}$ denote the x and y center points of the reachable and goal hyperrectangles, and σ_x, σ_y denote the current estimated standard deviation of g_x, g_y. This makes the MPC solver prioritize information gain, similarly to the case study in [18], while biasing it toward states that bring the system closer to the goal. Multiplying the bias terms by $e^{-\sigma}$ allows the bias to be overcome if the expected information gain is high enough, and the exponential function is specifically chosen to mirror the structure of the GP radial basis kernel.

As shown in Figure 3, the system is initially unable to find a feasible control strategy that drives the boat to the goal while avoiding the unsafe areas. Thus, it reverts to exploring the river in order to gather observations of the river flow behavior (first, second, and third plots). After sufficient exploration, the system is able to find a feasible strategy that ends in the goal region (fourth plot), at which point the algorithm executes that strategy and terminates. Thus, at all times, the boat either has a safe path back to the side of the river it started on, or a safe path to the other side.

For additional demonstrations involving an autonomous vehicle navigating an icy road and a planar multirotor in a wind field, see [10].

8.2 Comparing Optimism and Pessimism

We turn to the case study outlined in Section 1 of a planar quadrotor flying in an unknown wind field. In

this system, the horizontal and vertical position of the multirotor are denoted y and z, and the roll angle is denoted θ. There are also two inputs: thrust u_1 acting at the center of mass in the direction $\begin{bmatrix} -\sin\theta & \cos\theta \end{bmatrix}^T$, and the roll angular velocity u_2. Thus, the resulting dynamics with normalized mass and moment of inertia are

$$\begin{aligned} \ddot{y} &= -u_1 \sin\theta + g_1(z) \\ \ddot{z} &= u_1 \cos\theta - a_g + g_2(z) \\ \dot{\theta} &= u_2 \end{aligned} \tag{50}$$

where g_1 and g_2 constitute the unknown wind forces in the horizontal and vertical directions, respectively. For this case study, we note that both functions g_1 and g_2 are only dependent on z; this is a deliberate choice to illustrate the risk/reward tradeoff, as flying to different altitudes will naturally incur more cost in the short term with the hope of reducing cost in the long term.

The associated decomposition function takes the form

$$\delta(x, u, w, \widehat{x}, \widehat{w}) = \begin{bmatrix} v_y & d^{v_y} & v_z & d^{v_z} & \omega & u_2 \end{bmatrix}^T \tag{51}$$

$$d^{v_y} = -d^{b_1 b_2}\left(\begin{bmatrix} u_1 \\ d^{\sin}(\widehat{\theta}, \theta) \end{bmatrix}, \begin{bmatrix} u_1 \\ d^{\sin}(\theta, \widehat{\theta}) \end{bmatrix} \right) + w_1$$

$$d^{v_z} = d^{b_1 b_2}\left(\begin{bmatrix} u_1 \\ d^{\cos}(\theta, \widehat{\theta}) \end{bmatrix}, \begin{bmatrix} u_1 \\ d^{\cos}(\widehat{\theta}, \theta) \end{bmatrix} \right) - a_g + w_2$$

where $d^{b_1 b_2}$, $d^{\sin}$, and $d^{\cos}$ are defined as before. Thus, with this decomposition function we can craft our associated embedding system and MPC schemes accordingly.

We task the quadrotor with travelling to a known objective region while minimizing the sum of the absolute value of the forces F_1, F_2 applied to the system through the propellers; these are easily derived from u_1 and u_2 by solving the linear set of equations

$$\begin{bmatrix} 1 & 1 \\ \frac{L}{2} & -\frac{L}{2} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{52}$$

where L is the distance from the propeller to the center of mass of the multirotor.

Thus, our final cost function becomes

$$J(\Pi) = \sum_{d=0}^{D-1} |F_{1,d}| + |F_{2,d}|. \tag{53}$$

We perform a Monte Carlo simulation, varying the underlying disturbance functions g_1 and g_2 for each iteration, and initializing the system with a few observations.

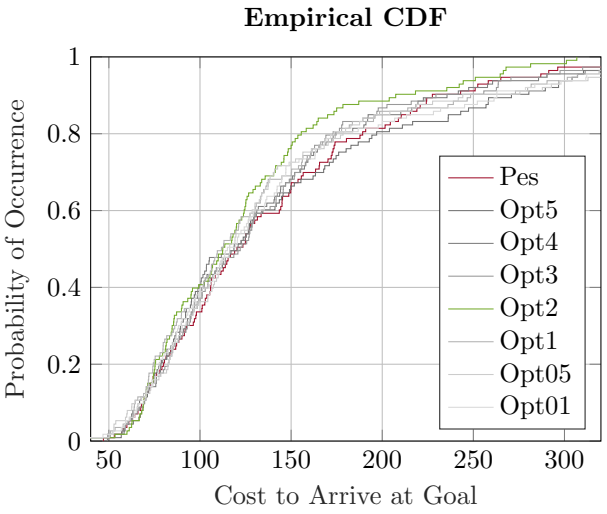

Fig. 4. Empirical cumulative distribution function (cdf) of the incurred cost of each strategy over 113 total runs. Each curve represents the overall results of each strategy executed in the Monte Carlo simulation. The horizontal axis denotes the cost incurred to arrive at the goal, while the vertical axis denotes the proportion of runs which incurred that cost or lower. The closer to the left a curve is, the better. For example, the Opt2 Strategy (green) incurred a cost of 150 or lower in approximately 80% of its runs. From these results, we can see that there is a tradeoff. Compared to the Pessimistic strategy (red), the Optimistic Strategies (gray) had a larger proportion of runs incur a cost of 200 or lower. However, they generally had a *smaller* proportion of runs incur a cost of 250 or lower. Only one strategy, Opt2 (green), which has a minimum probability $1 - \eta = 0.2$, unambiguously outperforms Pessimism. The respective minimum probabilities $1 - \eta$ of each strategy are outlined in Table 1.

Strategy	1 - η	Mean	Std. Dev.
Pessimism	$\geq$0.99	140.97	77.88
Opt5	$\geq$0.5	144.72	85.69
Opt4	$\geq$0.4	136.20	70.70
Opt3	$\geq$0.3	136.34	70.89
Opt2	$\geq$0.2	124.44	57.60
Opt1	$\geq$0.1	136.50	81.89
Opt05	$\geq$0.05	139.72	85.01
Opt01	$\geq$0.01	140.21	84.11

Table 1
Statistical summary of the incurred cost of the Pessimistic strategy as well as the Optimistic strategies tested with different minimum required probabilities.

We then simulate each strategy and record the total cost incurred once the multirotor has reached the objective. Figure 2 showcases an instance of the Monte Carlo simulation and the resulting trajectories produced by some of the strategies.

We plot the resulting empirical cumulative distribution

function (cdf) of the incurred cost of each strategy over 113 runs in Figure 4 and provide statistics in Table 1. Overall, the Optimistic strategies tended to outperform the Pessimistic strategy, though as can be seen, the minimum probability requirement affects the performance of the Optimistic strategy. It is especially important to note that there seems to be an ideal tradeoff; the lowest minimum probability does not necessarily translate to the lowest incurred cost. This is in line with the idea that a nonzero minimum probability η_o gives the MPC scheme the ability to "look ahead" while retaining some robustness.

These results suggest that Optimism is a step in the right direction towards deriving the optimal theoretical strategy. Allowing the controller to determine the best-case disturbance bounds while still ensuring it keeps known observations in mind is a good tradeoff between exploration and exploitation, and the minimum probability requirements can be tweaked as necessary.

9 Conclusion

We have presented a control scheme that is guaranteed to be safe with high probability while enabling both exploration and goal reaching. This control scheme leverages mixed monotonicity theory in an MPC formulation that is capable of calculating hyperrectangular overapproximations of reachable sets that hold with high probability. This MPC formulation is then used in an algorithm which produces a control strategy that is safe with high probability. The proposed algorithm is tunable for both exploration and goal reaching, incorporates unknown disturbances nonlinearly, and is scalable to systems of moderately high dimension due to the efficiency of the reachable set computations.

We then modified this control scheme into a cost-aware controller formulation that can speculate on the existence of lower-cost trajectories than the one resulting from assuming worst-case bounds on the disturbance behavior. By adjusting the confidence levels in the Gaussian processes, possible trajectories can be calculated. We formulated an Optimistic approach to selecting these trajectories and showed that it incurs lower expected cost than the nominal Pessimistic strategy, first with theoretical results on a simplified system, and then with empirical results on a planar multirotor system. Future work includes deriving results for general nonlinear systems.

Acknowledgements

This project was partially supported by the National Science Foundation under award #1749357, and the Ford Motor Company. Additionally, the NASA University Leadership Initiative (grant #80NSSC20M0163) provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity.

References

[1] Matthew Abate, Maxence Dutreix, and Samuel Coogan. Tight decomposition functions for continuous-time mixed-monotone systems with disturbances. *IEEE Control Systems Letters*, 5(1):139–144, 2021.

[2] Anayo K. Akametalu, Jaime F. Fisac, Jeremy H. Gillula, Shahab Kaynama, Melanie N. Zeilinger, and Claire J. Tomlin. Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control*, pages 1424–1431, 2014.

[3] Anayo K. Akametalu, Jaime F. Fisac, Jeremy H. Gillula, Shahab Kaynama, Melanie N. Zeilinger, and Claire J. Tomlin. Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control*, pages 1424–1431, 2014.

[4] D. Angeli and E. D. Sontag. Monotone control systems. *IEEE Transactions on Automatic Control*, 48(10):1684–1698, 2003.

[5] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[6] Mona Buisson-Fenet, Friedrich Solowjow, and Sebastian Trimpe. Actively learning gaussian process dynamics. In *Learning for dynamics and control*, pages 5–15. PMLR, 2020.

[7] Monimoy Bujarbaruah, Siddharth H. Nair, and Francesco Borrelli. A semi-definite programming approach to robust adaptive mpc under state dependent uncertainty. In *2020 European Control Conference (ECC)*, pages 960–965, 2020.

[8] Monimoy Bujarbaruah, Charlott Vallon, and Francesco Borrelli. Learning to satisfy unknown constraints in iterative mpc. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 6204–6209, 2020.

[9] Michael Enqi Cao, Matthieu Bloch, and Samuel Coogan. Efficient learning of hyperrectangular invariant sets using gaussian processes. *IEEE Open Journal of Control Systems*, pages 1–14, 2022.

[10] Michael Enqi Cao and Samuel Coogan. Safe learning-based predictive control from efficient reachability. In *AACC American Control Conference (ACC)*, 2023.

[11] S. Coogan. Mixed monotonicity for reachability and safety in dynamical systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5074–5085, 2020.

[12] Vikas Dhiman, Mohammad Javad Khojasteh, Massimo Franceschetti, and Nikolay Atanasov. Control barriers in bayesian learning of system dynamics. *IEEE Transactions on Automatic Control*, 68(1):214–229, 2023.

[13] Jaime F. Fisac, Anayo K. Akametalu, Melanie N. Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2019.

[14] Sylvia Herbert, Jason J. Choi, Suvansh Sanjeev, Marsalis Gibson, Koushil Sreenath, and Claire J. Tomlin. Scalable learning of safety guarantees for autonomous systems using

hamilton-jacobi reachability. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5914–5920, 2021.

[15] Pushpak Jagtap, George J. Pappas, and Majid Zamani. Control barrier functions for unknown nonlinear systems using gaussian processes. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3699–3704, 2020.

[16] Mouhyemen A. Khan, Tatsuya Ibuki, and Abhijit Chatterjee. Gaussian control barrier functions: Non-parametric paradigm to safety. *IEEE Access*, 10:99823–99836, 2022.

[17] Craig Knuth, Glen Chou, Necmiye Ozay, and Dmitry Berenson. Planning with learned dynamics: Probabilistic guarantees on safety and reachability via lipschitz constants. *IEEE Robotics and Automation Letters*, 6(3):5129–5136, 2021.

[18] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6059–6066, 2018.

[19] Johannes Köhler, Peter Kötting, Raffaele Soloperto, Frank Allgöwer, and Matthias A. Müller. A robust adaptive model predictive control framework for nonlinear uncertain systems. *International Journal of Robust and Nonlinear Control*, 31(18):8725–8749, 2021.

[20] Thomas Lew, Apoorva Sharma, James Harrison, Andrew Bylard, and Marco Pavone. Safe active dynamics learning and control: A sequential exploration–exploitation framework. *IEEE Transactions on Robotics*, 38(5):2888–2907, 2022.

[21] JM Manzano, J Calliess, D Munoz de la Pena, and D Limon. Online learning robust mpc: an exploration-exploitation approach. *IFAC-PapersOnLine*, 53(2):5292–5297, 2020.

[22] Sara Pohland, Sylvia Herbert, and Claire Tomlin. Efficient safe learning for robotic systems in unstructured environments. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, pages 82–86, 2019.

[23] B. Potočnik, G. Mušič, and B. Zupančič. Model predictive control of discrete-time hybrid systems with discrete inputs. *ISA Transactions*, 44(2):199–211, 2005.

[24] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2006.

[25] Ugo Rosolia and Francesco Borrelli. Learning model predictive control for iterative tasks. a data-driven control framework. *IEEE Transactions on Automatic Control*, 63(7):1883–1896, 2018.

[26] Lorenzo Sforni, Ivano Notarnicola, and Giuseppe Notarstefano. Learning-driven nonlinear optimal control via gaussian process regression. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4412–4417, 2021.

[27] Jennifer C. Shih, Franziska Meier, and Akshara Rai. A framework for online updates to safe sets for uncertain dynamics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5994–6001, 2020.

[28] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, USA, 2010.

[29] Dawei Sun, Mohammad Javad Khojasteh, Shubhanshu Shekhar, and Chuchu Fan. Uncertain-aware safe exploratory planning using gaussian process and neural control contraction metric. In *Learning for Dynamics and Control*, pages 728–741. PMLR, 2021.

[30] Kim Peter Wabersich and Melanie Zeilinger. Bayesian model predictive control: Efficient model exploration and regret bounds using posterior sampling. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 455–464. PMLR, 10–11 Jun 2020.