Reachability-Based Safe Learning with Gaussian Processes

Anayo K. Akametalu* Jaime F. Fisac* Jeremy H. Gillula
Shahab Kaynama Melanie N. Zeilinger Claire J. Tomlin

***Abstract*— Reinforcement learning for robotic applications faces the challenge of constraint satisfaction, which currently impedes its application to safety critical systems. Recent approaches successfully introduce safety based on reachability analysis, determining a safe region of the state space where the system can operate. However, overly constraining the freedom of the system can negatively affect performance, while attempting to learn less conservative safety constraints might fail to preserve safety if the learned constraints are inaccurate. We propose a novel method that uses a principled approach to learn the system's unknown dynamics based on a Gaussian process model and iteratively approximates the maximal safe set. A modified control strategy based on real-time model validation preserves safety under weaker conditions than current approaches. Our framework further incorporates safety into the reinforcement learning performance metric, allowing a better integration of safety and learning. We demonstrate our algorithm on simulations of a cart-pole system and on an experimental quadrotor application and show how our proposed scheme succeeds in preserving safety where current approaches fail to avoid an unsafe condition.**

I. Introduction

Reinforcement Learning (RL) has proven to be a valuable tool in robotics, where pre-specifying a policy for a robot to achieve a given task can be a major challenge. Through trial-and-error interactions with its environment, a robot can employ RL techniques online to find a control policy for achieving its task. Examples include rotorcraft performing aggressive maneuvers with high airflow interaction [1],[5], autonomous driving in extreme conditions [13] or fast quadruped locomotion through irregular terrain [10].

Traditional RL algorithms are not designed to guarantee constraint satisfaction, which makes them unemployable in safety-critical scenarios. For this reason, the past years have seen a growing interest in combining online learning with a supervisory framework that can guarantee the safety of the dynamical system. We propose a safety framework that has reduced interference with the learning process of the system. Its novel control strategy can preserve safety under weaker conditions than current safety approaches.

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley , CA 94720. {kakametalu, jfisac, jgillula, kaynama, melanie.zeilinger, tomlin} @eecs.berkeley.edu

*The first two authors contributed equally to this paper. This work is supported by the NSF CPS project ActionWebs under grant number 0931843, NSF CPS project FORCES under grant number 1239166, and by ONR under the HUNT, SMARTS and Embedded Humans MURIs, and by AFOSR under the CHASE MURI. The research of A.K. Akametalu has received funding from the NSF Bridge to Doctorate program. The research of J.F. Fisac has received funding from the "la Caixa" Foundation. The research of M.N. Zeilinger has received funding from the EU FP7 (FP7/2007-2013) under grant agreement no. PIOFGA-2011-301436-COGENT.

Different frameworks have been proposed in recent years using Lyapunov stability [16] or model predictive control with a nominal linear model [2]. One particular technique that has been explored recently for safe reinforcement learning [8],[9] is known as Hamilton-Jacobi-Isaacs (HJI) reachability analysis, a method that has been previously used to guarantee safety in a variety of contexts within the robotics literature [6]. By considering worst-case disturbances, this method determines a *safe* region in the state space and provides a control policy to stay within that region. The main advantage is that in the interior of this region one can execute any desired action as long as the safe control is applied at the boundary, leading to a least restrictive control law. The desired action can be specified by any method, including any learning algorithm.

However, reachability-based algorithms are not without shortcomings. First, in order to guarantee safety the system-designer must often rely on a nominal model that assumes conservative worst-case disturbances, which reduces the computed safe region, and thereby the region where the system can learn [9]. Second, the assumed bounds on the disturbances may not globally capture the true disturbance of the system, in which case current reachability-based methods can no longer guarantee safety. Lastly, the least restrictive control law framework decouples safety and learning, which can lead to poor results, since the learning controller has no notion of the unsafe regions of the state space and may attempt to drive the system into them. Beyond chattering due to controller switching, convergence of the learning controller may be extremely slow, if the safety frequently prevents the learning control from being applied.

The major contributions of this work address these three issues. Learning the disturbances from the data prevents an overly conservative reachability analysis, thus leading to a larger region of operation for the system [9]. We introduce a principled way of updating prior disturbance assumptions based on online data by means of a Gaussian process (GP) model. We propose a novel control strategy that validates the model online and becomes more conservative if its predictions account poorly for the observed dynamics. We present a method to validate the model in real time and provide an adapted control strategy, which can guarantee safety under relaxed conditions, even if assumptions on the disturbances are incorrect. This framework ultimately allows a less conservative modeling of system uncertainty for the reachability analysis. To our knowledge this is the first work in the area of reachability analysis that cross-validates the model online and proposes a solution for cases in which the

model does not accurately explain the observations. Lastly, we incorporate safety metrics in the learning algorithm that reduce the amount of switching between controllers.

The remainder of the paper is organized as follows. In Section II we introduce the model and briefly explain the employed tools, reachability analysis and GPs, which are at the core of our algorithm. The problem statement is presented in Section III, with a discussion of the limitations of state-of-the-art reachability-based safety algorithms. Section IV contains the proposed methodology to address these problems. Lastly, in Section V we demonstrate our framework on both simulations (cart-pole swing-up) and experimental results (quadrotor trajectory tracking).

II. Preliminaries

Let $x \in \mathbb{R}^n$ be the state of the system evolving according to dynamics $\dot{x} = f(x,u,d)$. We restrict f to the class of control-affine systems, for which parts of the dynamics are assumed known (or already identified) while other parts remain unknown[1]:

$$f(x,u,d) = h(x) + g(x)u + d, \tag{1}$$

where $h : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous functions representing the known parts of the dynamics, u is the control input belonging to some compact set $\mathscr{U} \subseteq \mathbb{R}^m$, and $d : \mathbb{R}^n \to \mathbb{R}^n$ is an unknown, but deterministic, state dependent disturbance capturing unmodeled dynamics. We assume that $d(x)$ is a locally Lipschitz continuous function. Though $d(x)$ is unknown, we bound it by a conservative compact set $\mathscr{D}(x)$, which is allowed to vary in the state space, and which in turn will be used in our initial safety assessment. Later on, we will use a GP model to approximate $d(x)$ with a tighter bound $\hat{\mathscr{D}}(x)$.

The model given by (1) captures an ample variety of systems, where the actuation is understood, but there exist uncertain factors such as air drag and turbulence for aircraft at high velocities or contact forces between the tires of a ground vehicle and spatially varying terrains.

A. Safety and Reachability Analysis

Consider the state constraint set $\mathscr{K}$, a compact subset of $\mathbb{R}^n$ that the system is required not to leave. Given some time horizon τ (possibly infinite) the goal is to compute the set of initial states of (1) for which there exists a control strategy that takes value in $\mathscr{U}$ such that $x(t) \in \mathscr{K}\ \ \forall t \in [0,\tau] =: \mathbb{T}$ regardless of the actions of the disturbance in $\mathscr{D}(x)$. This is known as the *discriminating kernel* of $\mathscr{K}$ denoted by $\mathrm{Disc}_{\mathbb{T}}(\mathscr{K},\mathscr{D})$ [3].

We can compute this set by solving the corresponding modified terminal value HJI partial differential equation [15]. Define a bounded Lipschitz continuous function $\kappa\colon \mathbb{R}^n \to \mathbb{R}$ such that $\mathscr{K} = \{x \in \mathbb{R}^n \mid \kappa(x) \geq 0\}$. A common choice for κ is the signed distance[2] to the set $\mathscr{K}^C = \mathbb{R}^n \setminus \mathscr{K}$. Then the viscosity solution $V : \mathbb{R}^n \times \mathbb{T} \to \mathbb{R}$ of

$$\frac{\partial V}{\partial t}(x,t) = -\min\left\{0, \sup_{u\in\mathscr{U}} \inf_{d\in\mathscr{D}(x)} \frac{\partial V}{\partial x}(x,t) f(x,u,d)\right\} \tag{2}$$

with $V(x,\tau) = \kappa(x)$ describes the discriminating kernel:

$$\mathrm{Disc}_{\mathbb{T}}(\mathscr{K},\mathscr{D}) = \{x \in \mathbb{R}^n \mid V(x,0) \geq 0\}. \tag{3}$$

The optimal policy $u^*(x)$, the optimizer of (2), attempts to drive the system to the safest possible state always assuming an adversarial disturbance. If the hypotheses of the reachability analysis hold true (namely $d \in \mathscr{D}(x)\ \ \forall x \in \mathscr{K}$), then one can allow the system to execute any desired control (in particular, any control u_l dictated by the learning agent) while in the interior $int(\mathrm{Disc}_{\mathbb{T}}(\mathscr{K},\mathscr{D}))$, as long as the safety preserving action $u^*(x)$ is taken whenever the state reaches the boundary $\partial\,\mathrm{Disc}_{\mathbb{T}}(\mathscr{K},\mathscr{D})$; the system is then guaranteed to remain inside $\mathscr{K}$ over $\mathbb{T}$. Typically, the level set computations will converge for a sufficiently large τ and it is then possible to guarantee safety for all time using the robust controlled invariant (henceforth simply controlled invariant) set $\mathrm{Disc}(\mathscr{K},\mathscr{D})$. For the rest of the paper we will only consider the converged discriminating kernel, and refer to $V(x)$ as the *safety value function*. The least restrictive control law for guaranteed safety under disturbance set $\mathscr{D}(x)$ is then given by:

$$u \in \begin{cases} \mathscr{U}, & \text{if } V(x) > 0 \\ \{u^*(x)\}, & \text{otherwise} \end{cases} \tag{4}$$

Lemma 1: Any nonnegative superlevel set of $V(x)$ is controlled invariant.

Proof: By Lipschitz continuity of f and κ, we have that V is Lipschitz continuous [7] and hence, by Rademacher's theorem, almost everywhere differentiable. The convergence of V in (2) implies that $\frac{\partial V}{\partial t}(x,t) = 0$. Therefore, $\forall \alpha \geq 0$ $\forall x \in \{x \mid V(x) \geq \alpha\}\ \exists u^*$ such that $\forall d\ \frac{\partial V}{\partial x}(x,t) f(x,u^*,d) \geq 0$ (otherwise r.h.s. would be non-zero). Then, the value of V along the trajectory is always non-decreasing, so $V \geq \alpha$. ■

Proposition 1: Consider two disturbance sets $\mathscr{D}_1(x)$ and $\mathscr{D}_2(x)$, and a compact set $\mathscr{M} \subset \mathbb{R}^n$ that is controlled invariant under $\mathscr{D}_1(x)$. If $\mathscr{D}_2(x) \subseteq \mathscr{D}_1(x)\ \ \forall x \in \partial\mathscr{M}$, then $\mathscr{M}$ is controlled invariant also under $\mathscr{D}_2(x)$.

Proof: For the sake of contradiction, assume that under the disturbance set $\mathscr{D}_2(x)$ there exists some trajectory starting at $x_0 \in \mathscr{M}$ such that for some $T < \infty$, $x(T) \notin \mathscr{M}$. Since f is locally Lipschitz, the trajectory $x\colon t \mapsto \mathbb{R}^n$ is continuous for $x \in \mathscr{M}$, and thus $\exists t' \in [t_0, T]$ such that $x(t') \in \partial\mathscr{M}$. However, since $\mathscr{M}$ is controlled invariant under $\mathscr{D}_1(x)$, we know that $\forall x \in \partial\mathscr{M}\ \exists u^*(x) \in \mathscr{U}$ such that no possible disturbance $d \in \mathscr{D}_1(x)$ can drive the system out of $\mathscr{M}$. Since $\mathscr{D}_2(x) \subseteq \mathscr{D}_1(x)$ $\forall x \in \partial\mathscr{M}$ the same control policy $u^*(x)$ on the boundary guarantees that no disturbance $d \in \mathscr{D}_2(x) \subseteq \mathscr{D}_1(x)$ can drive the system out of $\mathscr{M}$. Hence, $\mathscr{M}$ is a controlled invariant set under $\mathscr{D}_2(x)$. ■

[1]The proposed method applies to more general dynamical systems, but here we restrict our attention to classes of systems that can be addressed with currently available computational tools.

[2]For a given norm $\|\cdot\|$ on $\mathbb{R}^n$, the signed distance from $x \in \mathbb{R}^n$ to $\mathscr{A} \subset \mathbb{R}^n$ is $\inf\{\|x-y\|, y \in \mathscr{A}\}$ for $x \notin \mathscr{A}$, and $-\inf\{\|x-y\|, y \in \mathbb{R}^n \setminus \mathscr{A}\}$ for $x \in \mathscr{A}$.

Corollary 1: Let $\mathscr{Q}$ be a nonnegative superlevel set of the safety function $V(x)$ computed for some disturbance set $\mathscr{D}(x)$. If $d(x) \in \mathscr{D}(x)\ \forall x \in \partial\mathscr{Q}$, then $\mathscr{Q}$ is an invariant set under the optimal safe control policy $u^*(x)$ given by the HJI reachability analysis, even if $d(x) \notin \mathscr{D}(x)\ \forall x \in int(\mathscr{Q})$.

This corollary, which follows from Proposition 1 by considering the singleton $\{d(x)\}$, is an important result that will be at the core of the modified safety control policy presented in Section IV-B.

Finally, we define a *safe set* as any controlled invariant set $\mathscr{S}$ such that $\mathscr{S} \cap \mathscr{K}^C = \emptyset$. Therefore, any level set satisfying Corollary 1 is a safe set of the true system. Furthermore, if $\mathscr{D}(x) = \{d(x)\}\ \forall x \in \mathbb{R}^n$ then $\text{Disc}(\mathscr{K}, \mathscr{D})$ is the maximal safe set. In this case, the system dynamics are known completely and the disturbance set reduces to a singleton.

B. Gaussian Process

To estimate the disturbance function $d(x)$ over the state space we make use of GP regression. This is a powerful nonparametric regression technique that extends multivariate Gaussian regression to the infinite-dimensional space of functions and provides a closed form expression for Bayesian inference. Informally, a GP is a distribution over functions defined by a mean function $\mu(x)$ and a covariance kernel function $k(x,x')$:

$$d(x) \sim \mathscr{GP}(\mu(x), k(x,x')). \tag{5}$$

The class of the prior mean function and kernel function is chosen to capture the characteristics of the model (linearity, periodicity, etc), and is defined by a set of hyperparameters θ_p. These are typically set to maximize the marginal likelihood of an available set of training data, or alternatively to reflect some prior belief on the system. For a detailed description of GPs, see [18].

Given N measurements for the jth component of $d(x)$, i.e. $\hat{\mathbf{d}}^{\mathbf{j}} = \begin{bmatrix} \hat{d}_1^j & \dots & \hat{d}_N^j \end{bmatrix}^T$, observed with zero-mean Gaussian noise ε^j with variance σ_n^2 at the points $X = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}^T$, i.e. $\hat{d}_i^j = d^j(x_i) + \varepsilon^j$, and the Gaussian prior distribution, the posterior distribution of the function value $d^j(x_*)$ at a new point x_* is again a Gaussian distribution:

$$\begin{aligned} \bar{d}^j(x_*) &= \mu(x_*) + K(x_*,X)(K(X,X) + \sigma_n I)^{-1}(\hat{\mathbf{d}}^{\mathbf{j}}(\mathbf{x}) - \mu(X)), \\ \text{var}(d^j(x_*)) &= K(x_*,x_*) - K(x_*,X)(K(X,X) + \sigma_n I)^{-1}K(X,x_*), \end{aligned} \tag{6}$$

where $K_{ij}(X,X') = k(x_i,x'_j)$, and $\mu_i(X) = \mu(x_i)$. Therefore the GP provides both the expected value of the disturbance function at any arbitrary point x_* and a notion of the uncertainty of this estimate.

III. Problem Statement

Given that $d(x)$ is unknown, reachability analysis is computed with an estimated disturbance set $\hat{\mathscr{D}}(x)$ that attempts to provide a bound for $d(x)$. Since the maximal safe set of the system is given by $\text{Disc}(\mathscr{K}, \{d(x)\})$, a necessary condition to guarantee safety through (4) is $\text{Disc}(\mathscr{K}, \hat{\mathscr{D}}) \subseteq \text{Disc}(\mathscr{K}, \{d(x)\})$.

A. Safety and Learning Trade-Off

Ideally, $\text{Disc}(\mathscr{K}, \hat{\mathscr{D}}) = \text{Disc}(\mathscr{K}, \{d(x)\})$, yielding a control strategy that would allow the system to learn in the maximal safe set. Given prior observations of the disturbances our objective is to infer a small set that contains $d(x)$.

The approach in [9] begins with a conservative bound on the disturbance and then less conservative bounds are inferred from the data online (gradually growing the discriminating kernel). However, no principled way of updating the disturbance set is presented considering treatment of outliers, measurement noise, extrapolation, and interpolation. The technique proposed in this paper provides a principled inference method and additionally allows incorporation of prior beliefs in the model characteristics by employing a GP model.

B. Assumption Violations and Safety

A sufficient condition for safety under control strategy (4) is that $d(x)$ be captured by the estimated disturbance set. In the literature this has typically been achieved by using conservative bounds on the disturbance [8]. However, this approach generally produces a small (or even worse, empty) discriminating kernel.

Our objective, instead, is to infer $d(x)$, or at least to derive a tight bound, so as to have a control strategy that guarantees safety under less restrictive conditions, by monitoring the consistency between the model used for reachability and the observed system dynamics.

C. Integrating Safety Metric into Learning

The desired action in (4), applied in the interior of the safe set, is specified by a learning algorithm that has some performance objective typically not including safety [8], [9]. This causes the reachability analysis and the learning algorithm to be disjoint with respect to one another. Switching control from the learning algorithm can cause the system to chatter or stall.

In addition, RL algorithms use the feedback obtained from the environment to specify the next action that should be taken, which seeks to converge to or execute an optimal policy [11]. This is particularly critical for on-policy algorithms, which require that the specified learning control be the one executed by the system. Therefore, failing to incorporate safety metrics in the learning algorithm can degrade the convergence and performance. The objective here is to unify learning and safety by incorporating safety metrics in the learning.

IV. Methodology

The overall scheme of the proposed algorithm is as follows. GP regression is used to infer the disturbance set $\hat{\mathscr{D}}(x)$ from past observations of the dynamics; this disturbance set $\hat{\mathscr{D}}(x)$ is used to conduct reachability analysis[3] and obtain a safety function $V(x)$ and an optimal safe control policy $u^*(x)$. Online, we implement a modified control law, which

[3]If no data has yet been collected we use an initial disturbance model.

combines the least restrictive control with a novel safety strategy based on online model validation. Whenever the safety control law is not applied, the system applies the control specified by a safety-aware learning algorithm. Once a new batch of data of arbitrary size is available, the process is repeated.

A. Gaussian Process and Model Inference

Starting from a prior disturbance set, we use a GP to approximate the disturbance function $d(x)$ yielding a probabilistic bound $\hat{\mathscr{D}}(x)$. The objective is to obtain an estimated safe set $\text{Disc}(\mathscr{K},\hat{\mathscr{D}})$ that approaches the maximal safe set while providing safety guarantees. We assume that an approximation of the state derivatives is available (e.g. obtained by numerical differentiation), which is denoted by $\hat{f}(x,u)$. Since the disturbance considered in our formulation is additive, we can construct measurements of $d(x)$ for any state-input pair that has been visited as the residuals between the observed dynamics and the model's prediction:

$$\hat{d}(x) = \hat{f}(x,u(x)) - h(x) - g(x)u(x). \tag{7}$$

The residuals are used to infer a distribution over $d(x)$, and this distribution is used to construct the disturbance set $\hat{\mathscr{D}}(x)$ at every point x. We define the jth component of the disturbance, $d^j(x) \in \mathbb{R}$, as a GP with a zero mean function and squared exponential covariance function.

$$k(x,x') = \sigma_f^2 \exp\left(\frac{(x-x')^T L^{-1}(x-x')}{2}\right), \tag{8}$$

where L is a diagonal matrix, with L_i as the ith diagonal element, and $\theta_p = \left[\sigma_f^2, \sigma_n^2, L_1, \ldots, L_n\right]$ are the hyperparameters, σ_f^2 being the signal variance, σ_n^2 being the measurement noise variance, and L_i being the squared exponential's characteristic length for the ith state. The hyperparameters are chosen to maximize the marginal likelihood of the training data set, and are thus recomputed for each new batch of data.

We denote by $X = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^T$ the training data given by a subset of the measured states. We define the observed disturbance vector $\hat{\mathbf{d}}^{\mathbf{j}}(X)$, where the ith element is given by $d^j(x_i)$. Assuming that the disturbance is observed with Gaussian uncertainty and given the Gaussian prior distribution, the posterior distribution of $d^j(x_*)$ at any given state x_* can be obtained from (6).

The expected value $\bar{d}^j(x_*)$ and the standard deviation $\sigma_* = \sqrt{\text{var}(d^j(x_*))}$ of the disturbance can thus be computed for every point of interest x_*, and therefore the expected dynamics in (1) can be evaluated at any given state for the reachability analysis. A probabilistic bound is chosen for the maximum and minimum values of the disturbance, e.g. a ± 2-sigma bound for 95.5% confidence or ± 3-sigma for 99.7%. This results in the disturbance set:

$$\hat{\mathscr{D}}(x) = [\bar{d}(x_*) - m\sigma_*(x), \bar{d}(x_*) + m\sigma_*(x)] \quad \text{with } m \geq 0. \tag{9}$$

B. Safety through Online Model Validation

A key contribution of this paper is providing a method to anticipate and react to inaccuracies in the dynamic model that could invalidate reachability-based guarantees. The proposed algorithm uses online measurements of the system's evolution to monitor the local value of the disturbance $d(x)$ in real time using (7). We define the *model reliability margin* $\lambda(x)$ as:

$$\lambda(x) = \frac{\text{dist}\big(d(x), \hat{\mathscr{D}}(x)^C\big)}{\max\limits_{\delta \in \hat{\mathscr{D}}(x)} \text{dist}\big(\delta, \hat{\mathscr{D}}(x)^C\big)}, \tag{10}$$

where $\text{dist}(\cdot,\cdot) : \mathbb{R}^n \times 2^{\mathbb{R}^n} \to \mathbb{R}$ is the signed distance function. The model reliability margin is therefore a normalized signed distance function, with range [0,1] on the inside of $\hat{\mathscr{D}}(x)$ and negative outside, which provides a metric for confidence in the GP model along the system's trajectory. Given (9) with sufficiently large m, the probability of the disturbance function $d(x)$ taking values near or outside the boundary of $\hat{\mathscr{D}}(x)$ is arbitrarily low. Obtaining a measurement with small $\lambda(x) > 0$ indicates that the model is performing poorly, but its bound on the disturbance (on which reachability guarantees are based) is still correct locally. Conversely, if $\lambda(x) < 0$, then all theoretical safety guarantees are lost (the disturbance is playing an unexpected value for which the control action is not guaranteed to win the differential HJI game).

The new control strategy is given as follows:

$$V_L = \begin{cases} \max(V(x), V_L) & \text{if } \lambda(x) \leq \lambda_L \\ V_L, & \text{otherwise} \end{cases} \tag{11a}$$

$$u \in \begin{cases} \mathscr{U} & \text{if } V(x) - V_L > 0 \\ \{u^*(x)\}, & \text{otherwise} \end{cases}, \tag{11b}$$

where the *critical safety level* V_L is initialized to 0 whenever a new disturbance model is produced by the GP, and $\lambda_L \in (0,1)$ is a predefined threshold; the criteria to select its value will be discussed later in this section. We refer to the compact set $\{x : V(x) - V_L > 0\}$ as the *critical level set*; note that this set is initialized to $\text{Disc}(\mathscr{K},\hat{\mathscr{D}})$, and updated by the control strategy when the model reliability margin reaches λ_L. The purpose of the above control strategy is to keep the system within the critical level set. Effectively, the algorithm shrinks the allowed region of operation by pruning away those states potentially admitting disturbances that are not likely according to the model. The principle by which the new allowed operating region is chosen to be the critical level set is based on the following result.

Proposition 2: If a state z is reached such that $\lambda_L \geq \lambda(z) > 0$ and $\exists V_S \in [0, V(z)]$ such that $d(x) \in \hat{\mathscr{D}}(x)\ \forall x \in \{x : V(x) = V_S\}$, then the strategy given by (11) is guaranteed to keep the system safe.

Proof: Since $\forall x \in \{x : V(x) = V_S\}$, $d(x) \in \hat{\mathscr{D}}(x)$, by Corollary 1 the level set associated with V_S is indeed a controlled invariant set of the true system under control policy $u^*(x)$. Since the control strategy given by (11) will apply the control $u^*(x)$ for all x such that $V(x) \leq V_L$, then the system cannot leave the level set $\{x : V(x) \geq V_S\}$, and therefore safety is guaranteed. ■

Note that the conditions for guaranteeing safety under the control strategy in (11) are much less stringent than those

required by current reachability-based safety algorithms. In order to preserve safety, these frameworks require that the estimated disturbance set capture the true disturbance at least on the boundary of the computed discriminating kernel. Conversely, our algorithm guarantees safety as long as there exists *some* super-zero level set $\{x : V(x) = V_S\}$, $0 \leq V_S \leq V_L$ of the computed safety function $V(x)$ such that the disturbance is captured by the computed disturbance set on the boundary of the said set. In a continuous state space, there are an infinite number of candidate level set boundaries and it suffices that one of them satisfies the condition for safety guarantees to hold.

It is important to note that the choice of λ_L determines the conservativeness of the control strategy: a larger value of λ_L leads the algorithm to start applying the safe control policy $u^*(x)$ for smaller deviations in the measured disturbance with respect to its expected value as predicted by the GP. The likelihood of there existing a candidate level set boundary where the disturbance lies within the specified bounds is therefore larger for larger values of λ_L, but the algorithm will also be more sensitive to modeling error and become more restrictive for smaller inconsistencies with observations.

Moreover, Proposition 2 provides a *sufficient* condition for safety when implementing our proposed control strategy: safety may still be provided even when its premises do not hold. An example of this will be shown in Section V-B through an experiment.

C. Safety Metric Integration into Learning Algorithm

The goal is to reduce switching between the safe control and learning control, or equivalently minimize the number of times the system reaches the boundary of the critical level set. On the interior of the critical level set, the safety function $V(x) - V_L$ can be viewed as a measure of how far a state is from reaching the boundary of the set. The safety function is incorporated as a metric in the learning algorithm to discourage the system from reaching the boundary. This idea will be exemplified with the learning algorithm Policy Gradient via the Signed Derivative (PGSD).

PGSD is a model-free policy search algorithm introduced in [12]. We present the algorithm here briefly and refer the reader to [12] for a more detailed description. In policy gradient learning algorithms, a parametrized control policy is updated in order to optimize a given cost function over the state-action pair, $C(x,u)$. As an example, we consider a quadratic cost and a control that is linear in the state features

$$C(x,u) = \frac{1}{2}(x-x^*)^\top Q(x-x^*) + \frac{1}{2}u^\top Ru, \qquad u = \Theta\phi(x). \tag{12}$$

where x^* is the desired state, Q and R are diagonal positive semidefinite matrices penalizing deviation from x^* and control input respectively, $\phi(x) \in \mathbb{R}^k$ is a vector of features, each of which maps the state to a scalar value, and $\Theta \in \mathbb{R}^{m\times k}$ is a matrix of weights that linearly map these features into controls. The controller's objective is to minimize the cost incurred over a horizon H starting at x_0 while applying Θ:

$$J(x_0,\Theta) = \sum_{t=1}^{H} C(x_t,u_t), \qquad u_t = \Theta\phi(x). \tag{13}$$

For simplicity of notation we omit the arguments of the cost in the following. PGSD updates the parameters as:

$$\Theta \leftarrow \Theta - \alpha\nabla_\Theta J \tag{14a}$$

$$\nabla_\Theta J = \frac{1}{H}\sum_{t=0}^{H-1}(\nabla_{u_t} J)\phi(x_t)^\top \tag{14b}$$

$$\nabla_{u_t} J \approx \sum_{t'=t+1}^{H} S^\top Q(x_{t'} - x^*_{t'}) + Ru_{t'}, \tag{14c}$$

where $\alpha > 0$ is the step size, and $S_{i,j}$ is the sign of $\frac{\partial (x_{t'})_i}{\partial (u_t)_j}$ with the additional restriction that only one element in each row of S is nonzero corresponding to the control that has the largest effect on that given state. The safety metric can be included as follows:

$$C_S(x,u) = C(x,u) - \gamma\log(V(x) - V_L), \tag{15}$$

where γ is a weighting factor. The log barrier function goes to infinity at the boundary of the critical level set, and is approximately constant in the interior. Equation (14c) then becomes

$$\nabla_{u_t} J = \sum_{t'=t+1}^{H} S^\top Q(x_{t'} - x^*_{t'}) - \frac{\gamma S^\top \nabla_x V(x_{t'})}{V(x_{t'}) - V_L} + Ru_{t'}. \tag{16}$$

The gradient $\nabla_x V(x)$ is calculated from (2) during the reachability computations.

V. Simulations and Experiments

We highlight each aspect of our algorithm in different simulations and experiments. We use the Level Set Toolbox [14] to compute the discriminating kernel and the optimal safety-preserving control laws.The GP regression is done using the GPML toolbox [18].

The model and safe set are not computed at every time step due to computational cost. The cost of GP regression is cubic in the number of data points, and the cost of reachability analysis grows exponentially with the state dimension. We maintain a grid of the state space (over which we perform the reachability computations), and as such (9) must be obtained at each grid point which does not affect the computational complexity of the overall algorithm. We then have a look-up table specifying the disturbance set at each grid cell.

A. Cart-Pole Swing Up: Safety Metric Integration

In this simulation we highlight the benefits of including a safety metric in the learning algorithm. We implemented the PGSD algorithm described in Section IV-C on a simulated cart-pole system with one actuator controlling the thrust of the cart. The states of the system are $x = \begin{bmatrix} p & \dot{p} & \theta & \dot{\theta} \end{bmatrix}^\top$; position, velocity, clockwise angle (pendulum up is the origin), and angular velocity. The dynamics are

$$\ddot{p} = \frac{(u-d) + h_1\cos(\theta)\sin(\theta) + h_2\sin(\theta)(\dot{\theta})^2}{h_3 + h_4\cos^2(\theta)}, \tag{17}$$

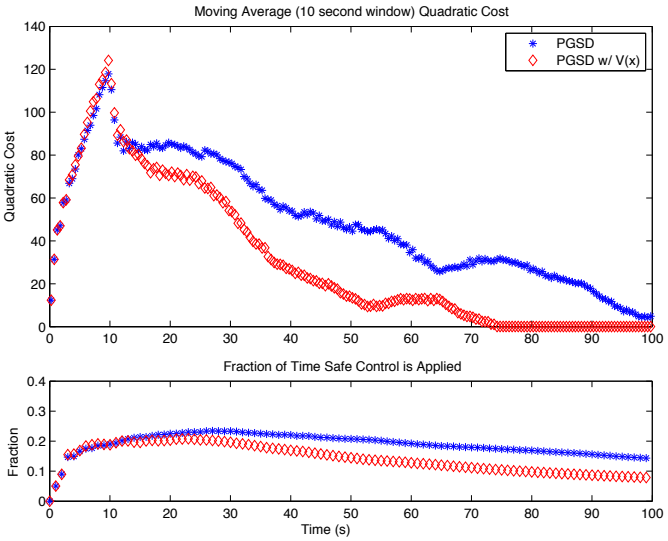

Fig. 1: Top: The pendulum learns the swing-up task faster when the safety metric is included in the learning. Bottom: Incorporating the safety metric reduces the intervention of the safety controller.

$$\ddot{\theta} = \frac{g_1 cos(\theta)(u-d) + g_2 \cos(\theta)\sin(\theta)(\dot{\theta})^2 + g_3 \sin(\theta)}{g_4 + g_5 \cos^2(\theta)}, \tag{18}$$

where $h_1, \ldots, h_4$ and $g_1, \ldots, g_5$ are physical constants, $\mathscr{U} = [-25 \text{ N}, 25 \text{ N}]$ is the set of valid thrust controls, and $\hat{\mathscr{D}}(x)$ is uniformly bounded over the entire state space by $[-5 \text{ N}, 5 \text{ N}]$.

The task is for the pendulum to swing itself up and stabilize at a fixed point $p^* = 0.25$ meters, while never leaving the track, defined by $\mathscr{K} = \{x : -0.5 \text{ m} \leq p \leq 0.5 \text{ m}\}$. The goal is to track the reference $x^* = \begin{bmatrix} p^* & 0 & 0 & 0 \end{bmatrix}^\top$, starting at the initial condition $x_0 = \begin{bmatrix} 0 & 0 & \pi & 0 \end{bmatrix}^\top$.

The feature vector, $\phi(x)$, now contains two sets of features. One set of features is made inactive (set equal to zero) when the pole is above the horizon ($\frac{\pi}{2} < |\theta|$), and the other set becomes inactive when the pendulum is below the horizon. The first set of features includes the error in each state, as well as the absolute position (five features in total). The second set of features contains the error in the position and velocity, the absolute position, and two disjoint features for the angle error that are inactivated depending on whether the sign of the angular velocity is the same as the sign of the angle error (five features in total). The absolute position acts as a safety feature that moves the cart away from the ends of the tracks. As for the cost function $R = 0, Q_{1,1} = 1, Q_{3,3} = 2$, and $Q_{2,2} = Q_{4,4} = 0$. For PGSD with safety $\gamma = 0.002$.

PGSD was run with and without the proposed safety metric in equation (15) for 100 seconds. The result can be seen in Fig. 1. Without the safety metric the controller switches more often, and it takes a longer time for the task to be completed.

B. Quadrotor Flight: Online Model Validation for Safety

We demonstrate the robust safety-preserving performance of our algorithm on the Stanford-Berkeley Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC), using an Ascending Technologies Pelican quadrotor (Fig. 2). The vehicle's altitude was controlled at 30 Hz based on state feedback from a VICON motion capture system. The code for our algorithm was developed in MATLAB and run online using the open-source control software starmac-ros-pkg [4] within the Robot Operating System (ROS) framework [17].

Rather than focusing on the performance of reinforcement learning within the safety framework, the purpose of the results presented here is to illustrate how our novel algorithm can provide safety under inexact models of the disturbance while a more accurate representation is learned. To this end, the vehicle is given a 60 second reference altitude trajectory designed to deliberately violate the state constraints (defined by the floor and ceiling of the room) as well as other potentially dangerous regions of the state space (e.g. aggressively approaching the floor or the ceiling); blindly attempting to follow this trajectory, or doing so with a standard reachability-based safety framework based on an inaccurate dynamic model, can result in loss of safety.

We use a nonlinear dynamic model of quadrotor vertical flight accounting for actuator delay, with state equations:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= k_T x_3^2 + g + d(x) \\ \dot{x}_3 &= k_p(u - x_3) \end{aligned} \tag{19}$$

where x_1 is the vehicle's altitude, x_2 is its vertical velocity, x_3 is the current average angular velocity of the four rotors and u is the commanded average angular velocity. State x_3 tracks control input u with a time constant $1/k_p$ measured to be 0.15 s. The gravitational acceleration is $g = -9.8$ m/s^2 and d is the disturbance term used to account for unmodeled forces in the system. The state constraints are $\mathscr{K} = \{x : 0.5 \text{ m} \leq x_1 \leq 2.8 \text{ m}\}$.

In the first experiment, we show the algorithm's ability to iteratively learn the disturbance throughout the state space using a GP, while guaranteeing safety even when a given iteration produces a bad estimate $\hat{\mathscr{D}}(x)$, which does not contain $d(x)$, leading to an overoptimistic approximation of the safe set. For every state x, we define $\hat{\mathscr{D}}(x)$ as the $\mu \pm 2\sigma$ interval of the distribution $d(x)$ predicted by the GP.

The vehicle starts off with an a priori conservative global bound on $d(x)$ and computes an initial conservative discriminating kernel $\Omega_0 = \text{Disc}(\mathscr{K}, \hat{\mathscr{D}})$ (shown in Fig. 4). It then follows the given trajectory avoiding the unsafe regions by transitioning to the safe control $u^*(x)$ on $\partial\Omega_0$, as shown in Fig. 3.

The disturbance is measured and monitored online during this test (with model reliability threshold $\lambda_L = 0.1$) and is found to be consistent with the initial conservative bound. After this iteration, a GP model is computed. Constructing the state-dependent disturbance bound $\hat{\mathscr{D}}(x) = \{d \in \mu(x) \pm$

Fig. 2: A STARMAC quadrotor during the flight test.

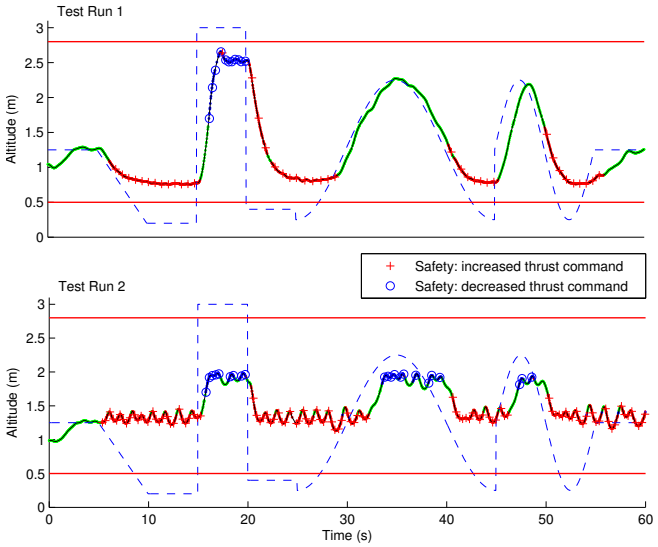

Fig. 3: Altitude trajectory of the quadrotor while learning the disturbance and tracking a reference. Run 1 is performed with a conservative disturbance model, while Run 2 is performed with an inferred model that fails to predict the true disturbance in certain regions of the state space.

$2\sigma(x)\}$, a second discriminating kernel Ω_1 is computed as an approximation of the unknown true safe set.

The results of this new reachability analysis are used for the second run: 5 seconds into the test, the vehicle measures a disturbance d that approaches the boundary of $\hat{\mathscr{D}}(x)$ so that $\lambda(x) < \lambda_L$, and immediately reacts by contracting the safe set boundary to the current level set $V(x) = V_L$. This contraction process takes place several times during the test run, as the vehicle measures disturbances which are close to becoming inconsistent with the model. Fig. 4 shows the contraction of Ω_1 to the level set $\tilde{\Omega}_1 = \{x : V(x) \geq V_L\}$; the algorithm succeeds in keeping the state of the system inside this controlled invariant set.

After the test is safely completed, a new GP is computed, resulting in an estimated safe set Ω_2, contained between the initial over-conservative discriminating kernel and the intermediate over-optimistic estimate, as depicted in Fig. 4. This indicates the algorithm's ability to safely converge to the true safe set (asymptotically, as sufficient data is collected) without requiring this process to consist in a monotonic succession of under-approximations (an assumption made by previous approaches often violated in practice).

To further illustrate the strength of our proposed approach, we consider one additional experiment, in which we compare the behavior of our safety algorithm with that of a standard reachability-based safety framework with no online validation when presented with an initial model that fails to account for the true disturbance. For the sake of fairness, we define a trajectory that does not explicitly attempt to drive the system out of the state constraints (which in this case are re-defined to be $\mathscr{K} = \{x : 0.2 \text{ m} \leq x_1 \leq 2.8 \text{ m}\}$, 0.2 m corresponding to the actual position of the body-frame origin when the base of the vehicle comes to physical contact with the ground). The initial model in this case assumes a prior bound on $d(x)$ which is 10 times smaller than in the previous experiment. Both algorithms begin with the same initial discriminating kernel Ω_0.

Once the test begins, the standard algorithm breaches the computed safe set on several occasions and violates the constraints incurring two consecutive ground collisions, marked in Fig. 5. Conversely, the proposed framework immediately detects a persistent unmodeled disturbance (in this case, $\lambda(x) < 0$) and consequently reduces the safe set to the current safety value, immediately applying the control action u^*: this constitutes the system's best effort for safety given its current knowledge of the system, even when the sufficient conditions from Proposition 2 do not hold (since the model reliability margin $\lambda(x)$ is already negative to begin with). Since the measured disturbance keeps breaking the model's assumptions the system keeps contracting its safe set to higher level curves until it reaches what is, to the best of its knowledge, the safest grid cell in the state space. It then computes a new disturbance model and the associated discriminating kernel. The resulting kernel Ω_1 is comparable in shape and size to the initial set Ω_0 in the previously discussed experiment, which means that the learning and state exploration can continue normally—and safely—after this step.[4] While the results in Fig. 5 may look quite restrictive, it should be noted that this is an extreme case where the model of the system is wrong everywhere. Under these conditions, temporarily restricting the vehicle's motion is generally preferable to a crash.

VI. Conclusions and Future Work

We have introduced a general reachability-based safe learning algorithm that leverages GPs to learn a model of the system disturbances and employs a novel control strategy based on online model validation, providing stronger safety guarantees than current state-of-the-art reachability-based frameworks. In addition, our algorithm makes improvements

[4]The results after Ω_1 is computed are omitted because they are similar to the previous experiment.

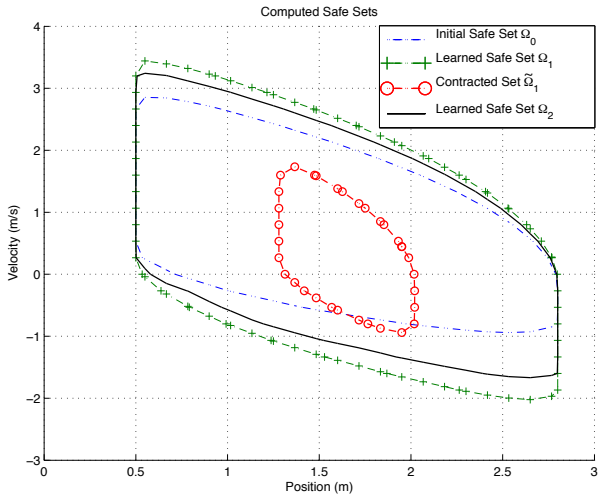

Fig. 4: Successive approximations of Disc$(\mathscr{K},\mathscr{D})$ under different disturbance models; a representative 2D section is shown for a fixed value of rotor velocity.

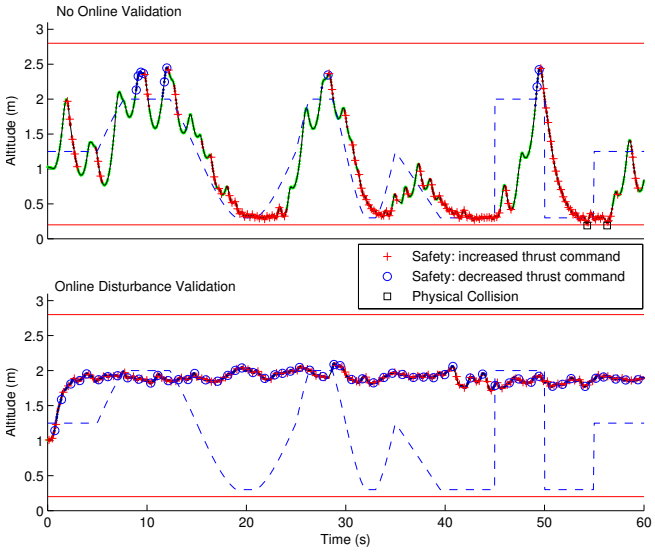

Fig. 5: Altitude trajectory of the quadrotor with an incorrect disturbance model. The standard reachability-based safety algorithm attempts to follow the trajectory, incurring several safety breaches and two physical collisions. The proposed algorithm rapidly detects the inconsistency between model and observations and retracts to the region with the highest computed safety value, until it can recompute a more accurate disturbance model and discriminating kernel.

upon these techniques by incorporating the safety value function into the performance metric of the learning algorithm, which reduces the interference of the safe control policy with the learning process.

In the future, we plan to develop an algorithm that can perform real-time updates to the safe set for every data point, in contrast to our current batch-based approach. In addition, we intend to look into speeding up GP computation by incorporating sparse GPs.

References

[1] P Abbeel, A Coates, M Quigley, and A Y Ng. An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19, 2007.

[2] A Aswani, H Gonzalez, S S Sastry, and C J Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 2013.

[3] J P Aubin, A M Bayen, and P Saint-Pierre. *Viability Theory: New Directions*. Springer Verlag, 2nd edition, 2011.

[4] P Bouffard. starmac-ros-pkg. http://www.ros.org/wiki/starmac-ros-pkg, 2011.

[5] A Coates, P Abbeel, and A Y Ng. Learning for control from multiple demonstrations. *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 144–151, July 2008.

[6] J Ding and et al. Hybrid systems in robotics: toward reachability-based controller design. *IEEE Robotics & Automation Magazine*, September 2011.

[7] L. C. Evans and P. E. Souganidis. Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. *Indiana University mathematics journal*, 33(5):773–797, 1984.

[8] J H Gillula and C J Tomlin. Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. *2012 IEEE International Conference on Robotics and Automation*, pages 2723–2730, May 2012.

[9] J H Gillula and C J Tomlin. Reducing conservativeness in safety guarantees by learning disturbances online: iterated guaranteed safe online learning. *Robotics: Science and Systems*, 2012.

[10] M Kalakrishnan, J Buchli, P Pastor, M Mistry, and S Schaal. Fast, robust quadruped locomotion over challenging terrain. *2010 IEEE International Conference on Robotics and Automation*, pages 2665–2670, May 2010.

[11] M Kearns and S Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232, 2002.

[12] J Z Kolter and A Y Ng. Policy search via the signed derivative. *Robotics: science and systems*, 2009.

[13] J Z Kolter, C Plagemann, D T Jackson, A Y Ng, and S Thrun. A probabilistic approach to mixed open-loop and closed-loop control, with application to extreme autonomous driving. *2010 IEEE International Conference on Robotics and Automation*, pages 839–845, May 2010.

[14] I M Mitchell. A toolbox of level set methods. Available: http://people.cs.ubc.ca/mitchell/ToolboxLS/index.html, 2009.

[15] I M Mitchell, A M Bayen, and C J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, July 2005.

[16] T J Perkins and A G Barto. Lyapunov design for safe reinforcement learning. *The Journal of Machine Learning Research*, 3:803–832, 2003.

[17] M Quigley, B Gerkey, K Conley, J Faust, T Foote, J Leibs, E Berger, R Wheeler, and A Y Ng. Ros : an open- source robot operating system. http://ai.stanford.edu/ang/papers/icraoss09-ROS.pdf, 2009.

[18] C E Rasmussen and C K I Williams. *Gaussian processes for machine learning.*, volume 14. MIT Press, April 2006.