Safe Learning-based Predictive Control from Efficient Reachability

Michael E. Cao and Samuel Coogan

Abstract— **We consider a dynamical system subject to a disturbance input that is an unknown function of the state. Given a target goal region, we propose a control scheme that encourages exploration of the state space in order to sample the dynamics and obtain an estimate of the unknown component while avoiding unsafe regions of the state space until the goal is able to be reached with high probability. By estimating the unknown component as a Gaussian process, we efficiently obtain hyperrectangular overapproximations of the reachable set for the system using the theory of mixed monotone systems, and these sets are improved over time as measurements of the dynamics are collected. Using these reachability estimates, we propose a model predictive scheme that avoids the unsafe region and ensures the system is always within reach of a conservative, guaranteed safe region that is given a priori, thus always ensuring feasibility until the goal is reachable. We demonstrate the approach on a model of an autonomous vehicle operating on an icy road and on a planar multirotor moving in an unknown wind field.**

I. 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 so as not to lead to unsafe behavior.

Learning and exploration with safety guarantees has been considered in [1], which proposes a discrete-time Model Predictive Control (MPC) algorithm that is guaranteed to be safe with high probability by ensuring that a path back to safety exists at every timestep. Alternatively, [2] presents a learning algorithm that explicitly considers safety defined in terms of Lyapunov stability guarantees, [3] proposes a general safety framework based on Hamilton-Jacobi reachability methods, [4]–[6] synthesize control barrier functions online to guarantee safety, and [7] achieves safety by estimating the Lipschitz constant of the disturbance. Other works, such as [8]–[10], explore learning and updating safety sets in an online manner. For MPC-based approaches, proposed frameworks are robust to certain types of uncertainty, for example by assuming a known Lipschitz constant [11], assuming the uncertainty is parametric [12], or applying MPC to iterative learning control [13].

We draw from the problem setup proposed in [1] and consider a nonlinear dynamical system whose dynamics are not fully known. As in [1], 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.

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 [14] for an overview) and utilize previous results [15], [16] 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 [17]. Lastly, we 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.

The rest of the paper is structured as follows: Section II introduces key notation, and Section III formally defines the problem. We then develop a controller that solves this problem in Section IV, before demonstrating its efficacy on two case studies in Section V: an autonomous vehicle traveling on an icy road and a planar multirotor operating in a wind field. Finally, we conclude with a short discussion in Section VI.

II. 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

M. E. Cao and S. Coogan are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, 30332, USA {mcao34, sam.coogan}@gatech.edu. S. Coogan is also with the School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, 30332, USA.

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.

hyperrectangle formed by the first and last n components of a, i.e., $[\![a]\!] := [x, y]$. Finally, let $\preceq_{\mathrm{SE}}$ denote the *southeast order* on $\mathbb{R}^{2n}$ defined by $(x, x') \preceq_{\mathrm{SE}} (y, y')$ if and only if $x \preceq y$ and $y' \preceq x'$. In particular, observe that when $x \preceq x'$ and $y \preceq y'$,

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

III. Problem Setup

Consider the continuous-time nonlinear dynamical system

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

with f 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 $\overline{u}$.

We denote by $\phi(t, x_0, \pi)$ the resulting closed-loop state trajectory of (2) under control strategy $u = \pi(t, x)$ 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.

Assumption 1. *There exists a known subset of the state space $\mathcal{X}_{\text{unsafe}} \subset \mathbb{R}^n$ which must be avoided. There also exists a known safe set $\mathcal{X}_{\text{safe}} \subset \mathbb{R}^n$ and corresponding time-invariant safety controller π_{safe} with $\pi_{safe}(x) \in \mathcal{U}$ 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.*

$$\phi(t, x_0, \pi_{safe}) \in (\mathcal{X}_{\text{unsafe}})^C \quad \forall t \geq 0, \quad \forall x_0 \in \mathcal{X}_{\text{safe}}. \qquad (3)$$

Our objective is to control the system to a goal region while avoiding the unsafe region.

Problem statement. *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 $\mathcal{X}_{\text{goal}} \subset \mathbb{R}^n$, compute a feedback control strategy $u = \pi(t, x)$ that reaches the goal while avoiding the unsafe region $\mathcal{X}_{\text{unsafe}}$,* i.e.*,*

$$\forall t \geq 0, \phi(t, x_0, \pi) \in (\mathcal{X}_{\text{unsafe}})^C \qquad (4)$$

$$\exists T > 0 \textit{ s.t. } \phi(T, x_0, \pi) \in \mathcal{X}_{\text{goal}}. \qquad (5)$$

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).

Our proposed control approach is to incrementally move towards the goal while 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. This safe return and guaranteed reach to the goal is ensured via a nonlinear MPC scheme which directly optimizes for a control input in both cases and incorporates uncertainty from the unknown component $g(x)$ of the dynamics. 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 next formalize this approach.

IV. Safe Learning Algorithm

In this section we develop a safe control scheme that is safe with high probability by leveraging the mixed monotonicity property of dynamical systems to calculate high-probability reachable sets, then utilize an MPC formulation to solve for a control strategy that always has a path back to safety. We define the reachable set of (2) at time $t = T$ initialized from any state $x_0 \in X_0$ with control policy u as

$$R(T, X_0, u) = \{\phi(T, x_0, u) \mid x_0 \in X_0\}. \qquad (6)$$

A. 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 $k \in \{1, \cdots, p\}$, $\frac{\partial \delta_i}{\partial w_k}(x, u, w, \widehat{x}, \widehat{w}) \geq 0$ & $\frac{\partial \delta_i}{\partial \widehat{w}_k}(x, u, w, \widehat{x}, \widehat{w}) \leq 0$ for all $x, \widehat{x}, u, w, \widehat{w}$.

For any system, there exists some decomposition function δ satisfying the above conditions [17], although one may not be readily available in closed form. In general, finding a decomposition function is problem specific; see [14] for further discussion and the case studies below for examples.

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 $(w, \widehat{w}) \in \mathbb{R}^p \times \mathbb{R}^p$:

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

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 [18] 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 $a \preceq_{\mathrm{SE}} a'$ and $c(t) \preceq_{\mathrm{SE}} c'(t)$ for all $t \geq 0$, then for all $t \geq 0$,

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

In other words, provided that the system is initialized within $[\underline{x}_0, \overline{x}_0]$, 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.*

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

We have shown that, given some assumptions [16, Assumptions 3 & 4] on $g(x)$, it is possible to derive bounding functions $\underline{\gamma}(x, \widehat{x}), \overline{\gamma}(x, \widehat{x})$ on w using GP theory that hold with probability at least $1 - \eta$ for any $\eta \in (0, 1)$. We preserve these assumptions and select our hyperparameters accordingly, resulting in the embedding system

$$\begin{bmatrix} \dot{x} \\ \dot{\widehat{x}} \end{bmatrix} = e(x, u, \widehat{x}) := \begin{bmatrix} \delta(x, u, \underline{\gamma}(x, \widehat{x}), \widehat{x}, \overline{\gamma}(x, \widehat{x})) \\ \delta(\widehat{x}, u, \overline{\gamma}(x, \widehat{x}), x, \underline{\gamma}(x, \widehat{x})) \end{bmatrix}. \quad (10)$$

We denote the state of (10) at time t when initialized at $(\underline{x}_0, \overline{x}_0)$ under some input signal $u : [0, \infty) \to \mathbb{R}^m$ as $\Phi^e(t; (\underline{x}_0, \overline{x}_0), u)$. Thus, the hyperrectangle of states defined by $[\![\Phi^e(t; (\underline{x}_0, \overline{x}_0), u)]\!]$ overapproximates the reachable sets of (2) with probability at least $1 - \eta$. Crucially, since $\underline{\gamma}, \overline{\gamma}$ bound the entire disturbance function with probability $1 - \eta$, the uncertainty does not compound over successive reachable set estimations. As observations of the unknown behavior $g(x)$ are collected and the confidence bounds $\underline{\gamma}, \overline{\gamma}$ tighten, the hyperrectangular reachable set overapproximations also tighten. In the next section, we insert these hyperrectangles into a safe model predictive formulation.

B. Safe With High Probability MPC

We sample the embedding system (10) with timestep h such that at each step $k = t/h$,

$$\begin{bmatrix} x[k+1] \\ \widehat{x}[k+1] \end{bmatrix} = \Phi^e(h; (x[k], \widehat{x}[k]), \pi_k) \quad (11)$$

where π_k is the 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 (i.e. with probability at least $1 - \eta$).

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

$$\begin{aligned} \underset{\Pi = \{u_0, \ldots, u_D\}}{\text{minimize}} \quad & J_{k,\text{obj}}(\widehat{R}_0, \ldots, \widehat{R}_D) && (12) \\ \text{subject to:} \quad & & & \\ & (11),\ x[0] = \widehat{x}[0] \text{ given},\ u_d \in \mathcal{U} && \forall d \in \{0, \ldots, D-1\} \\ & \widehat{R}_d = [x[d], \widehat{x}[d]],\ \widehat{R}_D \subset \mathcal{X}_{\text{obj}} && \forall d \in \{0, \ldots, D\} \\ & [x(t), \widehat{x}(t)] \subset (\mathcal{X}_{\text{unsafe}})^C && \forall t \in [0, T] \end{aligned}$$

where 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 $J_{k,\text{safe}}$ which are designed to prioritize goal-reaching and exploration, respectively. If (12) is feasible when obj = goal, then Π contains a control input 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 (12) 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 (12) 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 $\mathcal{X}_{\text{goal}}$. We note that, in practice, to ensure the safety condition $[x(t), \widehat{x}(t)] \subset (\mathcal{X}_{\text{unsafe}})^C$ for all $t \in [0, T]$ in (12), we check this condition at a large number of time instances between the sampling times.

Algorithm 1: Resulting Control Scheme

```
Data: Safety controller $\pi_{\text{safe}}$, embedding system (10)
      sampled as (11), bounding functions $\underline{\gamma}, \overline{\gamma}$
$\Pi_0 \leftarrow \{\pi_{\text{safe}}, \ldots, \pi_{\text{safe}}\}$;
for $k = 0, 1, \ldots$ do
    $(feasible, \Pi) \leftarrow$ solve MPC problem, obj = goal;
    if feasible then
        apply $u = \Pi$ to system
        break;
    $(feasible, \Pi) \leftarrow$ solve MPC problem, 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
      $t = (k+1)h$
```

Theorem 1. *Given a system* (2) *under Assumption 1 with* $x_0 \in \mathcal{X}_{\text{safe}}$*, the control strategy resulting from Algorithm 1 is safe with high probability.*

Proof Sketch. Consider Algorithm 1 at timestep $k = 1$. If the MPC problem is feasible for obj = goal, the resulting control strategy Π_1 is guaranteed with high probability to drive the system to $\mathcal{X}_{\text{goal}}$ while avoiding $\mathcal{X}_{\text{unsafe}}$ by the fact that the reachable set overapproximations calculated by the MPC problem hold with high probability. Thus, the entire strategy Π_1 is executed and the algorithm terminates. If the MPC problem is feasible for obj = safe, the resulting control strategy Π_1 is guaranteed with high probability to drive the system to $\mathcal{X}_{\text{safe}}$ while avoiding $\mathcal{X}_{\text{unsafe}}$, thus Π_1 is safe with high probability. If neither MPC problem is feasible, Π_1 is safe with high probability by virtue of Π_0 being safe, as Π_1 appends π_{safe} to Π_0, which ends in $\mathcal{X}_{\text{safe}}$. This continues by induction: at timestep k, if either MPC problem is feasible, Π_k is safe with high probability. If not, Π_k is safe with high probability due to Π_{k-1} being safe.

V. Case Studies

In this section we apply Algorithm 1 to case studies of a four-dimensional autonomous vehicle operating on an icy road and a six-dimensional planar multirotor operating in a wind field. We task each with safely reaching a goal area in the presence of unknown disturbances (i.e. the ice and wind) while avoiding unsafe sets. Both case studies were implemented using the Model Predictive Control Toolbox in MATLAB, and a code repository is available at `https://github.com/gtfactslab/Cao_ACC2023`.

A. Autonomous Vehicle on Icy Road

We consider an autonomous vehicle modeled by the four-dimensional kinematic planar bicycle, which has state $x = [X, Y, \psi, v]^T$ and relates positional coordinates X and Y, center-of-mass velocity v, heading angle ψ, side-slip angle $\beta(u_2)$, and front and rear distances from center of mass $l_f = 2.2$m and $l_r = 3.3$m as

$$\begin{aligned} \dot{X} &= v\cos\left(\psi + \beta(u_2)\right), & \dot{Y} &= v\sin\left(\psi + \beta(u_2)\right), \\ \dot{\psi} &= \frac{v}{l_r}\sin\left(\beta(u_2)\right), & \dot{v} &= u_1, \end{aligned} \tag{13}$$

where

$$\beta(u_2) = \arctan\left(\frac{l_r}{l_f + l_r}\tan\left(u_2\right)\right), \tag{14}$$

with inputs being the desired acceleration u_1 and steering angle u_2. We assume the system is subject to constraints $\mathcal{U} = [(-600, -\pi/3), (600, \pi/3)]$.

The vehicle is operating on a road which varies in friction due to the presence of ice patches. As a result, the actual velocity update dynamics are given by

$$\dot{v}_{\text{actual}} = (1 - g(X, Y))u_1, \tag{15}$$

where the true behavior of $g(X, Y) \in [0, 1)$ represents the state-dependent change in friction based on the road surface at that position. We estimate g using a GP with a radial basis kernel, initialized with several points around the starting point of the vehicle, and obtain high-confidence bounds by considering posterior estimates up to three standard deviations from the mean. We set $h = 0.05$s and collect an observation of $g(X, Y)$ at every timestep.

The associated decomposition function takes the form

$$\delta(x, u, w, \widehat{x}, \widehat{w}) = \begin{bmatrix} d^X & d^Y & d^\psi & d^v \end{bmatrix}^T \tag{16}$$

$$d^v = d^{b_1 b_2}\left(\begin{bmatrix} 1 - \widehat{w} \\ u_1 \end{bmatrix}, \begin{bmatrix} 1 - w \\ u_1 \end{bmatrix}\right)$$

where, for $b, \widehat{b} \in \mathbb{R}^2$,

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

and d^X, d^Y, d^ψ take the same forms as in [16, Section VI-A].

We define the safety set as

$$\mathcal{X}_{\text{safe}} = [(-2, -5, -2\pi, -50), (0, 5, 2\pi, 50)] \tag{17}$$

and task the system with entering the goal set

$$\mathcal{X}_{\text{goal}} = [(5, -5, -2\pi, -15), (7, 5, 2\pi, 15)] \tag{18}$$

while avoiding $\mathcal{X}_{\text{unsafe}}$, which is the union of a set of hyperrectangles (see the dashed red boxes in Figure 1).

We then leverage the following observation:

Observation 1. *Given vectors (x, x') and (y, y') such that $x \preceq x'$ and $y \preceq y'$,*

$$(x, x') \preceq_{SE} (y', y) \iff [x, x'] \cap [y, y'] \neq \emptyset. \tag{19}$$

Thus, our safety constraint is equivalently defined as

$$(x(t), \widehat{x}(t)) \not\preceq_{\text{SE}} (\overline{x}_{ui}, \underline{x}_{ui})\ \forall_{t \geq 0, i} \tag{20}$$

where $[\underline{x}_{ui}, \overline{x}_{ui}]$ are the unsafe hyperrectangles. We check that six intermediate reachable sets between each timestep satisfy $[x(t), \widehat{x}(t)] \subset (\mathcal{X}_{\text{unsafe}})^C$ and solve the MPC problem (12) for $D = 4$ timesteps.

We take

$$J_{k,\text{goal}}(\widehat{R}_0, ..., \widehat{R}_D) = \sum_{d=0}^{D} ||C_d - C_{\text{goal}}|| \tag{21}$$

where C_d, C_{goal} are the center points of the respective reachable sets and goal. If a feasible solution is found, the set of control actions is executed immediately, as these represent a control strategy that is guaranteed to end in $\mathcal{X}_{\text{goal}}$ with high probability. If this is infeasible, (12) is solved with cost

$$\begin{aligned} &J_{k,\text{safe}}(\widehat{R}_0, ..., \widehat{R}_D) = \\ &\quad -\sum_{d=0}^{D}\left(\alpha\sigma(C_d) - ||C_d - C_{\text{goal}}||e^{-\alpha\sigma(C_d)}\right) \end{aligned} \tag{22}$$

where $\alpha > 0$ is a user-specified constant. This reverts the objective back to pure exploration, with a bias toward observation points that take the system closer to the goal. Multiplying the bias term $||C_d - C_{\text{goal}}||$ by $e^{-\alpha\sigma(C_d)}$ 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 1, initially, the system is unable to find a control strategy that is guaranteed to reach the goal while avoiding the unsafe areas with high probability, so it reverts to exploring the state space for the first few timesteps (first and second plots). After sufficient exploration, the controller is able to compute a strategy that reaches the goal, and executes this strategy immediately (third and fourth plot). Thus, at all times, the controller has a safe path to either $\mathcal{X}_{\text{safe}}$ or $\mathcal{X}_{\text{goal}}$ that holds with high probability. On average, it takes approximately 18 minutes to compute the next control action single-threaded on a personal computer, though timing analysis shows that around 60 percent of this computation is spent sampling the GP; these processes are parallelizable, though outside the scope of this work.

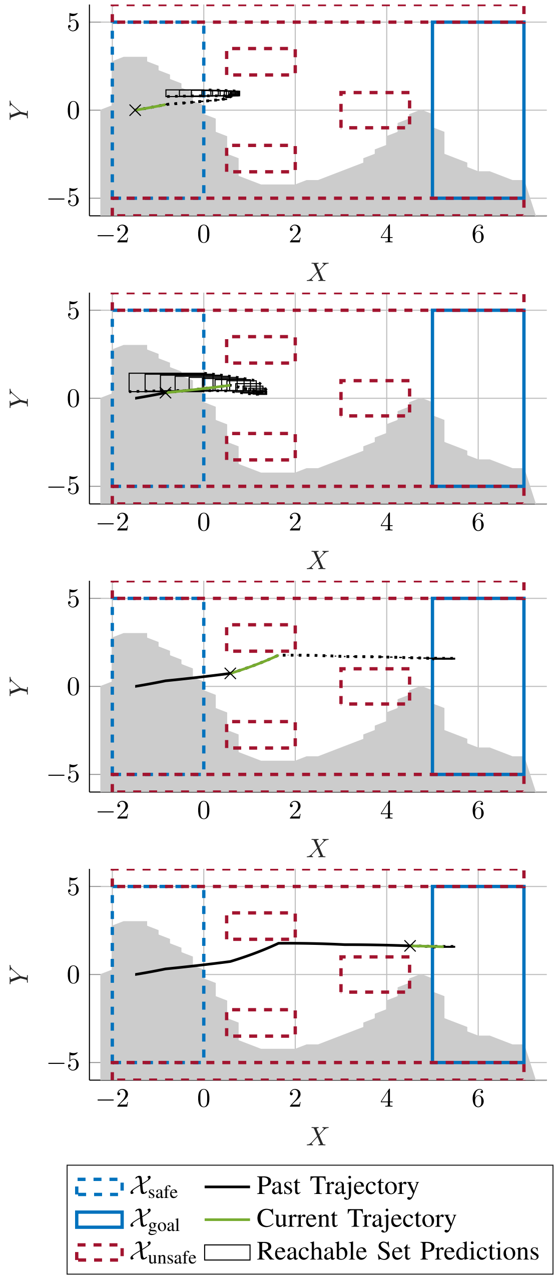

Fig. 1: Execution of the Autonomous Vehicle case study. The shaded region is the portion of the road unaffected by ice.

B. Planar Multirotor in Wind Field

We consider a multirotor aerial vehicle constrained to move in a vertical plane. The six-dimensional state x of the planar multirotor system consists of horizontal position y, vertical position z, roll angle θ, and their derivatives, $v_y = \dot{y}$, $v_z = \dot{z}$, $\omega = \dot{\theta}$, so that $x = \begin{bmatrix} y \; v_y \; z \; v_z \; \theta \; \omega \end{bmatrix}^T$. The two inputs are thrust u_1 acting at the center of mass in the direction $\begin{bmatrix} -\sin\theta & \cos\theta \end{bmatrix}^T$ perpendicular to the line segment connecting the rotors, and roll angular acceleration u_2. We assume the system is subject to input constraints $\mathcal{U} = [(-40, -2\pi), (40, 2\pi)]$, gravitational acceleration a_g, as well as an unknown force due to wind. We assume this force affects acceleration in both the horizontal and vertical directions and is a function of altitude z. The resulting dynamics with normalized mass and moment of inertia are

$$\begin{aligned} \ddot{y} &= \dot{v}_y = -u_1 \sin\theta + g_1(z) \\ \ddot{z} &= \dot{v}_z = u_1 \cos\theta - a_g + g_2(z) \\ \ddot{\theta} &= \dot{\omega} = u_2 \end{aligned} \tag{23}$$

where g_1 and g_2 constitute the unknown wind forces in the horizontal and vertical directions, respectively. We again estimate g_1 and g_2 using GPs with a radial basis function kernel, initialized with several points around the starting point of the multirotor, and obtain high confidence bounds by considering posterior estimates up to three standard deviations from the mean. We set $h = 0.2$s and collect an observation of $g_1(z)$ and $g_2(z)$ at every timestep.

The associated decomposition function takes the form

$$\delta(x, u, w, \widehat{x}, \widehat{u}, \widehat{w}) = \begin{bmatrix} v_y & d^{v_y} & v_z & d^{v_z} & \omega & u_2 \end{bmatrix}^T \tag{24}$$

$$\begin{aligned} d^{v_y} &= -d^{b_1 b_2}\left(\begin{bmatrix} u_1 \\ d^{\sin}(\widehat{\theta}, \theta) \end{bmatrix}, \begin{bmatrix} u_1 \\ d^{\sin}(\theta, \widehat{\theta}) \end{bmatrix} \right) + w_1 \\ d^{v_z} &= d^{b_1 b_2}\left(\begin{bmatrix} u_1 \\ d^{\cos}(\theta, \widehat{\theta}) \end{bmatrix}, \begin{bmatrix} u_1 \\ d^{\cos}(\widehat{\theta}, \theta) \end{bmatrix} \right) - a_g + w_2 \end{aligned}$$

where $d^{b_1 b_2}$ is defined as before and $d^{\sin}$, $d^{\cos}$ are the known tight decomposition functions for sin and cos, respectively (see [16, Equations 74–75]).

We define the safety set $\mathcal{X}_{\text{safe}}$ as the hyperrectangle $[(-5, -30, -2, -30, -2\pi, -2\pi), (5, 30, 0, 30, 2\pi, 2\pi)]$ and task the system with entering the goal set $\mathcal{X}_{\text{goal}} = [(-5, -8, 9, -8, -\pi, -\pi), (5, 8, 11, 8, \pi, \pi)]$ while avoiding $\mathcal{X}_{\text{unsafe}}$, which is the union of a set of hyperrectangles (see the dashed red boxes in Figure 2). We again leverage Observation 1 and define our safety constraint as (20). We check that seven intermediate reachable sets between each timestep satisfy $[x(t), \widehat{x}(t)] \subset (\mathcal{X}_{\text{unsafe}})^C$, and solve (12) for $D = 5$ timesteps.

Finally, we formulate our cost functions for each MPC scheme. We define our cost function for goal reaching as before with equation (21). If a feasible solution is found, the set of control actions is executed immediately, as these represent a control strategy that is guaranteed to end in $\mathcal{X}_{\text{goal}}$ with high probability. However, if this is infeasible, the controller attempts to explore the state space in a way that brings the rotor closer to the target altitude, while having a path back to safety. As g_1, g_2 are only functions of the altitude z, we can set altitude as the exploration metric $J_{k,\text{safe}}(\widehat{R}_0, \ldots, \widehat{R}_D) = -\sum_{d=0}^{D} C_{d,2}$.

As shown in Figure 2, the system is initially unable to compute a control strategy that is guaranteed to reach the goal while avoiding the unsafe areas, so it reverts to exploring the state space (first and second plots). After sufficient exploration, the controller is able to guarantee with high probability that it reaches the goal (third and fourth plot). Thus, at all times, the controller has a safe path to either $\mathcal{X}_{\text{safe}}$ or $\mathcal{X}_{\text{goal}}$ that holds with high probability. On average, it takes approximately nine minutes to compute the next control action single-threaded on a personal computer.

VI. 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. Future directions of research include developing different cost functions to optimize for alternative objectives and implementation on physical systems.

References

[1] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6059–6066, 2018.

[2] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[3] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019.

[4] M. A. Khan, T. Ibuki, and A. Chatterjee, "Gaussian control barrier functions: Non-parametric paradigm to safety," *IEEE Access*, vol. 10, pp. 99823–99836, 2022.

[5] P. Jagtap, G. J. Pappas, and M. Zamani, "Control barrier functions for unknown nonlinear systems using gaussian processes," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3699–3704, 2020.

[6] V. Dhiman, M. J. Khojasteh, M. Franceschetti, and N. Atanasov, "Control barriers in bayesian learning of system dynamics," *IEEE Transactions on Automatic Control*, vol. 68, no. 1, pp. 214–229, 2023.

[7] C. Knuth, G. Chou, N. Ozay, and D. Berenson, "Planning with learned dynamics: Probabilistic guarantees on safety and reachability via lipschitz constants," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5129–5136, 2021.

[8] M. Bujarbaruah, C. Vallon, and F. Borrelli, "Learning to satisfy unknown constraints in iterative mpc," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 6204–6209, 2020.

[9] J. C. Shih, F. Meier, and A. Rai, "A framework for online updates to safe sets for uncertain dynamics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5994–6001, 2020.

[10] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes," in *53rd IEEE Conference on Decision and Control*, pp. 1424–1431, 2014.

[11] M. Bujarbaruah, S. H. Nair, and F. Borrelli, "A semi-definite programming approach to robust adaptive mpc under state dependent uncertainty," in *2020 European Control Conference (ECC)*, pp. 960–965, 2020.

[12] J. Köhler, P. Kötting, R. Soloperto, F. Allgöwer, and M. A. Müller, "A robust adaptive model predictive control framework for nonlinear uncertain systems," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8725–8749, 2021.

[13] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. a data-driven control framework," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1883–1896, 2018.

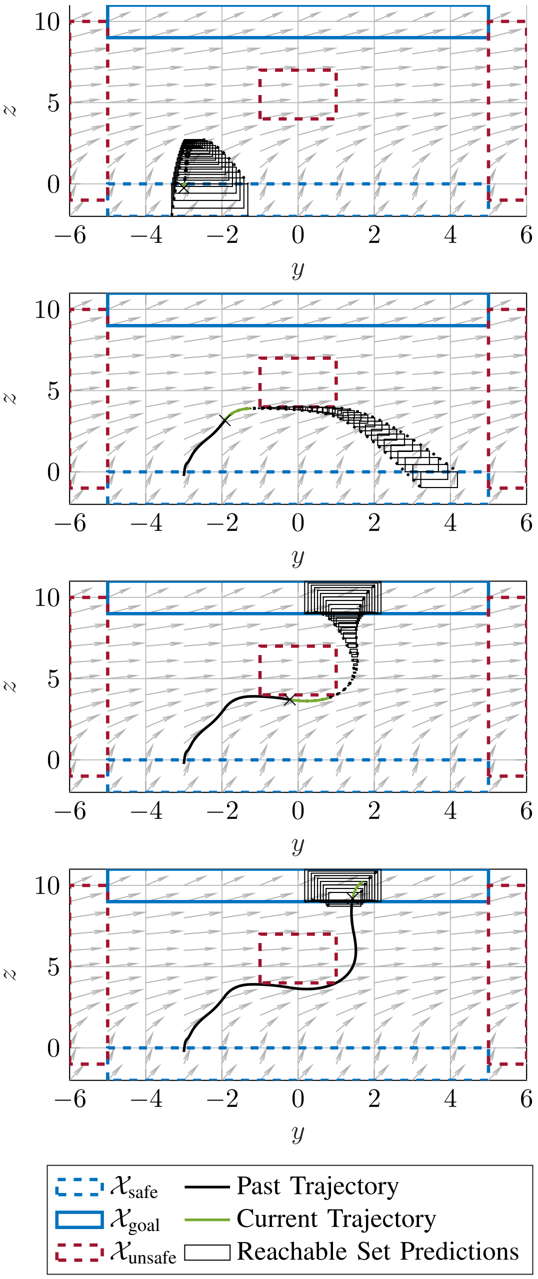

Fig. 2: Execution of the Planar Multirotor case study. The arrows denote the unknown wind force acting on the system.

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

[15] M. E. Cao, M. Bloch, and S. Coogan, "Estimating high probability reachable sets using gaussian processes," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 3881–3886, 2021.

[16] M. E. Cao, M. Bloch, and S. Coogan, "Efficient learning of hyperrectangular invariant sets using gaussian processes," *IEEE Open Journal of Control Systems*, pp. 1–14, 2022.

[17] M. Abate, M. Dutreix, and S. Coogan, "Tight decomposition functions for continuous-time mixed-monotone systems with disturbances," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 139–144, 2021.

[18] D. Angeli and E. D. Sontag, "Monotone control systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 10, pp. 1684–1698, 2003.