Uncertainty-aware Safe Exploratory Planning using Gaussian Process and Neural Control Contraction Metric

Dawei Sun DAWEIS2@ILLINOIS.EDU
University of Illinois Urbana-Champaign, Urbana, IL 61801, USA

Mohammad Javad Khojasteh MKHOJAST@MIT.EDU
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Shubhanshu Shekhar SHSHEKHA@ENG.UCSD.EDU
University of California San Diego, La Jolla, CA 92093, USA

Chuchu Fan CHUCHU@MIT.EDU
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Editors: A. Jadbabaie, J. Lygeros, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, M. N. Zeilinger

Abstract

In this paper, we consider the problem of using a robot to explore an environment with an unknown, state-dependent disturbance function while avoiding some forbidden areas. The goal of the robot is to safely collect observations of the disturbance and construct an accurate estimate of the underlying disturbance function. We use Gaussian Process (GP) to get an estimate of the disturbance from data with a high-confidence bound on the regression error. Furthermore, we use neural Contraction Metrics to derive a tracking controller and the corresponding high-confidence uncertainty tube around the nominal trajectory planned for the robot, based on the estimate of the disturbance. From the robustness of the Contraction Metric, error bound can be pre-computed and used by the motion planner such that the actual trajectory is guaranteed to be safe. As the robot collects more and more observations along its trajectory, the estimate of the disturbance becomes more and more accurate, which in turn improves the performance of the tracking controller and enlarges the free space that the robot can safely explore. We evaluate the proposed method using a carefully designed environment with a ground vehicle. Results show that with the proposed method the robot can thoroughly explore the environment safely and quickly.

Keywords: Gaussian Process, Control Contraction Metric, Learning Safe Exploratory Controller

1. Introduction

In the past few years, there has been an increasing interest in combining learning-based system identification and control theoretic techniques to accomplish complex tasks and control objectives (Deisenroth and Rasmussen, 2011; Dean et al., 2019; Sarkar et al., 2019; Coulson et al., 2019; Chen et al., 2018; Liu et al., 2020; Fan et al., 2020b; Chowdhary et al., 2014; Jagtap et al., 2020; Levine et al., 2016; Pan et al., 2018; Kahn et al., 2020; Thananjeyan et al., 2020; Srinivasan et al., 2020; Wabersich and Zeilinger, 2020a,b). Such a combination has shown to be able to reconcile the advantages of (deep) learned models which better represent data but are hard to be analyzed, and control techniques that are proven to work robustly but only on well-modeled control systems. Following this line of work, we study the problem of motion planning for robots to better learn the model uncertainties, while maintaining safety during the exploration process.

Consider the motivating example in Figure 1. The dynamics of a ground vehicle contain a disturbance term, which is an unknown function of the current position. For example, the friction factor will be different while the vehicle is driving on sand or grass. There are pools that the vehicle should avoid. To learn an accurate model of the vehicle, we have to safely drive the vehicle to every part of the environment and collect data about the friction while remaining safe. Similarly, safe exploratory planning is also a key yet challenging problem in many engineering domains such as Mars rover exploration as in (Ono et al., 2018; Ahmadi et al., 2020; Strader et al., 2020) and delivery drones as in (Cao et al., 2017; Berkenkamp and Schoellig, 2015).

We propose a novel framework to solve such uncertainty-aware safe exploratory problems by combining neural Control Contraction Metric (CCM) (cf. Sun et al., 2020) with Gaussian Process (GP). Let us use the above scenario as an example. The vehicle has dynamics $\dot{\mathbf{x}} = f(\mathbf{x}) + B(\mathbf{x})\mathbf{u} + d(\mathbf{x})$ where d is the unknown but bounded model error. The robot aims to approximate d with a GP model $\hat{d}$. Initially, the vehicle is only aware of its immediate surrounding environment. That is, it gets an initial guess of $\hat{d}$ and knows an upper bound on the variance of the estimation error $\|\hat{d} - d\|$ in a ball of radius ρ around itself. The algorithm then learns a robust control law using CCM such that in the ρ-ball, the vehicle can track any desired path $\mathbf{x}^*(t)$ with a tracking error $\mathcal{E}$. Therefore, a reference path $\mathbf{x}^*(t)$ is safe, if it is guaranteed to be at least $\mathcal{E}$ away from the forbidden areas. Then, at each time step, the vehicle will plan a safe path within the ρ-ball with the goal of obtaining more information about d. The algorithm will collect samples along the traveled path to continue updating the model $\hat{d}$, which in turn improves the performance of the tracking controller and characterizes more free space as safe to explore. This process will terminate when the free space has been fully explored in the sense that the estimation error $\|\hat{d} - d\|$ is uniformly below a threshold.

Figure 1: Motivating scenario: A ground vehicle needs to safely explore unseen environments to learn the effect of different terrains (in different colors) on its dynamics. The light blue regions are pools that the vehicle should avoid.

We evaluate the proposed method in the scenario as shown in Figure 1. We compare the proposed method with a baseline method where the error d is not learned on the fly but pre-estimated with hand-crafted bounds. Results show that with the proposed method the agent spends a shorter time exploring the environment and results in fewer collisions, which demonstrates the sample efficiency and safety guarantee of the proposed method. Moreover, by combining piece-wise linear paths and learned tracking controllers our method deals with nonlinear dynamics efficiently.

The major contributions of this paper are as follows: Firstly, we propose a framework for combining the GP model with neural contraction metric to safely collect data in an unknown environment. Secondly, we investigate the sample complexity of the GP regression and use the estimation variance of GP to determine the next point to explore, which improves the sample efficiency. Thirdly, we derive the criteria for determining when to update the tracking controller, which reduces unnecessary computation.

Related work Safe exploration has been studied in an extensive set of publications. Here we only mention a non-exhaustive list of related work. Liu et al. (2020) used neural networks to learn the

residual dynamics from data and use statistical learning theory to get a bound on the control performance. In (Nakka et al., 2020) the learned dynamics is projected into a finite-dimensional space using generalized polynomial chaos, and the trajectory planning problem is written as a convex optimization problem based on the approximated dynamics. Pravitra et al. (2020) used model predictive path integral control (MPPI) for motion planning, and used $L1$ adaptive control for handling the potential mismatch between the nominal and true dynamics. Koller et al. (2018) and Wabersich and Zeilinger (2020a,b) propose learning-based model predictive control (MPC) schemes that provide high-probability safety guarantees throughout the learning process using GPs. In MPC, nonlinear dynamics show up as constraints of the optimization problem, which reduce the efficiency of such methods for robots with complicated dynamics. Berkenkamp et al. (2016b) used GP to learn the unknown part of the dynamics and used Lyapunov functions to determine a region of attraction (ROA). With these guarantees, they provided an algorithm to actively and safely explore the state space to expand the ROA. Berkenkamp et al. (2016a) optimized the parameters of a controller while ensuring safety by modeling the underlying performance measure as a GP.

2. Problem setup and notations

We denote by $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$ the set of real and non-negative real numbers respectively. For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the notation $A \succ 0$ means A is positive definite. For a matrix-valued function $M(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$, its element-wise Lie derivative along a vector $\mathbf{v} \in \mathbb{R}^n$ is $\partial_{\mathbf{v}} M := \sum_i \mathbf{v}^{(i)} \frac{\partial M}{\partial \mathbf{x}^{(i)}}$. Unless otherwise stated, $\mathbf{x}^{(i)}$ denotes the i-th element of vector $\mathbf{x}$. For $A \in \mathbb{R}^{n\times n}$, we denote $A + A^\intercal$ by $\mathtt{sym}\,(A)$. The ball centered at $\mathbf{x}$ with radius ρ is denoted by $\mathcal{B}(\mathbf{x}, \rho)$.

We consider the problem of exploring an environment with unknown state-dependent disturbances and known obstacles. Assume that $\mathcal{X} \subseteq \mathbb{R}^n$ is the state space, $\mathcal{U} \subseteq \mathbb{R}^m$ is the input space, $\mathcal{D} \subseteq \mathbb{R}^n$ is the domain of disturbances, and $\mathcal{O} \subset \mathcal{X}$ is the region containing the obstacles. Let $\mathbf{x}(t) \in \mathcal{X}$ be the state of the agent; then the dynamics is given by

$$\dot{\mathbf{x}} = f\left(\mathbf{x}(t)\right) + B\left(\mathbf{x}(t)\right)\mathbf{u}(t) + d(\mathbf{x}(t)), \tag{1}$$

where dynamics functions $f : \mathcal{X} \mapsto \mathbb{R}^n$, $B : \mathcal{X} \mapsto \mathbb{R}^{n\times m}$ are smooth, $\mathbf{u} : \mathbb{R}_{\geq 0} \mapsto \mathcal{U}$ is the control input, and $d : \mathcal{X} \mapsto \mathcal{D}$ is a disturbance function. The functions f and B are assumed to be *known*, whereas d represents the *unknown* part of the dynamics, caused by discrepancies between the model and the real dynamics or by disturbances in the environment, such as drag or friction.

We assume that the agent can observe the disturbance $d(\mathbf{x})$ after it has visited a small neighborhood around state $\mathbf{x}$, that is, it only collects disturbance observations around its trajectory. The observations are noisy with i.i.d. additive Gaussian noise with zero mean and covariance $s^2 I_n$. The goal of the agent is to safely explore the environment to establish an accurate estimate $\hat{d}(\cdot)$ of the disturbance map $d(\cdot)$. At the same time, it should make use of the current estimate to explore the environment while avoiding the obstacles. Let $e(\mathbf{x}) = d(\mathbf{x}) - \hat{d}(\mathbf{x})$ be the estimation error. Formally, the overall goal is to find an estimate $\hat{d}$ such that $\|e(\mathbf{x})\|_2 \leq \psi_{\mathtt{th}}$ for all $\mathbf{x} \in \mathcal{X}$ in the free-space, and some given threshold $\psi_{\mathtt{th}} > 0$, while ensuring safety during exploration.

To derive analytical results, we need to limit the class of possible uncertainty map d. In particular, we work in a Bayesian framework and assume that d is a sample from a multivariate Gaussian process with zero mean and known covariance function (or kernel) $\mathcal{K}(\cdot,\cdot)$ (cf. Srinivas et al., 2012; Lederer et al., 2019b). The choice of the kernel is problem dependent; see, e.g., (Williams and

Rasmussen, 2006) for a review of common kernel choices. In addition we assume that the kernel $\mathcal{K}$ satisfies the following properties:

Assumption 1 *(i) $\mathcal{K}$ is* isotropic, *i.e., $\mathcal{K}(\mathbf{x}, \mathbf{y})$ depends on $\mathbf{x}$ and $\mathbf{y}$ only through $\|\mathbf{x}-\mathbf{y}\|$ and hence in the sequel we will also overload the notation and use $\mathcal{K}(\|\mathbf{x}-\mathbf{y}\|)$ to denote $\mathcal{K}(\mathbf{x}, \mathbf{y})$; (ii) There exist constants $C_K > 0$ and $\omega \in (0, 1]$ (depending on $\mathcal{K}$) such that we have $\sqrt{2(\mathcal{K}(0) - \mathcal{K}(r))} \leq C_K r^\omega$ for all $r > 0$. This condition is satisfied for most of the commonly used covariance functions such as Squared-Exponential (SE) and Matérn kernels (with half-integer smoothness) as noted by Shekhar and Javidi (2018); (iii) There exist constants $a_1, a_2, L > 0$, such that $\mathbb{P}\left(\{\sup_{\mathbf{x}\in\mathcal{X}} |\partial d^{(j)}(\mathbf{x})/\partial \mathbf{x}^{(j)}| < L\}\right) \geq 1 - a_1 n e^{-L^2/a_2^2}$ for $j = 1, \ldots, n$. Note that this assumption was employed in (Srinivas et al., 2010) to apply the GP based analysis to continuous domains $\mathcal{X}$.*

Overview of the method. The proposed method consists of three major components. (i) Gaussian Processes (GP) are used to learn the disturbance from observations and give the corresponding high-probability bound on the estimation error, which will be elaborated in Sec. 3; (ii) With the estimate of the disturbance, we apply the method proposed by Sun et al. (2020) to learn a tracking controller for the approximated dynamics. Using this controller, the system can track any nominal trajectory with bounded tracking error, which will be elaborated in Sec. 4; (iii) An uncertainty-aware data acquisition algorithm is used to ensure that the agent always visits the most informative points such that the estimation error can be efficiently reduced as the agent collects data around its trajectory. Also, a simple planning strategy is used to plan nominal trajectories in the environment considering the pre-computed tracking error bound such that the motion of the agent is guaranteed to be safe. The overall exploration algorithm will be shown in Sec. 5.

3. Gaussian process regression and sample complexity

We use GP as our Bayesian inference tool to estimate state-dependent disturbances d. Following (Berkenkamp et al., 2017) we develop a unidimensional GP regression for each dimension $d^{(i)}$, where $i = 1, \ldots, n$. Recall that the observations are disturbed with i.i.d. additive Gaussian noise with zero mean and covariance $s^2 I_n$. The training observations, for the i-th coordinate, at the sampling points $\mathbf{x}_{[N]} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\intercal$, are denoted by $\mathbf{y}_{i,[N]}$, which is the noisy version of the vector $[d^{(i)}(\mathbf{x}_1), \ldots, d^{(i)}(\mathbf{x}_N)]^\intercal$. Let κ_i be the kernel function for the i-th coordinate. The posterior distribution is again Gaussian and can be computed at the query test point $\mathbf{x}_*$, as follows (cf. Williams and Rasmussen, 2006).

$$
\begin{aligned}
&d^{(i)}(\mathbf{x}_*) \sim \mathcal{N}\big(\mu_N^{(i)}(\mathbf{x}_*),\ \sigma_N^{(i)}(\mathbf{x}_*)\big)\\
&\mu_N^{(i)}(\mathbf{x}_*) = K_i(\mathbf{x}_*, \mathbf{x}_{[N]})^\intercal (K_i(\mathbf{x}_{[N]}, \mathbf{x}_{[N]}) + s^2 I_N)^{-1} \mathbf{y}_{i,[N]}\\
&\sigma_N^{(i)}(\mathbf{x}_*) = \kappa_i(\mathbf{x}_*, \mathbf{x}_*) - K_i(\mathbf{x}_*, \mathbf{x}_{[N]})^\intercal (K_i(\mathbf{x}_{[N]}, \mathbf{x}_{[N]}) + s^2 I_N)^{-1} K_i(\mathbf{x}_*, \mathbf{x}_{[N]}),
\end{aligned}
$$

where $K_i(\mathbf{x}_{[N]}, \mathbf{x}_{[N]}) \in \mathbb{R}^{N\times N}$ with $[K_i(\mathbf{x}_{[N]}, \mathbf{x}_{[N]})]_{j,k} = \kappa_i(\mathbf{x}_j, \mathbf{x}_k)$, and $K_i(\mathbf{x}_*, \mathbf{x}_{[N]}) \in \mathbb{R}^{1\times N}$ with $[K_i(\mathbf{x}_*, \mathbf{x}_{[N]})]_j = \kappa_i(\mathbf{x}_*, \mathbf{x}_j)$. We estimate $d(\mathbf{x})$ with the mean of the GP posteriors. That is, $\hat{d}(\mathbf{x}) = \mu_N(\mathbf{x})$, where $\mu_N(\mathbf{x}) := [\mu_N^{(1)}(\mathbf{x}), \ldots, \mu_N^{(n)}(\mathbf{x})]^\intercal$.

3.1. Sample-dependent high confidence error bound

In this section, we derive a high probability upper bound on the number of observations required to ensure that the estimate error $e(x)$ can be made smaller than some prescribed value $\psi_{\texttt{th}}$ within a neighborhood of radius ρ around some given point $\mathbf{o}$.

We begin by stating an assumption on the sampling distribution of the agent, which formalizes the requirement that the agent can gather sufficient information within its neighborhood. This assumption is necessary for our main result of this section, Theorem 1, as our goal is to get uniformly good estimates of d at every point in the neighborhood.

Assumption 2 *We assume that when the agent is situated at some point $\boldsymbol{o} \in \mathcal{X}$, it can draw samples in a neighborhood $\mathcal{B}(\boldsymbol{o}, \rho)$ around the point according to a sampling distribution Q with support $\mathcal{B}(\boldsymbol{o}, \rho)$, which admits a density q satisfying the property $\underline{c} \leq q(x)$ for all $x \in \mathcal{B}(\boldsymbol{o}, \rho)$ for a positive constant $\underline{c} > 0$. Note that a special case of Q is the uniform distribution which admits a constant density $q(x) = \frac{1}{Vol(\mathcal{B}(\boldsymbol{o},\rho))}$.*

We can now state the main result of this section which provides a bound on the number of observations needed to ensure a uniformly good estimate of the model error function d.

Theorem 1 *Suppose the following conditions are satisfied: 1) The model error d in (1) is a sample from a zero-mean GP with the covariance function $\mathcal{K}$ satisfying Assumption 1, and 2) The agent can make observations in its neighborhood according to a sampling distribution Q satisfying Assumption 2. Then the number of observations $N(\rho, \delta)$, drawn according to the sampling distribution Q, that are required by the agent in a ball of radius ρ around some point $\boldsymbol{o}$ to ensure that $\|e(x)\|_2 \leq \psi$ for all $x \in \mathcal{B}(\boldsymbol{o}, \rho)$ with probability at least $1 - \delta$ is $\widetilde{\mathcal{O}}\left(\max\left\{2\sqrt{n}\psi^{-1}, \ \frac{\psi^{-2n/\omega}}{\underline{c}^2}, \ \frac{s^2\psi^{-(2\omega+n)/\omega}}{\underline{c}}\right\}\right)$ where the $\widetilde{\mathcal{O}}(\cdot)$ notation suppresses the poly-logarithmic factors of $\log(1/\delta)$ and $\log(1/\psi)$.*

Proof (sketch) Full proof can be found in Appendix A.

- First, following the proof of (Srinivas et al., 2010, Lemma 5.6), we first introduce a high probability event Ω_1 such that for all points x in a fine discretization (denoted by H) of $\mathcal{B}(\mathbf{o}, \rho)$, we have $|\mu_t^{(j)}(x) - d^{(j)}(x)| \leq \beta_N \sigma_{t-1}^{(j)}(x)$ for $1 \leq j \leq n$. By making the discretization H fine enough, we can ensure sufficiently accurate estimate of d at every point of $\mathcal{X}$ by appealing to property (iii) in Assumption 1.
- Next, we note that by using (Shekhar and Javidi, 2018, Prop. 3), to ensure uniformly tight estimate, we need to ensure that every point $x \in \mathcal{B}(\mathbf{o}, \rho)$ has sufficiently many samples in a ball of radius r_0 around it, for an appropriate choice of r_0. To achieve this, we consider a fixed $r_0/2$-covering of $\mathcal{B}(\mathbf{o}, \rho)$, denoted by E, and find N large enough which ensures that a $r_0/2$ neighborhood of every point in E has sufficiently many samples drawn according to Q.

■

Remark 2 *Note that our proof of Theorem 1 proceeds by first obtaining a uniform deviation bound by controlling the deviation on the elements of a sufficiently fine discretization H of the ball $\mathcal{B}(\boldsymbol{o}, \rho)$. Alternatively, we could also have employed the uniform error bounds derived in (Lederer et al., 2019b, Theorem 3.1) for this task. However, our approach leads to a slightly easier path to obtain the sample complexity, i.e., finding the value of N which ensures that the error is smaller than some given quantity ψ. Performing this "inversion" with the more general bounds derived by Lederer et al. (2019a); Gahlawat et al. (2020) may be more involved.*

4. Learning-based tracking controller and tracking error

In the last section, we constructed a high confidence bound on the estimation error of the disturbance. In this section, we show how to learn a tracking controller with high confidence bound on the tracking error.

Contraction theory (Lohmiller and Slotine, 1998) analyzes the incremental stability of a system by considering the evolution of the distance between any pairs of arbitrarily close neighboring trajectories. The existence of a Control Contraction Metric (CCM) (Manchester and Slotine, 2017) ensures the existence of a tracking controller that can drive the system to any nominal trajectories.

We apply the method proposed by Sun et al. (2020) to jointly learn a tracking controller and a contraction metric function for the dynamics with the estimate of the disturbance, i.e. $\dot{\mathbf{x}} = \hat{f}(\mathbf{x}(t)) + B(\mathbf{x}(t))\,\mathbf{u}(t)$, where $\hat{f}(\mathbf{x}) = f(\mathbf{x}) + \hat{d}(\mathbf{x})$. As shown by Sun et al. (2020), the learned metric $M(\cdot)$ is just a mapping from the state $\mathbf{x}$ to an $n \times n$ positive definite matrix. The learned tracking controller is a feedback controller of the form $\mathbf{u}(\mathbf{x}, \mathbf{x}^*, \mathbf{u}^*)$, where $\mathbf{x}$ is the current state and $\mathbf{x}^*, \mathbf{u}^*$ are the nominal state and control input. We want to find a metric function $M(\cdot)$ and a feedback controller $\mathbf{u}(\cdot)$ satisfying that for all $\mathbf{x} \in \mathcal{X}$, $\mathbf{x}^* \in \mathcal{X}$, $\mathbf{u}^* \in \mathcal{U}$, and some $\lambda > 0$,

$$\dot{M} + \mathtt{sym}\left(M(A + BK)\right) + 2\lambda M \prec 0, \tag{2}$$

where $A := \frac{\partial \hat{f}}{\partial \mathbf{x}} + \sum_{j=1}^{m} \mathbf{u}^{(j)} \frac{\partial b_j}{\partial \mathbf{x}}$, b_j is the j-th column of B, $\mathbf{u}^{(j)}$ is the j-th element of $\mathbf{u}$, $K = \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$, and $\dot{M}$ is the derivative of $M(\mathbf{x}(t))$ w.r.t. time. We refer the readers to (Sun et al., 2020) for more details. Note that the above formulation uses the estimated dynamics by plugging $\hat{d}(\mathbf{x})$ in (1). The following theorem shows that when applied to the real dynamics, the tracking error of the learned controller is still bounded.

Theorem 3 (Robustness to dynamics error, Sun et al. 2020) *Given M and $\mathbf{u}$ satisfying inequality (2), since $M(\mathbf{x})$ is positive definite, there exist $\overline{m} \geq \underline{m} > 0$ such that $\underline{m}\mathbf{I} \preceq M(\mathbf{x}) \preceq \overline{m}\mathbf{I}$ for all $\mathbf{x}$. Assume that error of the dynamics is bounded as $\|e(\mathbf{x})\| \leq \psi$ for all $\mathbf{x}$ and some $\psi > 0$. Now considering the trajectory $\mathbf{x}(t)$ of the closed-loop system, the distance between $\mathbf{x}(t)$ and any given reference $\mathbf{x}^*(t)$ is bounded as $\|\mathbf{x}(t) - \mathbf{x}^*(t)\|_2 \leq \frac{R_0}{\sqrt{\underline{m}}} e^{-\lambda t} + \sqrt{\frac{\overline{m}}{\underline{m}}} \cdot \frac{\psi}{\lambda}(1 - e^{-\lambda t})$, where R_0 is the Riemannian distance between $\mathbf{x}(0)$ and $\mathbf{x}^*(0)$ under metric M.*

Moreover, if $\mathbf{x}(0) = \mathbf{x}^*(0)$, then the Riemannian distance $R_0 = 0$. This is usually the case since the reference trajectory planned by the open-loop motion planner exactly starts from the current state of the agent. Thus, the tracking error of the learned controller is upper bounded by $\mathcal{E} = \sqrt{\frac{\overline{m}}{\underline{m}}} \frac{\psi}{\lambda}$. If we can ensure that the planned nominal trajectory is at least $\mathcal{E}$ away from the obstacles, then the realized trajectory is guaranteed to be safe. As will be shown in Sec. 5, this is equivalent to bloating the obstacles by $\mathcal{E}$ before planning. The following corollary immediately follows from Theorem 1 and Theorem 3.

Corollary 4 *Suppose that a ball $\mathcal{B}(\mathbf{o}, \rho)$ contains N samples such that N, ρ, δ, ψ satisfy the condition of Theorem 1 for some δ and ψ. If there exists a controller and metric satisfying the CCM condition (2) and the motion of the closed-loop system is restricted in $\mathcal{B}(\mathbf{o}, \rho)$, then the tracking error is less than or equal to $\mathcal{E} = \sqrt{\frac{\overline{m}}{\underline{m}}} \cdot \frac{\psi}{\lambda}$ with probability at least $1 - \delta$.*

Retraining of the controller. As mentioned before, the agent gradually collects more and more observations and keeps improving the estimate $\hat{d}$. In this case, we might have to learn a new controller $\mathbf{u}$ and a new contraction metric M such that condition (2) still holds. Retraining of this controller is expensive, and thus we use the following method to reduce the number of retrainings. The basic idea is to impose some robust margin on condition (2) during training, such that the learned metric and controller are robust to the change of $\hat{d}$ to some extent. Specifically, instead of condition (2), we use the following condition for learning,

$$\dot{M} + \mathtt{sym}\left(M(A+BK)\right) + 2\lambda M \prec -\mathcal{M}\mathbf{I}, \tag{3}$$

where $\mathcal{M} > 0$ is the margin for robustness. Intuitively, if we impose the above condition, small changes in $\hat{d}$ will not lead to a violation of condition (2). Retraining is only needed when the change in $\hat{d}$ crosses a certain threshold. Formally, we have the following theorem.

Theorem 5 *Consider two estimates $\hat{d}_1$ and $\hat{d}_2$ and their difference $\mathcal{R} = \hat{d}_1 - \hat{d}_2$. If the metric M and controller $\mathbf{u}$ satisfy the robust condition (3) for the estimate $\hat{d}_1$ and the difference $\mathcal{R}$ satisfies the following condition for all $\mathbf{x}$,*

$$\|\partial_{\mathcal{R}} M + \mathtt{sym}\left(M\mathcal{R}\right)\|_2 \leq \mathcal{M}, \tag{4}$$

then the original condition (2) is also satisfied for the estimate $\hat{d}_2$.

The proof can be found in Appendix B. In practice, we evaluate condition (4) only in the region of our interest instead of the whole state space. Moreover, evaluating whether condition (4) holds on an uncountable set is hard. Instead, we randomly sample a number of points from the set and say condition (4) holds for the whole set only if it holds for all sampled points with a robust margin determined by the Lipschitz constant of the LHS of condition (4) (cf. Sun et al., 2020, Sec. 3.2).

5. Algorithm

The overall framework is shown in Algorithm 1. Several components are explained in order.

Compute the estimation error. In Algorithm 1, we need to determine the estimation error ψ in a ball $\mathcal{B}(\mathbf{o}, \rho)$ given the current observations and the confidence level δ. Theorem 1 provides a high probability bound on the number of samples needed to ensure uniformly good estimate within a ball $\mathcal{B}(\mathbf{o}, \rho)$. Based on this theorem, we now present a practical heuristic to compute an upper bound on estimation error, which in turn provides a stopping rule for the sampling. We proceed as follows. We use black-box optimization to find the maximizer of $\sigma_N^{(j)}$ over the domain $\mathcal{B}(\mathbf{o}, \rho)$ for all $1 \leq j \leq n$ which we denote by $\tilde{\sigma}_N^{(j)}$. As we mentioned in the proof sketch of Theorem 1 the absolute value of the estimation error for the j-th coordinate is bounded by $\beta_N \sigma_{t-1}^{(j)}(x)$. Hence, we stop the sampling if $\beta_N \sqrt{\sum_{j=1}^{n} [\tilde{\sigma}_N^{(j)}]^2}$, which represents the upper bound on the 2-norm of the total estimation error, is smaller that $\psi_{\mathtt{th}}$. Here, β_N is a quantity defined in Appendix A.

Find the next point to visit. At each time step, the agent has to determine the next point to visit and collect observations around its trajectory. To make the exploration efficient, the next point to visit must be informative. Therefore, we choose the one with highest estimate variance. Formally,

$$\mathbf{g} = \arg\max_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^{n} \left(\kappa_i(\mathbf{x}, \mathbf{x}) - K_i(\mathbf{x}, \mathbf{x}_{[N]})^{\intercal} (K_i(\mathbf{x}_{[N]}, \mathbf{x}_{[N]}) + s^2 I_N)^{-1} K_i(\mathbf{x}, \mathbf{x}_{[N]})\right). \tag{5}$$

Algorithm 1: Safe exploration.

Input: Initial state $\mathbf{x}$; Obstacles $\mathcal{O} \subset \mathcal{X}$;
Input: Error tolerance $\psi_{\texttt{th}}$; Confidence level δ;
Output: Final estimate $\hat{d}$;
Function `Plan(`$\mathbf{x}$, $\mathbf{g}$, $\mathcal{E}$`)`**:**
  Data: current state $\mathbf{x}$; goal $\mathbf{g}$; bloating factor $\mathcal{E}$;
  Bloating obstacles: $\tilde{\mathcal{O}} = \mathcal{O} \bigoplus \mathcal{B}(0, \mathcal{E})$;
  Plan from $\mathbf{x}$ to $\mathbf{g}$ while avoiding $\tilde{\mathcal{O}}$;
while *not satisfied* **do**
  Find next goal $\mathbf{g}$ to visit using Eq. (5);
  $\rho = \rho_0$; `path` $=$ `null`;
  while `path` *is* `null` **do**
    Compute $\mathcal{E}$ in $\mathcal{B}(\mathbf{x}, \rho)$;
    `path` = `Plan(`$\mathbf{x}$, $\mathbf{g}$, $\mathcal{E}$`)`;
    Decrease ρ;
  end
  Move along `path` until reaching the boundary of $\mathcal{B}(\mathbf{x}, \rho)$;
  Enlarge the observation set and update $\hat{d}$;
  Retrain the controller if needed;
end

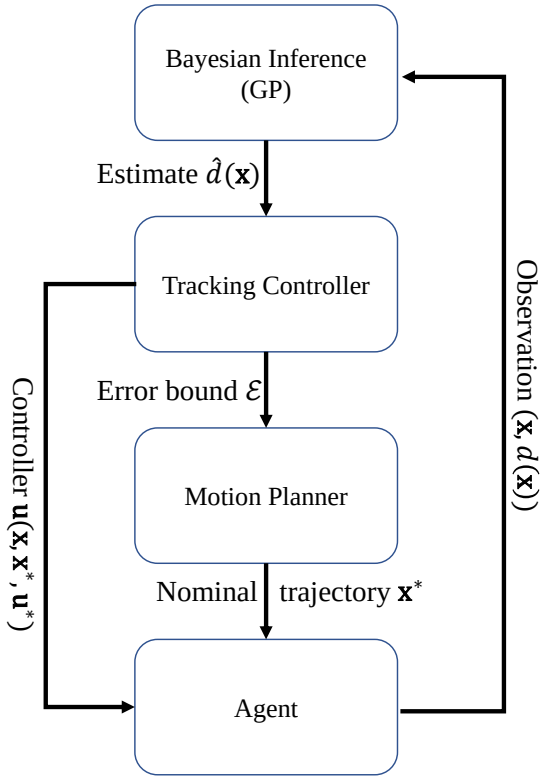

Figure 2: Diagram of the proposed method.

Planning a feasible path. After determining the next point to visit, the agent has to plan a nominal trajectory feasible for the nominal dynamics (i.e. without considering $e(\mathbf{x})$) such that controlled by the learned controller in Sec. 4, the agent can safely track this nominal trajectory and reach the goal. As mentioned in Sec. 4, the distance between the actual trajectory and the nominal one is bounded by $\mathcal{E}$. Therefore, we first bloat the obstacles with the error bound $\mathcal{E}$: Let the obstacles be $\mathcal{O} \in \mathcal{X}$; the bloated obstacles are just $\mathcal{O} \bigoplus \mathcal{B}(0, \mathcal{E})$, where $\bigoplus$ denotes the Minkowski summation and $\mathcal{B}(0, \mathcal{E})$ denotes the ball centered at the origin with radius $\mathcal{E}$. Then, a nominal trajectory to the goal is planned while avoiding the bloated obstacles. Any motion planner could suffice, e.g. (Vitus et al., 2008; Fan et al., 2020a), and we adopt RRT* (Karaman and Frazzoli, 2011). RRT* is used to generate a piece-wise linear path. However, this path is usually not feasible for the agent. Thus, an additional step is required to generate a feasible trajectory and the corresponding reference control input. To this end, a simple linear feedback controller is used to track the planned piece-wise linear path (again, without considering $e(\mathbf{x})$). The trajectory generated by the agent controlled by the simple controller will be used as the reference, i.e. $\mathbf{x}^*(t)$ and $\mathbf{u}^*(t)$ for the tracking controller. Due to the tracking error introduced by the simple controller, $\mathbf{x}^*$ may be unsafe. If that happens, we will bloat the obstacles a bit more and repeat planning until we find a safe $\mathbf{x}^*$. However, in the experiments we found that this was very rarely needed.

Putting it all together. At each time step, the agent first determines the next point to visit. Then, it initializes the radius $\rho = \rho_0$ and computes the upper bound on the estimation error and the corresponding tracking error $\mathcal{E}$ in the ball $\mathcal{B}(\mathbf{x}, \rho)$. Using $\mathcal{E}$, the agent searches for a safe path to the goal. If it failed to find such a path, then ρ is decreased a bit and the above process is repeated until

a safe path is found. Then, controlled by the learned controller, the agent moves along the path and collects new observations on the disturbance until it reaches the boundary of the ball $\mathcal{B}(\mathbf{x}, \rho)$. Then, GP is invoked to update the estimate $\hat{d}$ using the new observations. After that, we might retrain the controller if needed as shown in Sec. 4. The exploration will terminate once we have collect enough samples such that $\|e(\mathbf{x})\|_2 \leq \psi_{\texttt{th}}$ for all $\mathbf{x} \in \mathcal{X} \setminus \mathcal{O}$ with probability at least $1 - \delta$.

6. Experimental evaluations

In order to evaluate the proposed safe-exploration framework, we designed a scenario as shown in Fig. 1. Several components of the scenario are explained in order.

Dynamics. We adopted the Dubins car model for the agent. The state of the system is $\mathbf{x} := [p_x, p_y, \theta, v, \omega]^\intercal$, where (p_x, p_y) the position of the car, θ the heading angle, v the velocity, and ω is the angular velocity. The control input is $\mathbf{u} := [f, \tau]^\intercal$, where f is the force and τ is the torque. The dynamics of the car is

$$\dot{\mathbf{x}} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \\ -0.4v \\ -0.4\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} + d(\mathbf{x}).$$

Workspace and obstacles. In the experiments, we want the agent to explore a square region $[0, 10] \times [0, 10]$ on the 2D plane. We randomly generate 10 obstacles, which are shown in Fig. 1.

Disturbance function. The disturbance $d(\mathbf{x})$ is a function of the first two elements of $\mathbf{x}$ and models the effect caused by the ground at position (p_x, p_y). We use an image from (Sturtevant, 2012) as the terrain map. In order to define the 5-dimensional disturbance based on the color values of the image, we use a 5×3 projection matrix P to map the color space to disturbance space. The disturbance at (p_x, p_y) is obtained by multiplying P and the corresponding RGB color value.

Simulation. At the beginning of the simulation, we assign a random initial position to the agent such that it is safe initially. Then, the trajectories are simulated with a constant time step $\Delta t = 0.01$ s. At each time step, we check the safety of the agent. If the distance to an obstacle is less than a threshold $\texttt{thr} > 0$, then it is said to be unsafe. In the following experiments, we set $\texttt{thr}$ to 0.1 meters. The goal of the agent is to collect observations on the disturbance to construct an estimate $\hat{d}$ of the actual disturbance map d and maintain safety in this process.

Metrics for evaluation. We use the following metrics for comparison: 1. *Unsafe* is the percentage of iterations at which the agent is unsafe; 2. *Travel time* is the time needed for exploring the workspace; 3. *Tracking error* is the average tracking error, which is the distance between the actual trajectory and its nominal trajectory averaged over all time steps. Moreover, all the metrics reported in this section are averaged over 5 runs.

Comparison with the baseline method. The baseline method is a variant of Algorithm 1. The baseline method does not make use of the current estimate $\hat{d}$ to retrain the controller and compute the high-probability tracking error $\mathcal{E}$. Instead, $\mathcal{E}$ is set to be a constant. In the experiments, we set $\mathcal{E} = 0$, 0.1, or 0.3. We also tried to use larger $\mathcal{E}$, e.g. $\mathcal{E} = 0.6$, however, in that case, the bloated obstacles blocked the free space, which makes it impossible to finish the exploration. For all the

Table 1: Comparison with the baseline method.

Method	Unsafe (%)	Travel time (s)	Tracking error
Algorithm 1	0.3	236	0.051
Baseline ($\mathcal{E} = 0$)	10.3	208	0.243
Baseline ($\mathcal{E} = 0.1$)	5.1	314	0.221
Baseline ($\mathcal{E} = 0.3$)	4.0	515	0.230

methods we set $\rho_0 = 1$, $\psi_{\texttt{th}} = 0.1$, and $\delta = 0.05$. The results are shown in Table 1. Compared to the baseline methods, the proposed method results in higher safety and shorter travel time, which demonstrates the sufficiency of the proposed method. Further illustration of the experimental results can be found in Appendix C.

7. Discussion and Future Work

In this paper, we consider the problem of using a robot to safely explore an unknown environment and propose a framework where GP and contraction metric are combined to drive the robot efficiently and safely in the environment. Results on a ground vehicle model verify the efficiency of the proposed safe exploration framework. There are several interesting directions for future research.

- We developed an independent GP regression for each coordinate. For cases where strong correlations exist between components, we could employ Matrix-Variate GP, as in existing works such as (Khojasteh et al., 2020; Louizos and Welling, 2016; Cheng et al., 2020).
- In this work, we assume the unknown part of the dynamics is a sample from GP. Alternatively, depending on specific applications and the available prior knowledge, it may be more suitable to apply other estimation techniques such as random forests, neural networks or counter-example guided learning (Chen et al., 2020).
- In this work, we plan the agent action toward the point with the highest estimate variance (5) and empirically showed its benefits. An important question to investigate for future work is whether there exist planning strategies that can provably improve upon our method. Ideas from the literature on active learning (Buisson-Fenet et al., 2020; Capone et al., 2020; Lew et al., 2020; Nakka et al., 2020) may be useful in designing such optimal strategies.

Acknowledgments The authors acknowledge support from the DARPA Assured Autonomy under contract FA8750-19-C-0089 and from the Defense Science and Technology Agency in Singapore. The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense, the U.S. Government, DSTA Singapore, or the Singapore Government.

References

Mohamadreza Ahmadi, Masahiro Ono, Michel D Ingham, Richard M Murray, and Aaron D Ames. Risk-averse planning under uncertainty. In *2020 American Control Conference (ACC)*, pages 3305–3312. IEEE, 2020.

Felix Berkenkamp and Angela P Schoellig. Safe and robust learning control with gaussian processes. In *2015 European Control Conference (ECC)*, pages 2496–2501. IEEE, 2015.

Felix Berkenkamp, Riccardo Moriconi, Angela P Schoellig, and Andreas Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4661–4666. IEEE, 2016a.

Felix Berkenkamp, Angela P Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 491–496. IEEE, 2016b.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in neural information processing systems*, pages 908–918, 2017.

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

Gang Cao, Edmund M-K Lai, and Fakhrul Alam. Gaussian process model predictive control of an unmanned quadrotor. *Journal of Intelligent & Robotic Systems*, 88(1):147–162, 2017.

Alexandre Capone, Gerrit Noske, Jonas Umlauft, Thomas Beckers, Armin Lederer, and Sandra Hirche. Localized active learning of gaussian process state space models. In *Learning for Dynamics and Control*, pages 490–499. PMLR, 2020.

Steven Chen, Kelsey Saulnier, Nikolay Atanasov, Daniel D Lee, Vijay Kumar, George J Pappas, and Manfred Morari. Approximating explicit model predictive control using constrained neural networks. In *2018 Annual American Control Conference (ACC)*, pages 1520–1527. IEEE, 2018.

Yuxiao Chen, Sumanth Dathathri, Tung Phan-Minh, and Richard M Murray. Counter-example guided learning of bounds on environment behavior. *arXiv preprint arXiv:2001.07233*, 2020.

Richard Cheng, Mohammad Javad Khojasteh, Aaron D Ames, and Joel W Burdick. Safe multi-agent interaction through robust control barrier functions with learned uncertainties. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 777–783. IEEE, 2020.

Girish Chowdhary, Hassan A Kingravi, Jonathan P How, and Patricio A Vela. Bayesian nonparametric adaptive control using Gaussian processes. *IEEE transactions on neural networks and learning systems*, 26(3):537–550, 2014.

Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control: in the shallows of the deepc. In *2019 18th European Control Conference (ECC)*, pages 307–312. IEEE, 2019.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, Aug 2019.

Marc Deisenroth and Carl E Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

Chuchu Fan and Sayan Mitra. Bounded verification with on-the-fly discrepancy computation. *arXiv preprint arXiv:1502.01801*, 2015.

Chuchu Fan, Kristina Miller, and Sayan Mitra. Fast and guaranteed safe controller synthesis for nonlinear vehicle models. In Shuvendu K. Lahiri and Chao Wang, editors, *Computer Aided Verification*, pages 629–652, Cham, 2020a. Springer International Publishing. ISBN 978-3-030-53288-8.

David D Fan, Ali-akbar Agha-mohammadi, and Evangelos A Theodorou. Deep learning tubes for tube mpc. *arXiv preprint arXiv:2002.01587*, 2020b.

Aditya Gahlawat, Pan Zhao, Andrew Patterson, Naira Hovakimyan, and Evangelos Theodorou. L1-GP: L1 adaptive control with Bayesian learning. 2020.

Pushpak Jagtap, George J Pappas, and Majid Zamani. Control barrier functions for unknown nonlinear systems using gaussian processes. *arXiv preprint arXiv:2010.05818*, 2020.

Gregory Kahn, Pieter Abbeel, and Sergey Levine. Badgr: An autonomous self-supervised learning-based navigation system. *arXiv preprint arXiv:2002.05700*, 2020.

Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

Mohammad Javad Khojasteh, Vikas Dhiman, Massimo Franceschetti, and Nikolay Atanasov. Probabilistic safety constraints for learned high relative degree system dynamics. In *Learning for Dynamics and Control*, pages 781–792, 2020.

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. IEEE, 2018.

Armin Lederer, Jonas Umlauft, and Sandra Hirche. Posterior variance analysis of gaussian processes with application to average learning curves. *arXiv preprint arXiv:1906.01404*, 2019a.

Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform error bounds for gaussian process regression with application to safe control. In *Advances in Neural Information Processing Systems*, pages 659–669, 2019b.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Thomas Lew, Apoorva Sharma, James Harrison, and Marco Pavone. Safe model-based meta-reinforcement learning: A sequential exploration-exploitation framework. *arXiv preprint arXiv:2008.11700*, 2020.

Anqi Liu, Guanya Shi, Soon-Jo Chung, Anima Anandkumar, and Yisong Yue. Robust regression for safe exploration in control. In *Learning for Dynamics and Control*, pages 608–619, 2020.

Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016.

Ian R Manchester and Jean-Jacques E Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control*, 2017.

Yashwanth Kumar Nakka, Anqi Liu, Guanya Shi, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems. *arXiv preprint arXiv:2005.04374*, 2020.

Masahiro Ono, Matthew Heverly, Brandon Rothrock, Eduardo Almeida, Fred Calef, Tariq Soliman, Nathan Williams, Hallie Gengl, Takuto Ishimatsu, Austin Nicholas, et al. Mars 2020 site-specific mission performance analysis: Part 2. surface traversability. In *2018 AIAA SPACE and Astronautics Forum and Exposition*, page 5419, 2018.

Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntak Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile autonomous driving using end-to-end deep imitation learning. In *Robotics: science and systems*, 2018.

Jintasit Pravitra, Kasey A Ackerman, Chengyu Cao, Naira Hovakimyan, and Evangelos A Theodorou. L1-adaptive MPPI architecture for robust and agile control of multirotors. *arXiv preprint arXiv:2004.00152*, 2020.

Tuhin Sarkar, Alexander Rakhlin, and Munther A Dahleh. Finite-time system identification for partially observed LTI systems of unknown order. *arXiv preprint arXiv:1902.01848*, 2019.

Shubhanshu Shekhar and Tara Javidi. Gaussian process bandits with adaptive discretization. *Electronic Journal of Statistics*, 12(2):3829–3874, 2018.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *In International Conference on Machine Learning*, 2010.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

Mohit Srinivasan, Amogh Dabholkar, Samuel Coogan, and Patricio Vela. Synthesis of control barrier functions using a supervised machine learning approach. *arXiv preprint arXiv:2003.04950*, 2020.

Jared Strader, Kyohei Otsu, and Ali-akbar Agha-mohammadi. Perception-aware autonomous mast motion planning for planetary exploration rovers. *Journal of Field Robotics*, 37(5):812–829, 2020.

N. Sturtevant. Benchmarks for grid-based pathfinding. *Transactions on Computational Intelligence and AI in Games*, 4(2):144 – 148, 2012. URL http://web.cs.du.edu/~sturtevant/papers/benchmarks.pdf.

Dawei Sun, Susmit Jha, and Chuchu Fan. Learning certified control using contraction metric. In *Conference on Robot Learning*, 2020.

Brijen Thananjeyan, Ashwin Balakrishna, Ugo Rosolia, Felix Li, Rowan McAllister, Joseph E Gonzalez, Sergey Levine, Francesco Borrelli, and Ken Goldberg. Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks. *IEEE Robotics and Automation Letters*, 5(2):3612–3619, 2020.

Michael Vitus, Vijay Pradeep, Gabriel Hoffmann, Steven Waslander, and Claire Tomlin. Tunnel-milp: Path planning with sequential convex polytopes. In *AIAA guidance, navigation and control conference and exhibit*, page 7132, 2008.

Kim P Wabersich and Melanie N Zeilinger. Bayesian model predictive control: Efficient model exploration and regret bounds using posterior sampling. *arXiv preprint arXiv:2005.11744*, 2020a.

Kim P Wabersich and Melanie N Zeilinger. Performance and safety of bayesian model predictive control: Scalable model-based rl with guarantees. *arXiv preprint arXiv:2006.03483*, 2020b.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Appendix A. Proof of Theorem 1

To present the details of the proof, we need to introduce some additional notation. Let E denote the $r_0/2$ covering of $\mathcal{B}(\mathbf{o}, \rho)$ for some $0 < r_0 < 1$, and H denote the r_1 covering for some $r_1 < r_0/2$. Both the terms r_0 and r_1 will be specified later. Throughout this proof, we will use m_E and m_H to denote the cardinality of E and H respectively, and furthermore, for any $x \in \mathcal{B}(\mathbf{o}, \rho)$ we will use $[x]_E$ and $[x]_H$ to denote the element in E and H (respectively) that is closest to x. In the case of more than one point being the closest we will choose according to some predetermined rule. Finally, we will enumerate the elements of E as $\{z_1, z_2, \ldots, z_m\}$.

Now, suppose that the agent draws N i.i.d. points according to a sampling distribution Q from the region $\mathcal{B}(\mathbf{o}, \rho)$, and denote the drawn points by $S_N = \{X_1, X_2, \ldots, X_N\}$. Introduce the random variables $m_i = |S_N \cap \mathcal{B}(z_i, r_0/2)|$, denoting the numbers of random samples falling in the $r_0/2$ neighborhood of z_i, for $1 \leq i \leq m$.

For some given confidence level $\delta \in (0, 1)$ we introduce the following three events which can be ensured to occur simultaneously with probability at least $1 - \delta$.

- Suppose the set H is an $r_1 = 1/(NL\sqrt{n})$ (where $L = a_2\sqrt{\log(3a_1 n/\delta)}$) covering of $\mathcal{B}(\mathbf{o}, \rho)$ (recall that the terms a_1 and a_2 from Assumption 1). Introduce the event

$$\Omega_1 = \{|d^{(j)}(z) - \mu^{(j)}(z)| \leq \beta_N \sigma_t^{(j)}(z),\ \forall z \in H, \forall 1 \leq t \leq N\},$$

  where

$$\beta_N = \sqrt{2\log(3Nm_H/\delta)} \text{ and } m_H = C_n \left(\frac{NL\sqrt{n}}{\rho}\right)^n$$

  for some constant $C_n > 0$ depending only on n. Then, we have $\mathbb{P}(\Omega_1) \geq 1 - \delta/3$

  Proof The proof of this statement proceeds along the lines of the proofs of (Srinivas et al., 2010, Lemmas 5.5 & 5.6). In particular, we note that for any $z \in H$, the posterior is a normal random variable with mean $\mu_t(z)$ and variance $\sigma_t^2(z)$, and thus by the Gaussian tail inequality and two union bounds (one over the elements of H for a fixed t, and the second over $t = 1, 2, \ldots, N$) we get the required result. ■

- Next, we introduce the event $\Omega_2 = \{|\partial d(x)/\partial x| < L,\ \forall x \in \mathcal{B}(\mathbf{o}, \rho),\ \forall j = 1, 2, \ldots, n\}$ with $L = a_2\sqrt{\log\left(\frac{3a_1 n}{\delta}\right)}$. Then we have $\mathbb{P}(\Omega_2) \geq 1 - \delta/3$.

  Proof This result follows directly from the assumption on the covariance function, stated in Assumption 1, that there exist constants a_1 and a_2 such that for any $L > 0$, the event Ω_2 is satisfied with probability at least $1 - a_1 n e^{-L^2/a_2^2}$. The result then follows by plugging in the value of L used in the definition of the event Ω_2. ■

- Finally, we introduce the event $\Omega_3 = \{|m_i - Np_i| \leq \sqrt{2N\log(3m/\delta)},\ \forall 1 \leq i \leq m\}$ where $p_i = \int_{\mathcal{B}(z_i, r_0/2)} q(x)dx$ is the probability that a uniformly drawn sample from $\mathcal{B}(\mathbf{o},\ \rho)$ falls in $\mathcal{B}(x, r_0/2)$. Then we have $\mathbb{P}(\Omega_3) \geq 1 - \delta/3$.

  Proof The result follows by an application of Hoeffding's inequality and a union bound over elements of E followed by another union bound over the N time steps. ■

For the rest of the proof, we will work under the event $\Omega_1 \cap \Omega_2 \cap \Omega_3$, which as shown above occurs with probability at least $1-\delta$.

As a consequence of the simultaneous occurrence of Ω_1 and Ω_2, we note that for any $x \in \mathcal{B}(\mathbf{o}, \rho)$ we must have $d^{(j)}(x) \leq \mu_t^{(j)}([x]_H) + \beta_N \sigma_t^{(j)}([x]_H) + 1/N$. Thus if $N \geq 2\sqrt{n}/\psi$, then to obtain the required result, it suffices to show that $\beta_N \sigma_t^{(j)}(x) \leq \psi/(2\sqrt{n})$ for all $x \in H$. We proceed in the following steps:

- For any point $x \in H$, we note that there exists at least one $z_i \in E$ such that $\|x - z_i\| \leq r_0/2$. Consequently, the ball $\mathcal{B}(z_i, r_0/2)$ is contained in the larger ball of radius r_0 centered around x, i.e., $\mathcal{B}(x, r_0)$. Since, we assume that the event Ω_3 holds, this implies that the number of random points from S_N which fall in the ball $\mathcal{B}(x, r_0)$ is at least $m_i \geq N\left(p_{r_0} - \sqrt{\frac{2\log(2m/\delta)}{N}}\right)$.

  Thus by an application of (Shekhar and Javidi, 2018, Proposition 3), we note that after collecting N observations, the approximation error at the point x can be upper bounded as $|d^{(j)}(x) - \mu_t^{(j)}(x)| \leq \beta_N \sigma_t^{(j)}(x) \leq \beta_N \left(\frac{\sigma}{\sqrt{m_i}} + C_K r_0^{\omega}\right)$, where C_K is introduced in Assumption 1.

  Now, assuming that **(i)** $\beta_N \leq a$ for some $a > 0$, and **(ii)** that N is large enough to ensure that $\sigma/\sqrt{m_i} \leq C_K r_0^{\omega}$. Together these two assumptions imply that a suitable value of r_0 is $\left(\frac{\psi}{2aC_K}\right)^{1/\omega}$.

- Now, we obtain the sufficient conditions on N to ensure that the above two assumptions are satisfied. Recall, that we have already imposed the condition that N is large enough to ensure that $1/N < \psi/(2\sqrt{n})$ or equivalently $N > 2\sqrt{n}/\psi$. Additionally, we need N to be large enough to ensure that $\sigma/\sqrt{m_i} \leq C_K r_0^{\omega}$, and we break it into two parts:

  - N is large enough to ensure that $2\log(3m/\delta)/N \leq (p_i/2)^2$, a sufficient condition for which is to ensure that $2\log(3m/\delta)/N \leq (1/4)(\underline{c} C_n r_0^n)^2$, where the term $\underline{c}$ is introduced in Assumption 2. Since $a^2 \geq 2\log(2m/\delta)$ a sufficient condition for this is

$$N \geq a^{2+2n/\omega}\left(\frac{2C_K}{\psi}\right)^{2n/\omega}\left(\frac{2^{2n-2}}{\underline{c}^2 C_n^2}\right)$$

  - N is large enough to ensure that $\sigma/\sqrt{Np_i/2} \leq C_K r_0^{\omega}$ for all i, a sufficient condition for which is

$$N \geq \frac{2\sigma^2}{C_K^2 C_n \underline{c}}\left(\frac{2C_K a}{\psi}\right)^{(2\omega+n)/\omega}.$$

- Now, it remains to show that there exists an $a > 0$ such that if N satisfies the above two conditions then $2\log(2N^2/\delta) \leq a$. A sufficient condition for this is that

$$a \geq 2\max\left\{\log\left(\frac{8\sigma^2(2\rho)^{2n}}{\delta C_K^2}\left(\frac{C_K}{\psi}\right)^{(2\omega+n)/\omega}\right),\ \log\left(\frac{2(2\rho)^{4n}}{\delta}\left(\frac{C_K}{\psi}\right)^{4n/\omega}\right),\ a^*\right\}, \text{with}$$

$$a^* = \max\left\{e^{-W(-1/(8n/\omega+8))},\ e^{-W(-\omega/(4\omega+2n))}\right\}, \quad \text{where } W \text{ is the Lambert W-function.}$$

To conclude, a sufficient condition for ensuring that the estimated value of d is good enough with probability at least $1-\delta$ is that the agent draws at least N uniform samples in the ball $\mathcal{B}(\mathbf{o},\,\rho)$, where N satisfies:

$$N = \widetilde{\mathcal{O}}\left(\max\left\{2\sqrt{n}\psi^{-1},\ \frac{\psi^{-2n/\omega}}{\underline{c}^2},\ \frac{s^2\psi^{-(2\omega+n)/\omega}}{\underline{c}}\right\}\right),$$

where the notation $\widetilde{\mathcal{O}}$ suppresses the polylogarithmic factors of $\log(1/\delta)$ and $\log(1/\psi)$ (arising from the conditions on a).

Appendix B. Proof of Theorem 5

The following lemma is used for the proof of Theorem 5.

Lemma 6 *For any two symmetric matrices $A, B \in \mathbb{R}^{n\times n}$, the difference of their largest eigenvalues satisfies:*

$$|\lambda_{\max}(A) - \lambda_{\max}(B)| \leq \|A - B\|_2.$$

Lemma 6 is a well-known result that follows from the Courant-Fischer minimax theorem. The detailed proof can be found at Fan and Mitra (2015).

Proof (of Theorem 5). Plugging $\hat{d}_1$ and $\hat{d}_2$ into Equation (2), denote the LHS by $LHS(\hat{d}_1)$ and $LHS(\hat{d}_2)$ respectively. Then, we have

$$LHS(\hat{d}_1) - LHS(\hat{d}_2) = \partial_{\mathcal{R}} M + \mathtt{sym}\left(M\mathcal{R}\right).$$

Then, following from Lemma 6 and the assumption that $\hat{d}_1$ satisfies the robust condition, we have

$$\begin{aligned}
&\lambda_{\max}(LHS(\hat{d}_2))\\
\leq&\lambda_{\max}(LHS(\hat{d}_1)) + \|LHS(\hat{d}_1) - LHS(\hat{d}_2)\|_2\\
\leq&-\mathcal{M} + \|\partial_{\mathcal{R}} M + \mathtt{sym}\left(M\mathcal{R}\right)\|_2\\
\leq&0.
\end{aligned}$$

Thus, $LHS(\hat{d}_2) \prec 0$, which means $\hat{d}_2$ satisfies the original condition (2). ■

Appendix C. More Experimental Results

The progress of exploration is visualized in Fig. 3. A video is available at https://youtu.be/cG4o29ntBbE.

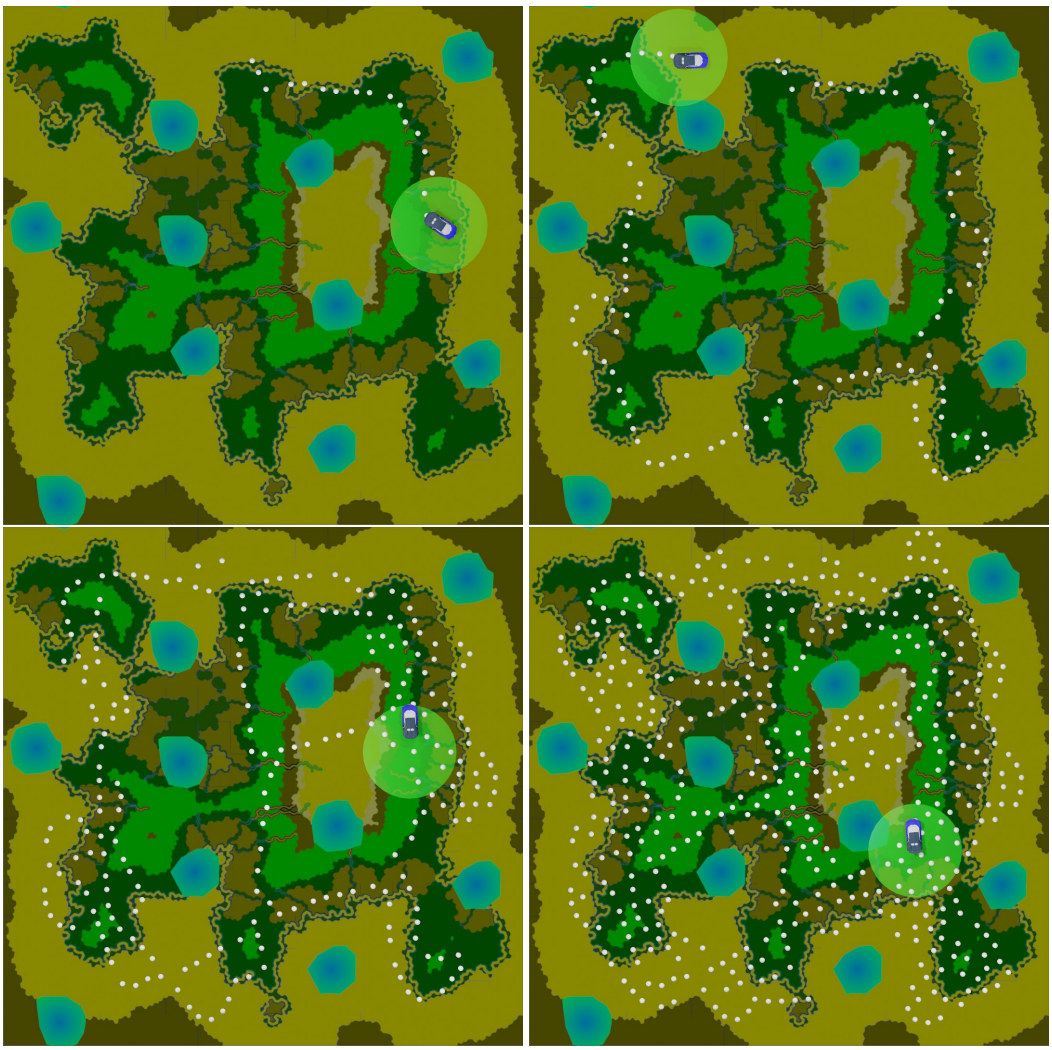

Figure 3: Exploration progress of the proposed method. White dots indicate the collected observations on the disturbance. Green transparent circle around the car is the ball $\mathcal{B}(\mathbf{x}, \rho)$ in Algorithm 1.