Safe Model-based Reinforcement Learning with Stability Guarantees

Felix Berkenkamp
Department of Computer Science
ETH Zurich
befelix@inf.ethz.ch

Matteo Turchetta
Department of Computer Science,
ETH Zurich
matteotu@inf.ethz.ch

Angela P. Schoellig
Institute for Aerospace Studies
University of Toronto
schoellig@utias.utoronto.ca

Andreas Krause
Department of Computer Science
ETH Zurich
krausea@ethz.ch

Abstract

Reinforcement learning is a powerful paradigm for learning optimal policies from experimental data. However, to find optimal policies, most reinforcement learning algorithms explore all possible actions, which may be harmful for real-world systems. As a consequence, learning algorithms are rarely applied on safety-critical systems in the real world. In this paper, we present a learning algorithm that explicitly considers safety, defined in terms of stability guarantees. Specifically, we extend control-theoretic results on Lyapunov stability verification and show how to use statistical models of the dynamics to obtain high-performance control policies with provable stability certificates. Moreover, under additional regularity assumptions in terms of a Gaussian process prior, we prove that one can effectively and safely collect data in order to learn about the dynamics and thus both improve control performance and expand the safe region of the state space. In our experiments, we show how the resulting algorithm can safely optimize a neural network policy on a simulated inverted pendulum, without the pendulum ever falling down.

在model-based的前提下保障安全,
使用Lyapunov 实现完备的理论证明

1 Introduction

While reinforcement learning (RL, [1]) algorithms have achieved impressive results in games, for example on the Atari platform [2], they are rarely applied to real-world physical systems (e.g., robots) outside of academia. The main reason is that RL algorithms provide optimal policies only in the long-term, so that intermediate policies may be unsafe, break the system, or harm their environment. This is especially true in safety-critical systems that can affect human lives. Despite this, safety in RL has remained largely an open problem [3].

强化学习考虑了长期的最优奖励，导致了短期的策略是不安全的

Consider, for example, a self-driving car. While it is desirable for the algorithm that drives the car to improve over time (e.g., by adapting to driver preferences and changing environments), any policy applied to the system has to guarantee safe driving. Thus, it is not possible to learn about the system through random exploratory actions, which almost certainly lead to a crash. In order to avoid this problem, the learning algorithm needs to consider its ability to safely recover from exploratory actions. In particular, we want the car to be able to recover to a safe state, for example, driving at a reasonable speed in the middle of the lane. This ability to recover is known as *asymptotic stability* in control theory [4]. Specifically, we care about the *region of attraction* of the closed-loop system under a policy. This is a subset of the state space that is forward invariant so that any state trajectory that starts within this set stays within it for all times and converges to a goal state eventually.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

In this paper, we present a RL algorithm for continuous state-action spaces that provides these kind of high-probability safety guarantees for policies. In particular, we show how, starting from an initial, safe policy we can expand our estimate of the region of attraction by collecting data inside the safe region and adapt the policy to both increase the region of attraction and improve control performance.

这里的region of attraction是指什么意思？

Related work Safety is an active research topic in RL and different definitions of safety exist [5, 6]. *Discrete* Markov decision processes (MDPs) are one class of tractable models that have been analyzed. In risk-sensitive RL, one specifies risk-aversion in the reward [7]. For example, [8] define risk as the probability of driving the agent to a set of known, undesirable states. Similarly, robust MDPs maximize rewards when transition probabilities are uncertain [9, 10]. Both [11] and [12] introduce algorithms to safely explore MDPs so that the agent never gets stuck without safe actions. All these methods require an accurate probabilistic model of the system.

将risk定义为没有见过的state？

现在的这些方法都需要环境的dynamic model

In *continuous* state-action spaces, model-free policy search algorithms have been successful. These update policies without a system model by repeatedly executing the same task [13]. In this setting, [14] introduces safety guarantees in terms of constraint satisfaction that hold in expectation. High-probability worst-case safety guarantees are available for methods based on Bayesian optimization [15] together with Gaussian process models (GP, [16]) of the cost function. The algorithms in [17] and [18] provide high-probability safety guarantees for any parameter that is evaluated on the real system. These methods are used in [19] to safely optimize a parametric control policy on a quadrotor. However, resulting policies are task-specific and require the system to be reset.

现有的一些在连续环境中的安全保障，值适用于特定的环境，在新的环境中需要重新训练

In the *model-based* RL setting, research has focused on safety in terms of state constraints. In [20, 21], *a priori* known, safe global backup policies are used, while [22] learns to switch between several safe policies. However, it is not clear how one may find these policies in the first place. Other approaches use model predictive control with constraints, a model-based technique where the control actions are optimized online. For example, [23] models uncertain environmental constraints, while [24] uses approximate uncertainty propagation of GP dynamics along trajectories. In this setting, robust feasability and constraint satisfaction can be guaranteed for a learned model with bounded errors using robust model predictive control [25]. The method in [26] uses reachability analysis to construct safe regions in the state space. The theoretical guarantees depend on the solution to a partial differential equation, which is approximated.

Theoretical guarantees for the stability exist for the more tractable stability analysis and verification under a *fixed* control policy. In control, stability of a known system can be verified using a Lyapunov function [27]. A similar approach is used by [28] for deterministic, but unknown dynamics that are modeled as a GP, which allows for provably safe learning of regions of attraction for fixed policies. Similar results are shown in [29] for stochastic systems that are modeled as a GP. They use Bayesian quadrature to compute provably accurate estimates of the region of attraction. These approaches do not update the policy.

Our contributions We introduce a novel algorithm that can safely optimize policies in continuous state-action spaces while providing high-probability safety guarantees in terms of stability. Moreover, we show that it is possible to exploit the regularity properties of the system in order to *safely learn* about the dynamics and thus improve the policy and increase the estimated safe region of attraction without ever leaving it. Specifically, starting from a policy that is known to stabilize the system locally, we gather data at informative, safe points and improve the policy safely based on the improved model of the system and prove that any exploration algorithm that gathers data at these points reaches a natural notion of *full exploration*. We show how the theoretical results transfer to a *practical algorithm* with safety guarantees and apply it to a simulated inverted pendulum stabilization task.

2 Background and Assumptions

We consider a deterministic, discrete-time dynamic system

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) = h(\mathbf{x}_t, \mathbf{u}_t) + g(\mathbf{x}_t, \mathbf{u}_t), \tag{1}$$

with states $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^q$ and control actions $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^p$ and a discrete time index $t \in \mathbb{N}$. The true dynamics $f \colon \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ consist of two parts: $h(\mathbf{x}_t, \mathbf{u}_t)$ is a known, prior model that can be obtained from first principles, while $g(\mathbf{x}_t, \mathbf{u}_t)$ represents *a priori* unknown model errors. While the model errors are unknown, we can obtain noisy measurements of $f(\mathbf{x}, \mathbf{u})$ by driving the system to the state $\mathbf{x}$ and taking action $\mathbf{u}$. We want this system to behave in a certain way, e.g., the car driving on the road. To this end, we need to specify a control policy $\pi\colon \mathcal{X} \to \mathcal{U}$ that, given the current state, determines the appropriate control action that drives the system to some goal state, which we set as the origin without loss of generality [4]. We encode the performance requirements of how to drive the system to the origin through a positive cost $r(\mathbf{x}, \mathbf{u})$ that is associated with states and actions and has $r(\mathbf{0}, \mathbf{0}) = 0$. The policy aims to minimize the cumulative, discounted costs for each starting state.

The goal is to safely learn about the dynamics from measurements and adapt the policy for performance, without encountering system failures. Specifically, we define the safety constraint on the state divergence that occurs when leaving the region of attraction. This means that adapting the policy is not allowed to decrease the region of attraction and exploratory actions to learn about the dynamics $f(\cdot)$ are not allowed to drive the system outside the region of attraction. The region of attraction is *not known a priori*, but is implicitly defined through the system dynamics and the choice of policy. Thus, the policy not only defines performance as in typical RL, but also determines safety and where we can obtain measurements.

Model assumptions In general, this kind of safe learning is impossible without further assumptions. For example, in a discontinuous system even a slight change in the control policy can lead to drastically different behavior. Moreover, to expand the safe set we need to generalize learned knowledge about the dynamics to (potentially unsafe) states that we have not visited. To this end, we restrict ourselves to the general and practically relevant class of models that are Lipschitz continuous. This is a typical assumption in the control community [4]. Additionally, to ensure that the closed-loop system remains Lipschitz continuous when the control policy is applied, we restrict policies to the rich class of L_π-Lipschitz continuous functions Π_L, which also contains certain types of neural networks [30].

Assumption 1 (continuity)**.** *The dynamics $h(\cdot)$ and $g(\cdot)$ in* (1) *are L_h- and L_g Lipschitz continuous with respect to the 1-norm. The considered control policies π lie in a set Π_L of functions that are L_π-Lipschitz continuous with respect to the 1-norm.*

To enable safe learning, we require a reliable statistical model. While we commit to GPs for the exploration analysis, for safety any suitable, well-calibrated model is applicable.

Assumption 2 (well-calibrated model)**.** *Let $\mu_n(\cdot)$ and $\Sigma_n(\cdot)$ denote the posterior mean and covariance matrix functions of the statistical model of the dynamics* (1) *conditioned on n noisy measurements. With $\sigma_n(\cdot) = \mathrm{trace}(\Sigma_n^{1/2}(\cdot))$, there exists a $\beta_n > 0$ such that with probability at least $(1 - \delta)$ it holds for all $n \geq 0$, $\mathbf{x} \in \mathcal{X}$, and $\mathbf{u} \in \mathcal{U}$ that $\|f(\mathbf{x}, \mathbf{u}) - \mu_n(\mathbf{x}, \mathbf{u})\|_1 \leq \beta_n \sigma_n(\mathbf{x}, \mathbf{u})$.*

This assumption ensures that we can build confidence intervals on the dynamics that, when scaled by an appropriate constant β_n, cover the true function with high probability. We introduce a specific statistical model that fulfills both assumptions under certain regularity assumptions in Sec. 3.

Lyapunov function To satisfy the specified safety constraints for safe learning, we require a tool to determine whether individual states and actions are safe. In control theory, this safety is defined through the region of attraction, which can be computed for a fixed policy using Lyapunov functions [4]. Lyapunov functions are continuously differentiable functions $v\colon \mathcal{X} \to \mathbb{R}_{\geq 0}$ with $v(\mathbf{0}) = 0$ and $v(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X} \setminus \{\mathbf{0}\}$. The key idea behind using Lyapunov functions to show stability of the system (1) is similar to that of gradient descent on strictly quasiconvex functions: if one can show that, given a policy π, applying the dynamics f on the state maps it to strictly smaller values on the Lyapunov function ('going downhill'), then the state eventually converges to the equilibrium point at the origin (minimum). In particular, the assumptions in Theorem 1 below imply that v is strictly quasiconvex within the region of attraction if the dynamics are Lipschitz continuous. As a result, the one step decrease property for all states within a level set guarantees eventual convergence to the origin.

Theorem 1 ([4])**.** *Let v be a Lyapunov function, f Lipschitz continuous dynamics, and π a policy. If $v(f(\mathbf{x}, \pi(\mathbf{x}))) < v(\mathbf{x})$ for all $\mathbf{x}$ within the level set $\mathcal{V}(c) = \{\mathbf{x} \in \mathcal{X} \setminus \{\mathbf{0}\} \,|\, v(\mathbf{x}) \leq c\}$, $c > 0$, then $\mathcal{V}(c)$ is a region of attraction, so that $\mathbf{x}_0 \in \mathcal{V}(c)$ implies $\mathbf{x}_t \in \mathcal{V}(c)$ for all $t > 0$ and $\lim_{t\to\infty} \mathbf{x}_t = \mathbf{0}$.*

It is convenient to characterize the region of attraction through a level set of the Lyapunov function, since it replaces the challenging test for convergence with a one-step decrease condition on the Lyapunov function. For the theoretical analysis in this paper, we assume that a Lyapunov function is given to determine the region of attraction. For ease of notation, we also assume $\partial v(\mathbf{x})/\partial \mathbf{x} \neq \mathbf{0}$ for all $\mathbf{x} \in \mathcal{X} \setminus \mathbf{0}$, which ensures that level sets $\mathcal{V}(c)$ are connected if $c > 0$. Since Lyapunov functions are continuously differentiable, they are L_v-Lipschitz continuous over the compact set $\mathcal{X}$.

In general, it is not easy to find suitable Lyapunov functions. However, for physical models, like the prior model h in (1), the energy of the system (e.g., kinetic and potential for mechanical systems) is a good candidate Lyapunov function. Moreover, it has recently been shown that it is possible to compute suitable Lyapunov functions [31, 32]. In our experiments, we exploit the fact that value functions in RL are Lyapunov functions if the costs are strictly positive away from the origin. This follows directly from the definition of the value function, where $v(\mathbf{x}) = r(\mathbf{x}, \pi(\mathbf{x})) + v(f(\mathbf{x}, \pi(\mathbf{x})) \leq v(f(\mathbf{x}, \pi(\mathbf{x})))$. Thus, we can obtain Lyapunov candidates as a by-product of approximate dynamic programming.

Initial safe policy Lastly, we need to ensure that there exists a safe starting point for the learning process. Thus, we assume that we have an initial policy π_0 that renders the origin of the system in (1) asymptotically stable within some small set of states $\mathcal{S}_0^x$. For example, this policy may be designed using the prior model h in (1), since most models are locally accurate but deteriorate in quality as state magnitude increases. This policy is explicitly *not safe* to use throughout the state space $\mathcal{X} \setminus \mathcal{S}_0^x$.

3 Theory

In this section, we use these assumptions for safe reinforcement learning. We start by computing the region of attraction for a fixed policy under the statistical model. Next, we optimize the policy in order to expand the region of attraction. Lastly, we show that it is possible to safely learn about the dynamics and, under additional assumptions about the model and the system's reachability properties, that this approach expands the estimated region of attraction safely. We consider an idealized algorithm that is amenable to analysis, which we convert to a practical variant in Sec. 4. See Fig. 1 for an illustrative run of the algorithm and examples of the sets defined below.

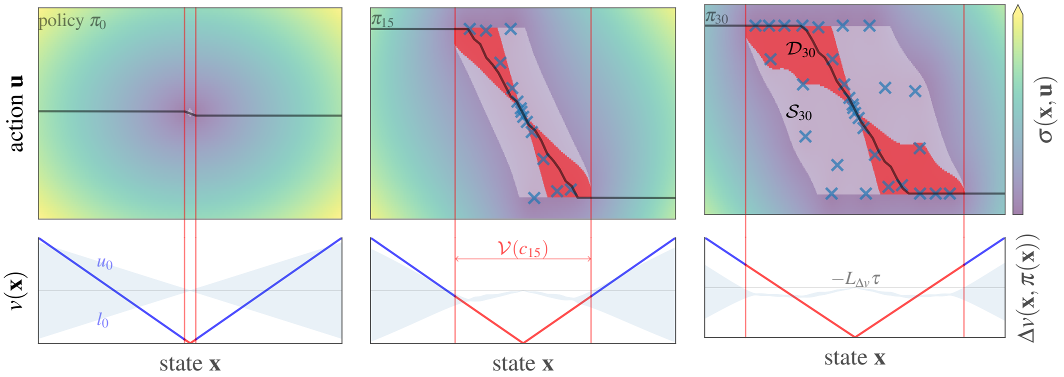
Region of attraction We start by computing the region of attraction for a fixed policy. This is an extension of the method in [28] to discrete-time systems. We want to use the Lyapunov decrease condition in Theorem 1 to guarantee safety for the statistical model of the dynamics. However, the posterior uncertainty in the statistical model of the dynamics means that one step predictions about $v(f(\cdot))$ are uncertain too. We account for this by constructing high-probability confidence intervals on $v(f(\mathbf{x}, \mathbf{u}))$: $\mathcal{Q}_n(\mathbf{x}, \mathbf{u}) := [v(\mu_{n-1}(\mathbf{x}, \mathbf{u})) \pm L_v \beta_n \sigma_{n-1}(\mathbf{x}, \mathbf{u})]$. From Assumption 2 together with the Lipschitz property of v, we know that $v(f(\mathbf{x}, \mathbf{u}))$ is contained in $\mathcal{Q}_n(\mathbf{x}, \mathbf{u})$ with probability at least $(1 - \delta)$. For our exploration analysis, we need to ensure that safe state-actions cannot become unsafe; that is, an initial set of safe set $\mathcal{S}_0$ remains safe (defined later). To this end, we intersect the confidence intervals: $\mathcal{C}_n(\mathbf{x}, \mathbf{u}) := \mathcal{C}_{n-1} \cap \mathcal{Q}_n(\mathbf{x}, \mathbf{u})$, where the set $\mathcal{C}$ is initialized to $\mathcal{C}_0(\mathbf{x}, \mathbf{u}) = (-\infty, v(\mathbf{x}) - L_{\Delta v} \tau)$ when $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}_0$ and $\mathcal{C}_0(\mathbf{x}, \mathbf{u}) = \mathbb{R}$ otherwise. Note that $v(f(\mathbf{x}, \mathbf{u}))$ is contained in $\mathcal{C}_n(\mathbf{x}, \mathbf{u})$ with the same $(1 - \delta)$ probability as in Assumption 2. The upper and lower bounds on $v(f(\cdot))$ are defined as $u_n(\mathbf{x}, \mathbf{u}) := \max \mathcal{C}_n(\mathbf{x}, \mathbf{u})$ and $l_n(\mathbf{x}, \mathbf{u}) := \min \mathcal{C}_n(\mathbf{x}, \mathbf{u})$.

Given these high-probability confidence intervals, the system is stable according to Theorem 1 if $v(f(\mathbf{x}, \mathbf{u})) \leq u_n(\mathbf{x}) < v(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{V}(c)$. However, it is intractable to verify this condition directly on the continuous domain without additional, restrictive assumptions about the model. Instead, we consider a discretization of the state space $\mathcal{X}_\tau \subset \mathcal{X}$ into cells, so that $\|\mathbf{x} - [\mathbf{x}]_\tau\|_1 \leq \tau$ holds for all $\mathbf{x} \in \mathcal{X}$. Here, $[\mathbf{x}]_\tau$ denotes the point in $\mathcal{X}_\tau$ with the smallest l_1 distance to $\mathbf{x}$. Given this discretization, we bound the decrease variation on the Lyapunov function for states in $\mathcal{X}_\tau$ and use the Lipschitz continuity to generalize to the continuous state space $\mathcal{X}$.

Theorem 2. *Under Assumptions 1 and 2 with $L_{\Delta v} := L_v L_f (L_\pi + 1) + L_v$, let $\mathcal{X}_\tau$ be a discretization of $\mathcal{X}$ such that $\|\mathbf{x} - [\mathbf{x}]_\tau\|_1 \leq \tau$ for all $\mathbf{x} \in \mathcal{X}$. If, for all $\mathbf{x} \in \mathcal{V}(c) \cap \mathcal{X}_\tau$ with $c > 0$, $\mathbf{u} = \pi(\mathbf{x})$, and for some $n \geq 0$ it holds that $u_n(\mathbf{x}, \mathbf{u}) < v(\mathbf{x}) - L_{\Delta v} \tau$, then $v(f(\mathbf{x}, \pi(\mathbf{x}))) < v(\mathbf{x})$ holds for all $\mathbf{x} \in \mathcal{V}(c)$ with probability at least $(1 - \delta)$ and $\mathcal{V}(c)$ is a region of attraction for* (1) *under policy* π.

The proof is given in Appendix A.1. Theorem 2 states that, given confidence intervals on the statistical model of the dynamics, it is sufficient to check the stricter decrease condition in Theorem 2 on the discretized domain $\mathcal{X}_\tau$ to guarantee the requirements for the region of attraction in the continuous domain in Theorem 1. The bound in Theorem 2 becomes tight as the discretization constant τ and $|v(f(\cdot)) - u_n(\cdot)|$ go to zero. Thus, the discretization constant trades off computation costs for accuracy, while u_n approaches $v(f(\cdot))$ as we obtain more measurement data and the posterior model uncertainty about the dynamics, $\sqrt{\beta_n} \sigma_n$ decreases. The confidence intervals on $v(f(\mathbf{x}, \pi(\mathbf{x})) - v(\mathbf{x})$ and the corresponding estimated region of attraction (red line) can be seen in the bottom half of Fig. 1.

Policy optimization So far, we have focused on estimating the region of attraction for a fixed policy. Safety is a property of states under a fixed policy. This means that the policy directly determines



(a) Initial safe set (in red). (b) Exploration: 15 data points. (c) Final policy after 30 evaluations.

Figure 1: Example application of Algorithm 1. Due to input constraints, the system becomes unstable for large states. We start from an initial, local policy π_0 that has a small, safe region of attraction (red lines) in Fig. 1(a). The algorithm selects safe, informative state-action pairs within $\mathcal{S}_n$ (top, white shaded), which can be evaluated without leaving the region of attraction $\mathcal{V}(c_n)$ (red lines) of the current policy π_n. As we gather more data (blue crosses), the uncertainty in the model decreases (top, background) and we use (3) to update the policy so that it lies within $\mathcal{D}_n$ (top, red shaded) and fulfills the Lyapunov decrease condition. The algorithm converges to the largest safe set in Fig. 1(c). It improves the policy without evaluating unsafe state-action pairs and thereby without system failure.

which states are safe. Specifically, to form a region of attraction all states in the discretizaton $\mathcal{X}_\tau$ within a level set of the Lyapunov function need to fulfill the decrease condition in Theorem 2 that depends on the policy choice. The set of all state-action pairs that fulfill this decrease condition is given by

$$\mathcal{D}_n = \left\{ (\mathbf{x}, \mathbf{u}) \in \mathcal{X}_\tau \times \mathcal{U} \,|\, u_n(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}) < -L_{\Delta v} \tau \right\}, \tag{2}$$

see Fig. 1(c) (top, red shaded). In order to estimate the region of attraction based on this set, we need to commit to a policy. Specifically, we want to pick the policy that leads to the largest possible region of attraction according to Theorem 2. This requires that for each discrete state in $\mathcal{X}_\tau$ the corresponding state-action pair under the policy must be in the set $\mathcal{D}_n$. Thus, we optimize the policy according to

$$\pi_n, c_n = \underset{\pi \in \Pi_L, c \in \mathbb{R}_{>0}}{\operatorname{argmax}} \; c, \quad \text{such that for all } \mathbf{x} \in \mathcal{V}(c) \cap \mathcal{X}_\tau \colon (\mathbf{x}, \pi(\mathbf{x})) \in \mathcal{D}_n. \tag{3}$$

The region of attraction that corresponds to the optimized policy π_n according to (3) is given by $\mathcal{V}(c_n)$, see Fig. 1(b). It is the largest level set of the Lyapunov function for which all state-action pairs $(\mathbf{x}, \pi_n(\mathbf{x}))$ that correspond to discrete states within $\mathcal{V}(c_n) \cap \mathcal{X}_\tau$ are contained in $\mathcal{D}_n$. This means that these state-action pairs fulfill the requirements of Theorem 2 and $\mathcal{V}(c_n)$ is a region of attraction of the true system under policy π_n. The following theorem is thus a direct consequence of Theorem 2 and (3).

Theorem 3. *Let $\mathcal{R}_{\pi_n}$ be the true region of attraction of* (1) *under the policy π_n. For any $\delta \in (0, 1)$, we have with probability at least $(1 - \delta)$ that $\mathcal{V}(c_n) \subseteq \mathcal{R}_{\pi_n}$ for all $n > 0$.*

Thus, when we optimize the policy subject to the constraint in (3) the estimated region of attraction is always an inner approximation of the true region of attraction. However, solving the optimization problem in (3) is intractable in general. We approximate the policy update step in Sec. 4.

Collecting measurements Given these stability guarantees, it is natural to ask how one might obtain data points in order to improve the model of $g(\cdot)$ and thus efficiently increase the region of attraction. This question is difficult to answer in general, since it depends on the property of the statistical model. In particular, for general statistical models it is often not clear whether the confidence intervals contract sufficiently quickly. In the following, we make additional assumptions about the model and reachability within $\mathcal{V}(c_n)$ in order to provide exploration guarantees. These assumptions allow us to highlight fundamental requirements for safe data acquisition and that safe exploration is possible.

We assume that the unknown model errors $g(\cdot)$ have bounded norm in a reproducing kernel Hilbert space (RKHS, [33]) corresponding to a differentiable kernel k, $\|g(\cdot)\|_k \leq B_g$. These are a class of well-behaved functions of the form $g(\mathbf{z}) = \sum_{i=0}^{\infty} \alpha_i k(\mathbf{z}_i, \mathbf{z})$ defined through representer points $\mathbf{z}_i$ and weights α_i that decay sufficiently fast with i. This assumption ensures that g satisfies the Lipschitz property in Assumption 1, see [28]. Moreover, with $\beta_n = B_g + 4\sigma\sqrt{\gamma_n + 1 + \ln(1/\delta)}$ we can use GP models for the dynamics that fulfill Assumption 2 if the state if fully observable and the measurement noise is σ-sub-Gaussian (e.g., bounded in $[-\sigma, \sigma]$), see [34]. Here γ_n is the information capacity. It corresponds to the amount of mutual information that can be obtained about g from nq measurements, a measure of the size of the function class encoded by the model. The information capacity has a sublinear dependence on n for common kernels and upper bounds can be computed efficiently [35]. More details about this model are given in Appendix A.2.

In order to quantify the exploration properties of our algorithm, we consider a discrete action space $\mathcal{U}_\tau \subset \mathcal{U}$. We define exploration as the number of state-action pairs in $\mathcal{X}_\tau \times \mathcal{U}_\tau$ that we can safely learn about without leaving the true region of attraction. Note that despite this discretization, the policy takes values on the continuous domain. Moreover, instead of using the confidence intervals directly as in (3), we consider an algorithm that uses the Lipschitz constants to slowly expand the safe set. We use this in our analysis to quantify the ability to generalize beyond the current safe set. In practice, nearby states are sufficiently correlated under the model to enable generalization using (2).

Suppose we are given a set $\mathcal{S}_0$ of state-action pairs about which we can learn safely. Specifically, this means that we have a policy such that, for any state-action pair $(\mathbf{x}, \mathbf{u})$ in $\mathcal{S}_0$, if we apply action $\mathbf{u}$ in state $\mathbf{x}$ and then apply actions according to the policy, the state converges to the origin. Such a set can be constructed using the initial policy π_0 from Sec. 2 as $\mathcal{S}_0 = \{(\mathbf{x}, \pi_0(\mathbf{x})) \,|\, \mathbf{x} \in \mathcal{S}_0^x\}$. Starting from this set, we want to update the policy to expand the region of attraction according to Theorem 2. To this end, we use the confidence intervals on $v(f(\cdot))$ for states inside $\mathcal{S}_0$ to determine state-action pairs that fulfill the decrease condition. We thus redefine $\mathcal{D}_n$ for the exploration analysis to

$$\mathcal{D}_n = \bigcup_{(\mathbf{x},\mathbf{u}) \in \mathcal{S}_{n-1}} \left\{ \mathbf{z}' \in \mathcal{X}_\tau \times \mathcal{U}_\tau \,|\, u_n(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}) + L_{\Delta v} \|\mathbf{z}' - (\mathbf{x}, \mathbf{u})\|_1 < -L_{\Delta v}\tau \right\}. \tag{4}$$

This formulation is equivalent to (2), except that it uses the Lipschitz constant to generalize safety. Given $\mathcal{D}_n$, we can again find a region of attraction $\mathcal{V}(c_n)$ by committing to a policy according to (3). In order to expand this region of attraction effectively we need to decrease the posterior model uncertainty about the dynamics of the GP by collecting measurements. However, to ensure safety as outlined in Sec. 2, we are not only restricted to states within $\mathcal{V}(c_n)$, but also need to ensure that the state after taking an action is safe; that is, the dynamics map the state back into the region of attraction $\mathcal{V}(c_n)$. We again use the Lipschitz constant in order to determine this set,

$$\mathcal{S}_n = \bigcup_{\mathbf{z} \in \mathcal{S}_{n-1}} \left\{ \mathbf{z}' \in \mathcal{V}(c_n) \cap \mathcal{X}_\tau \times \mathcal{U}_\tau \,|\, u_n(\mathbf{z}) + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 \leq c_n \right\}. \tag{5}$$

The set $\mathcal{S}_n$ contains state-action pairs that we can safely evaluate under the current policy π_n without leaving the region of attraction, see Fig. 1 (top, white shaded).

What remains is to define a strategy for collecting data points within $\mathcal{S}_n$ to effectively decrease model uncertainty. We specifically focus on the high-level requirements for any exploration scheme without committing to a specific method. In practice, any (model-based) exploration strategy that aims to decrease model uncertainty by driving the system to specific states may be used. Safety can be ensured by picking actions according to π_n whenever the exploration strategy reaches the boundary of the safe region $\mathcal{V}(c_n)$; that is, when $u_n(\mathbf{x}, \mathbf{u}) > c_n$. This way, we can use π_n as a backup policy for exploration.

The high-level goal of the exploration strategy is to shrink the confidence intervals at state-action pairs $\mathcal{S}_n$ in order to expand the safe region. Specifically, the exploration strategy should aim to visit state-action pairs in $\mathcal{S}_n$ at which we are the most uncertain about the dynamics; that is, where the confidence interval is the largest:

$$(\mathbf{x}_n, \mathbf{u}_n) = \underset{(\mathbf{x},\mathbf{u}) \in \mathcal{S}_n}{\operatorname{argmax}}\, u_n(\mathbf{x}, \mathbf{u}) - l_n(\mathbf{x}, \mathbf{u}). \tag{6}$$

As we keep collecting data points according to (6), we decrease the uncertainty about the dynamics for different actions throughout the region of attraction and adapt the policy, until eventually we

Algorithm 1 SafeLyapunovLearning

```
Input: Initial safe policy $\pi_0$, dynamics model $\mathcal{GP}(\mu(\mathbf{z}), k(\mathbf{z}, \mathbf{z}'))$
for all $n = 1, \dots$ do
    Compute policy $\pi_n$ via SGD on (7)
    $c_n = \operatorname{argmax}_c c$, such that $\forall \mathbf{x} \in \mathcal{V}(c_n) \cap \mathcal{X}_\tau \colon u_n(\mathbf{x}, \pi_n(\mathbf{x})) - v(\mathbf{x}) < -L_{\Delta v}\tau$
    $\mathcal{S}_n = \{(\mathbf{x}, \mathbf{u}) \in \mathcal{V}(c_n) \times \mathcal{U}_\tau \,|\, u_n(\mathbf{x}, \mathbf{u}) \leq c_n\}$
    Select $(\mathbf{x}_n, \mathbf{u}_n)$ within $\mathcal{S}_n$ using (6) and drive system there with backup policy $\pi_n$
    Update GP with measurements $f(\mathbf{x}_n, \mathbf{u}_n) + \epsilon_n$
```

have gathered enough information in order to expand it. While (6) implicitly assumes that any state within $\mathcal{V}(c_n)$ can be reached by the exploration policy, it achieves the high-level goal of any exploration algorithm that aims to reduce model uncertainty. In practice, any safe exploration scheme is limited by unreachable parts of the state space.

We compare the active learning scheme in (6) to an oracle baseline that starts from the same initial safe set $\mathcal{S}_0$ and knows $v(f(\mathbf{x}, \mathbf{u}))$ up to ϵ accuracy within the safe set. The oracle also uses knowledge about the Lipschitz constants and the optimal policy in Π_L at each iteration. We denote the set that this baseline manages to determine as safe with $\overline{R}_\epsilon(\mathcal{S}_0)$ and provide a detailed definition in Appendix A.3.

Theorem 4. *Assume σ-sub-Gaussian measurement noise and that the model error $g(\cdot)$ in* (1) *has RKHS norm smaller than B_g. Under the assumptions of Theorem 2, with $\beta_n = B_g + 4\sigma\sqrt{\gamma_n + 1 + \ln(1/\delta)}$, and with measurements collected according to* (6)*, let n^* be the smallest positive integer so that $\frac{n^*}{\beta^2_{n^*}\gamma_{n^*}} \geq \frac{Cq(|\overline{R}(\mathcal{S}_0)|+1)}{L_v^2\epsilon^2}$ where $C = 8/\log(1+\sigma^{-2})$. Let $\mathcal{R}_\pi$ be the true region of attraction of* (1) *under a policy π. For any $\epsilon > 0$, and $\delta \in (0,1)$, the following holds jointly with probability at least* $(1-\delta)$ *for all $n > 0$:*

(i) $\mathcal{V}(c_n) \subseteq \mathcal{R}_{\pi_n}$ *(ii)* $f(\mathbf{x}, \mathbf{u}) \in \mathcal{R}_{\pi_n} \; \forall (\mathbf{x}, \mathbf{u}) \in \mathcal{S}_n$. *(iii)* $\overline{R}_\epsilon(\mathcal{S}_0) \subseteq \mathcal{S}_n \subseteq \overline{R}_0(\mathcal{S}_0)$.

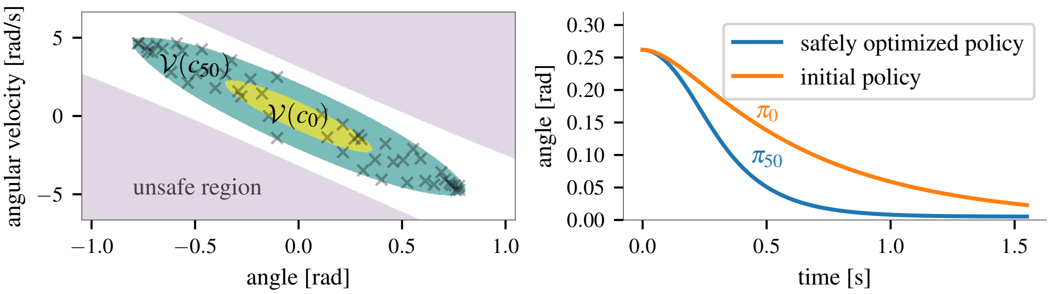
Theorem 4 states that, when selecting data points according to (6), the estimated region of attraction $\mathcal{V}(c_n)$ is (i) contained in the true region of attraction under the current policy and (ii) selected data points do not cause the system to leave the region of attraction. This means that any exploration method that considers the safety constraint (5) is able to safely learn about the system without leaving the region of attraction. The last part of Theorem 4, (iii), states that after a finite number of data points n^* we achieve at least the exploration performance of the oracle baseline, while we do not classify unsafe state-action pairs as safe. This means that the algorithm explores the largest region of attraction possible for a given Lyapunov function with residual uncertaint about $v(f(\cdot))$ smaller than ϵ. Details of the comparison baseline are given in the appendix. In practice, this means that any exploration method that manages to reduce the maximal uncertainty about the dynamics within $\mathcal{S}_n$ is able to expand the region of attraction.

An example run of repeatedly evaluating (6) for a one-dimensional state-space is shown in Fig. 1. It can be seen that, by only selecting data points within the current estimate of the region of attraction, the algorithm can efficiently optimize the policy and expand the safe region over time.

4 Practical Implementation and Experiments

In the previous section, we have given strong theoretical results on safety and exploration for an idealized algorithm that can solve (3). In this section, we provide a practical variant of the theoretical algorithm in the previous section. In particular, while we retain safety guarantees, we sacrifice exploration guarantees to obtain a more practical algorithm. This is summarized in Algorithm 1.

The policy optimization problem in (3) is intractable to solve and only considers safety, rather than a performance metric. We propose to use an approximate policy update that that maximizes approximate performance while providing stability guarantees. It proceeds by optimizing the policy first and then computes the region of attraction $\mathcal{V}(c_n)$ for the new, fixed policy. This does not impact safety, since data is still only collected inside the region of attraction. Moreover, should the optimization fail and the region of attraction decrease, one can always revert to the previous policy, which is guaranteed to be safe.



(a) Estimated safe set.

(b) State trajectory (lower is better).

Figure 2: Optimization results for an inverted pendulum. Fig. 2(a) shows the initial safe set (yellow) under the policy π_0, while the green region represents the estimated region of attraction under the optimized neural network policy. It is contained within the true region of attraction (white). Fig. 2(b) shows the improved performance of the safely learned policy over the policy for the prior model.

In our experiments, we use approximate dynamic programming [36] to capture the performance of the policy. Given a policy π_θ with parameters θ, we compute an estimate of the cost-to-go $J_{\pi_\theta}(\cdot)$ for the mean dynamics μ_n based on the cost $r(\mathbf{x}, \mathbf{u}) \geq 0$. At each state, $J_{\pi_\theta}(\mathbf{x})$ is the sum of γ-discounted rewards encountered when following the policy π_θ. The goal is to adapt the parameters of the policy for minimum cost as measured by J_{π_θ}, while ensuring that the safety constraint on the worst-case decrease on the Lyapunov function in Theorem 2 is not violated. A Lagrangian formulation to this constrained optimization problem is

$$\pi_n = \operatorname*{argmin}_{\pi_\theta \in \Pi_L} \sum_{\mathbf{x} \in \mathcal{X}_\tau} r(\mathbf{x}, \pi_\theta(\mathbf{x})) + \gamma J_{\pi_\theta}(\mu_{n-1}(\mathbf{x}, \pi_\theta(\mathbf{x})) + \lambda\Big(u_n(\mathbf{x}, \pi_\theta(\mathbf{x})) - v(\mathbf{x}) + L_{\Delta v}\tau\Big), \quad (7)$$

where the first term measures long-term cost to go and $\lambda \geq 0$ is a Lagrange multiplier for the safety constraint from Theorem 2. In our experiments, we use the value function as a Lyapunov function candidate, $v = J$ with $r(\cdot, \cdot) \geq 0$, and set $\lambda = 1$. In this case, (7) corresponds to an high-probability upper bound on the cost-to-go given the uncertainty in the dynamics. This is similar to worst-case performance formulations found in robust MDPs [9, 10], which consider worst-case value functions given parametric uncertainty in MDP transition model. Moreover, since $L_{\Delta v}$ depends on the Lipschitz constant of the policy, this simultaneously serves as a regularizer on the parameters θ.

To verify safety, we use the GP confidence intervals l_n and u_n directly, as in (2). We also use confidence to compute $\mathcal{S}_n$ for the active learning scheme, see Algorithm 1, Line 5. In practice, we do not need to compute the entire set $\mathcal{S}_n$ to solve (3), but can use a global optimization method or even a random sampling scheme within $\mathcal{V}(c_n)$ to find suitable state-actions. Moreover, measurements for actions that are far away from the current policy are unlikely to expand $\mathcal{V}(c_n)$, see Fig. 1(c). As we optimize (7) via gradient descent, the policy changes only locally. Thus, we can achieve better data-efficiency by restricting the exploratory actions $\mathbf{u}$ with $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}_n$ to be close to π_n, $\mathbf{u} \in [\pi_n(\mathbf{x}) - \bar{u}, \pi_n(\mathbf{x}) + \bar{u}]$ for some constant $\bar{u}$.

Computing the region of attraction by verifying the stability condition on a discretized domain suffers from the curse of dimensionality. However, it is not necessary to update policies in real time. In particular, since any policy that is returned by the algorithm is provably safe within some level set, any of these policies can be used safely for an arbitrary number of time steps. To scale this method to higher-dimensional system, one would have to consider an adaptive discretization for the verification as in [27].

Experiments A Python implementation of Algorithm 1 and the experiments based on TensorFlow [37] and GPflow [38] is available at `https://github.com/befelix/safe_learning`.

We verify our approach on an inverted pendulum benchmark problem. The true, continuous-time dynamics are given by $ml^2\ddot{\psi} = gml\sin(\psi) - \lambda\dot{\psi} + u$, where ψ is the angle, m the mass, g the gravitational constant, and u the torque applied to the pendulum. The control torque is limited, so that the pendulum necessarily falls down beyond a certain angle. We use a GP model for the *discrete-time* dynamics, where the mean dynamics are given by a linearized and discretized model of the true dynamics that considers a wrong, lower mass and neglects friction. As a result, the optimal policy for the mean dynamics does not perform well and has a small region of attraction as it underactuates the system. We use a combination of linear and Matérn kernels in order to capture the model errors that result from parameter and integration errors.

For the policy, we use a neural network with two hidden layers and 32 neurons with ReLU activations each. We compute a conservative estimate of the Lipschitz constant as in [30]. We use standard approximate dynamic programming with a quadratic, normalized cost $r(\mathbf{x}, \mathbf{u}) = \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u}$, where $\mathbf{Q}$ and $\mathbf{R}$ are positive-definite, to compute the cost-to-go J_{π_θ}. Specifically, we use a piecewise-linear triangulation of the state-space as to approximate J_{π_θ}, see [39]. This allows us to quickly verify the assumptions that we made about the Lyapunov function in Sec. 2 using a graph search. In practice, one may use other function approximators. We optimize the policy via stochastic gradient descent on (7).

The theoretical confidence intervals for the GP model are conservative. To enable more data-efficient learning, we fix $\beta_n = 2$. This corresponds to a high-probability decrease condition per-state, rather than jointly over the state space. Moreover, we use local Lipschitz constants of the Lyapunov function rather than the global one. While this does not affect guarantees, it greatly speeds up exploration.

For the initial policy, we use approximate dynamic programming to compute the optimal policy for the prior mean dynamics. This policy is unstable for large deviations from the initial state and has poor performance, as shown in Fig. 2(b). Under this initial, suboptimal policy, the system is stable within a small region of the state-space Fig. 2(a). Starting from this initial safe set, the algorithm proceeds to collect safe data points and improve the policy. As the uncertainty about the dynamics decreases, the policy improves and the estimated region of attraction increases. The region of attraction after 50 data points is shown in Fig. 2(a). The resulting set $\mathcal{V}(c_n)$ is contained within the true safe region of the optimized policy π_n. At the same time, the control performance improves drastically relative to the initial policy, as can be seen in Fig. 2(b). Overall, the approach enables safe learning about dynamic systems, as all data points collected during learning are safely collected under the current policy.

5 Conclusion

We have shown how classical reinforcement learning can be combined with safety constraints in terms of stability. Specifically, we showed how to safely optimize policies and give stability certificates based on statistical models of the dynamics. Moreover, we provided theoretical safety and exploration guarantees for an algorithm that can drive the system to desired state-action pairs during learning. We believe that our results present an important first step towards safe reinforcement learning algorithms that are applicable to real-world problems.

Acknowledgments

This research was supported by SNSF grant 200020_159557, the Max Planck ETH Center for Learning Systems, NSERC grant RGPIN-2014-04634, and the Ontario Early Researcher Award.

References

[1] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. MIT press, 1998.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv:1606.06565 [cs]*, 2016.

[4] Hassan K. Khalil and J. W. Grizzle. *Nonlinear systems*, volume 3. Prentice Hall, 1996.

[5] Martin Pecka and Tomas Svoboda. Safe exploration techniques for reinforcement learning – an overview. In *Modelling and Simulation for Autonomous Systems*, pages 357–375. Springer, 2014.

[6] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 16:1437–1480, 2015.

[7] Stefano P. Coraluppi and Steven I. Marcus. Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes. *Automatica*, 35(2):301–309, 1999.

[8] Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *J. Artif. Intell. Res.(JAIR)*, 24:81–108, 2005.

[9] Aviv Tamar, Shie Mannor, and Huan Xu. Scaling Up Robust MDPs by Reinforcement Learning. In *Proc. of the International Conference on Machine Learning (ICML)*, 2014.

[10] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust Markov Decision Processes. *Mathematics of Operations Research*, 38(1):153–183, 2012.

[11] Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in Markov decision processes. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 1711–1718, 2012.

[12] Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. pages 4305–4313, 2016.

[13] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225, 2006.

[14] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proc. of the International Conference on Machine Learning (ICML)*, 2017.

[15] Jonas Mockus. *Bayesian approach to global optimization*, volume 37 of *Mathematics and Its Applications*. Springer, Dordrecht, 1989.

[16] Carl Edward Rasmussen and Christopher K.I Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge MA, 2006.

[17] Jens Schreiter, Duy Nguyen-Tuong, Mona Eberts, Bastian Bischoff, Heiner Markert, and Marc Toussaint. Safe exploration for active learning with Gaussian processes. In *Machine Learning and Knowledge Discovery in Databases*, number 9286, pages 133–149. Springer International Publishing, 2015.

[18] Yanan Sui, Alkis Gotovos, Joel W. Burdick, and Andreas Krause. Safe exploration for optimization with Gaussian processes. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 997–1005, 2015.

[19] Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with Gaussian processes. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 493–496, 2016.

[20] J. Garcia and F. Fernandez. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, pages 515–564, 2012.

[21] Alexander Hans, Daniel Schneegaß, Anton Maximilian Schäfer, and Steffen Udluft. Safe exploration for reinforcement learning. In *Proc. of the European Symposium on Artificial Neural Networks (ESANN)*, pages 143–148, 2008.

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

[23] Dorsa Sadigh and Ashish Kapoor. Safe control under uncertainty with Probabilistic Signal Temporal Logic. In *Proc. of Robotics: Science and Systems*, 2016.

[24] Chris J. Ostafew, Angela P. Schoellig, and Timothy D. Barfoot. Robust constrained learning-based NMPC enabling reliable mobile robot path tracking. *The International Journal of Robotics Research (IJRR)*, 35(13):1547–1536, 2016.

[25] Anil Aswani, Humberto Gonzalez, S. Shankar Sastry, and Claire Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, 2013.

[26] Anayo K. Akametalu, Shahab Kaynama, Jaime F. Fisac, Melanie N. Zeilinger, Jeremy H. Gillula, and Claire J. Tomlin. Reachability-based safe learning with Gaussian processes. In *Proc. of the IEEE Conference on Decision and Control (CDC)*, pages 1424–1431, 2014.

[27] Ruxandra Bobiti and Mircea Lazar. A sampling approach to finding Lyapunov functions for nonlinear discrete-time systems. In *Proc. of the European Control Conference (ECC)*, pages 561–566, 2016.

[28] Felix Berkenkamp, Riccardo Moriconi, Angela P. Schoellig, and Andreas Krause. Safe learning of regions of attraction in nonlinear systems with Gaussian processes. In *Proc. of the Conference on Decision and Control (CDC)*, pages 4661–4666, 2016.

[29] Julia Vinogradska, Bastian Bischoff, Duy Nguyen-Tuong, Henner Schmidt, Anne Romer, and Jan Peters. Stability of controllers for Gaussian process forward models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 545–554, 2016.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2014.

[31] Huijuan Li and Lars Grüne. Computation of local ISS Lyapunov functions for discrete-time systems via linear programming. *Journal of Mathematical Analysis and Applications*, 438(2):701–719, 2016.

[32] Peter Giesl and Sigurdur Hafstein. Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems, Series B*, 20(8):2291–2337, 2015.

[33] Bernhard Schölkopf. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2002.

[34] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. *arXiv preprint arXiv:1704.00445*, 2017.

[35] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[36] Warren B. Powell. *Approximate dynamic programming: solving the curses of dimensionality*. John Wiley & Sons, 2007.

[37] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv:1603.04467 [cs]*, 2016.

[38] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: a Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017.

[39] Scott Davies. Multidimensional triangulation and interpolation for reinforcement learning. In *Proc. of the Conference on Neural Information Processing Systems (NIPS)*, pages 1005–1011, 1996.

[40] Andreas Christmann and Ingo Steinwart. *Support Vector Machines*. Information Science and Statistics. Springer, New York, NY, 2008.

A Proofs

A.1 Stability verification

Lemma 1. *Using Assumptions 1 and 2, let $\mathcal{X}_\tau$ be a discretization of $\mathcal{X}$ such that $\|\mathbf{x} - [\mathbf{x}]_\tau\|_1 \leq \tau$ for all $\mathbf{x} \in \mathcal{X}$. Then, for all $\mathbf{x} \in \mathcal{X}$, we have with probability at least $1 - \delta$ that*

$$\left| v(\mu_{n-1}([\mathbf{z}]_\tau)) - v([\mathbf{x}]_\tau) - \big(v(f(\mathbf{z})) - v(\mathbf{x})\big) \right| \leq L_v \beta_n \sigma_{n-1}([\mathbf{z}]_\tau) + (L_v L_f (L_\pi + 1) + L_v)\tau, \quad (8)$$

where $\mathbf{z} = (\mathbf{x}, \pi(\mathbf{x}))$ and $[\mathbf{z}]_\tau = ([\mathbf{x}]_\tau, \pi([\mathbf{x}]_\tau))$.

Proof. Let $\mathbf{z} = (\mathbf{x}, \pi(\mathbf{x}))$, $[\mathbf{z}]_\tau = ([\mathbf{x}]_\tau, \pi([\mathbf{x}]_\tau))$, and $\mu = \mu_{n-1}$, $\sigma = \sigma_{n-1}$. Then we have that

$$\begin{aligned}
&\left| v(\mu([\mathbf{z}]_\tau)) - v([\mathbf{x}]_\tau) - \big(v(f(\mathbf{z})) - v(\mathbf{x})\big) \right|, \\
&= \left| v(\mu([\mathbf{z}]_\tau)) - v([\mathbf{x}]_\tau) - v(f(\mathbf{z})) + v(\mathbf{x}) \right|, \\
&= \left| v(\mu([\mathbf{z}]_\tau)) - v(f([\mathbf{z}]_\tau)) + v(f([\mathbf{z}]_\tau)) - v(f(\mathbf{z})) + v(\mathbf{x}) - v([\mathbf{x}]_\tau) \right|, \\
&\leq \left| v(\mu([\mathbf{z}]_\tau)) - v(f([\mathbf{z}]_\tau)) \right| + \left| v(f([\mathbf{z}]_\tau)) - v(f(\mathbf{z})) \right| + \left| v(\mathbf{x}) - v([\mathbf{x}]_\tau) \right|, \\
&\leq L_v \|\mu([\mathbf{z}]_\tau) - f([\mathbf{z}]_\tau)\|_1 + L_v \|f([\mathbf{z}]_\tau) - f(\mathbf{z})\|_1 + L_v \|\mathbf{x} - [\mathbf{x}]_\tau\|_1, \\
&\leq L_v \beta_n \sigma([\mathbf{z}]_\tau) + L_v L_f \|[\mathbf{z}]_\tau - \mathbf{z}\|_1 + L_v \|\mathbf{x} - [\mathbf{x}]_\tau\|_1,
\end{aligned}$$

where the last three inequalities follow from Assumptions 1 and 2 to last inequality follows from Lemma 3. The result holds with probability at least $1 - \delta$. By definition of the discretization and the policy class Π_L we have on each grid cell that

$$\begin{aligned}
\|\mathbf{z} - [\mathbf{z}]_\tau\|_1 &= \|\mathbf{x} - [\mathbf{x}]_\tau\|_1 + \|\pi(\mathbf{x}) - \pi([\mathbf{x}]_\tau)\|_1, \\
&\leq \tau + L_\pi \|\mathbf{x} - [\mathbf{x}]_\tau\|_1, \\
&\leq (L_\pi + 1)\tau,
\end{aligned}$$

where the equality in the first step follows from the definition of the 1-norm. Plugging this into the previous bound yields

$$\left| v(\mu([\mathbf{z}]_\tau)) - v([\mathbf{x}]_\tau) - \big(v(f(\mathbf{z})) - v(\mathbf{x})\big) \right| \leq L_v \beta_n \sigma([\mathbf{z}]_\tau) + \left(L_v L_f (1 + L_\pi) + L_v\right)\tau,$$

which completes the proof. □

Lemma 2. $v(f(\mathbf{x}, \mathbf{u})) \in \mathcal{Q}_n$ *holds for all* $\mathbf{x} \in \mathcal{X}$, $\mathbf{u} \in \mathcal{U}$, *and* $n > 0$ *with probability at least* $(1 - \delta)$.

Proof. The proof is analogous to Lemma 1 and follows from Assumptions 1 and 2. □

Corollary 1. $v(f(\mathbf{x}, \mathbf{u})) \in \mathcal{C}_n$ *holds for all* $\mathbf{x} \in \mathcal{X}$, $\mathbf{u} \in \mathcal{U}$, *and* $n > 0$ *with probability at least* $(1 - \delta)$.

Proof. Direct consequence of the fact that Lemma 2 holds jointly for all $n > 0$ with probability at least $1 - \delta$. □

Lemma 1 show that the decrease on the Lyapunov function on the discrete grid $\mathcal{X}_\tau$ is close to that on the continuous domain $\mathcal{X}$. Given these confidence intervals, we can now establish the region of attraction using Theorem 1:

Theorem 2. *Under Assumptions 1 and 2 with $L_{\Delta v} := L_v L_f (L_\pi + 1) + L_v$, let $\mathcal{X}_\tau$ be a discretization of $\mathcal{X}$ such that $\|\mathbf{x} - [\mathbf{x}]_\tau\|_1 \leq \tau$ for all $\mathbf{x} \in \mathcal{X}$. If, for all $\mathbf{x} \in \mathcal{V}(c) \cap \mathcal{X}_\tau$ with $c > 0$, $\mathbf{u} = \pi(\mathbf{x})$, and for some $n \geq 0$ it holds that $u_n(\mathbf{x}, \mathbf{u}) < v(\mathbf{x}) - L_{\Delta v}\tau$, then $v(f(\mathbf{x}, \pi(\mathbf{x}))) < v(\mathbf{x})$ holds for all $\mathbf{x} \in \mathcal{V}(c)$ with probability at least $(1 - \delta)$ and $\mathcal{V}(c)$ is a region of attraction for (1) under policy π.*

Proof. Using Lemma 1 it holds that $v(f(\mathbf{x}, \pi(\mathbf{x})) - v(\mathbf{x}) < 0$ for all continuous states $\mathbf{x} \in \mathcal{V}(c)$ with probability at least $1 - \delta$, since all discrete states $\mathbf{x}_\tau \in \mathcal{V}(c) \cap \mathcal{X}$ fulfill Theorem 2. Thus we can use Theorem 1 to conclude that $\mathcal{V}(c)$ is a region of attraction for (1). □

Theorem 3. *Let $\mathcal{R}_{\pi_n}$ be the true region of attraction of (1) under the policy π_n. For any $\delta \in (0, 1)$, we have with probability at least $(1 - \delta)$ that $\mathcal{V}(c_n) \subseteq \mathcal{R}_{\pi_n}$ for all $n > 0$.*

Proof. Following the definition of $\mathcal{D}_n$ in (2), it is clear from the constraint in the optimization problem (3) that for all $\mathbf{x} \in \mathcal{D}_n$ it holds that $(\mathbf{x}, \pi_n(\mathbf{x})) \in \mathcal{D}_n$ or, equivalently that $u_n(\mathbf{x}, \pi(\mathbf{x})) - v(\mathbf{x}) < -L_{\Delta v}\tau$, see (2). The result $\mathcal{V}(c_n) \subseteq \mathcal{R}_{\pi_n}$ then follows from Theorem 2. □

Note that the initialization of the confidence intervals $\mathcal{Q}_0$ ensures that the decrease condition is always fulfilled for the initial policy.

A.2 Gaussian process model

One particular assumption that satisfies both the Lipschitz continuity and allows us to use GPs as a model of the dynamics is that the model errors $g(\mathbf{x}, \mathbf{u})$ live in some reproducing kernel Hilbert space (RKHS, [40]) corresponding to a differentiable kernel k and have RKHS norm smaller than B_g [35]. In our theoretical analysis, we use this assumption to prove exploration guarantees.

A $\mathcal{GP}(\mu(\mathbf{z}), k(\mathbf{z}, \mathbf{z}'))$ is a distribution over well-behaved, smooth functions $f \colon \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ (see Remark 1 for the vector-case, $\mathbb{R}^q$) that is parameterized by a mean function μ and a covariance function (kernel) k, which encodes assumptions about the functions [16]. In our case, the mean is given by the prior model h, while the kernel corresponds to the one in the RKHS. Given noisy measurements of the dynamics, $\hat{f}(\mathbf{z}) = f(\mathbf{z}) + \epsilon$ with $\mathbf{z} = (\mathbf{x}, \mathbf{u})$ at locations $A_n = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$, corrupted by independent, Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ (we relax the Gaussian noise assumption in our analysis), the posterior is a GP distribution again with mean, $\mu_n(\mathbf{z}) = \mathbf{k}_n(\mathbf{z})^{\mathrm{T}}(\mathbf{K}_n + \sigma^2\mathbf{I})^{-1}\mathbf{y}_n$, covariance $k_n(\mathbf{z}, \mathbf{z}') = k(\mathbf{z}, \mathbf{z}') - \mathbf{k}_n(\mathbf{z})^{\mathrm{T}}(\mathbf{K}_n + \sigma^2\mathbf{I})^{-1}\mathbf{k}_n(\mathbf{z}')$, and variance $\sigma_n^2(\mathbf{z}) = k_n(\mathbf{z}, \mathbf{z})$. The vector $\mathbf{y}_n = [\hat{f}(\mathbf{z}_1) - h(\mathbf{z}_1), \dots, \hat{f}(\mathbf{z}_n) - h(\mathbf{z}_n)]^{\mathrm{T}}$ contains observed, noisy deviations from the mean, $\mathbf{k}_n(\mathbf{z}) = [k(\mathbf{z}, \mathbf{z}_1), \dots, k(\mathbf{z}, \mathbf{z}_n)]$ contains the covariances between the test input $\mathbf{z}$ and the data points in $\mathcal{D}_n$, $\mathbf{K}_n \in \mathbb{R}^{n \times n}$ has entries $[\mathbf{K}_n]_{(i,j)} = k(\mathbf{z}_i, \mathbf{z}_j)$, and $\mathbf{I}$ is the identity matrix.

Remark 1. *In the case of multiple output dimensions ($q > 1$), we consider a function with one-dimensional output $f'(\mathbf{x}, \mathbf{u}, i) \colon \mathcal{X} \times \mathcal{U} \times \mathcal{I} \to \mathbb{R}$, with the output dimension indexed by $i \in \mathcal{I} = \{1, \dots, q\}$. This allows us to use the standard definitions of the RKHS norm and GP model. In this case, we define the GP posterior distribution as $\mu_n(\mathbf{z}) = [\mu_n(\mathbf{z}, 1), \dots, \mu_n(\mathbf{z}, q)]^{\mathrm{T}}$ and $\sigma_n(\mathbf{z}) = \sum_{1 \leq i \leq q} \sigma_n(\mathbf{z}, i)$, where the unusual definition of the standard deviation is used in Lemma 3.*

Given the previous assumptions, it follows from [28, Lemma 2] that the dynamics in (1) are Lipschitz continuous with Lipschitz constant $L_f = L_h + L_g$, where L_g depends on the properties (smoothness) of the kernel.

Moreover, we can construct high-probability confidence intervals on the dynamics in (1) that fulfill Assumption 2 using the GP model.

Lemma 3. *([35, Theorem 6]) Assume σ-sub-Gaussian noise and that the model error $g(\cdot)$ in (1) has RKHS norm bounded by B_g. Choose $\beta_n = B_g + 4\sigma\sqrt{\gamma_n + 1 + \ln(1/\delta)}$. Then, with probability at least $1 - \delta$, $\delta \in (0, 1)$, for all $n \geq 1$, $\mathbf{x} \in \mathcal{X}$, and $\mathbf{u} \in \mathcal{U}$ it holds that $\|f(\mathbf{x}, \mathbf{u}) - \mu_{n-1}(\mathbf{x}, \mathbf{u})\|_1 \leq \beta_n \sigma_{n-1}(\mathbf{x}, \mathbf{u})$.*

Proof. From [34, Theorem 2] it follows that $|f(\mathbf{x}, \mathbf{u}, i) - \mu_{n-1}(\mathbf{x}, \mathbf{u}, i)| \leq \beta_n \sigma_n(\mathbf{x}, \mathbf{u}, i)$ holds with probability at least $1 - \delta$ for all $1 \leq i \leq q$. Following Remark 1, we can model the multi-output function as a single-output function over an extended parameter space. Thus the result directly transfers by definition of the one norm and our definition of σ_n for multiple output dimensions in Remark 1. Note that by iteration n we have obtained nq measurements in the information capacity γ_n. $\square$

That is, the true dynamics are contained within the GP posterior confidence intervals with high probability. The bound depends on the information capacity,

$$\gamma_n = \max_{A \subset \mathcal{X} \times \mathcal{U} \times \mathcal{I} \colon |A| = nq} \mathrm{I}(\mathbf{y}_A; \mathbf{f}_A), \tag{9}$$

which is the maximum mutual information that could be gained about the dynamics f from samples. The information capacity has a sublinear dependence on $n (\neq t)$ for many commonly used kernels such as the linear, squared exponential, and Matérn kernels and it can be efficiently and accurately approximated [35]. Note that we explicitly account for the q measurements that we get for each of the q states in (9).

Remark 2. *The GP model assumes Gaussian noise, while Lemma 3 considers σ-sub-Gaussian noise. Moreover, we consider functions with bounded RKHS norm, rather than samples from a GP. Lemma 3 thus states that even though we make different assumptions than the model, the confidence intervals are conservative enough to capture the true function with high probability.*

A.3 Safe exploration

Remark 3. *In the following we assume that $\mathcal{D}_n$ and $\mathcal{S}_n$ are defined as in* (4) *and* (5)*.*

Baseline As a baseline, we consider a class of algorithms that know about the Lipschitz continuity properties of v, f, and π. In addition, we can learn about $v(f(\mathbf{x}, \mathbf{u}))$ up to some arbitrary statistical accuracy ϵ by visiting state $\mathbf{x}$ and obtaining a measurement for the next state after applying action $\mathbf{u}$, but face the safety restrictions defined in Sec. 2. Suppose we are given a set $\mathcal{S}$ of state-action pairs about which we can learn safely. Specifically, this means that we have a policy such that, for any state-action pair $(\mathbf{x}, \mathbf{u})$ in $\mathcal{S}$, if we apply action $\mathbf{u}$ in state $\mathbf{x}$ and then apply actions according to the policy, the state converges to the origin. Such a set can be constructed using the initial policy π_0 from Sec. 2 as $\mathcal{S}_0 = \{(\mathbf{x}, \pi_0(\mathbf{x})) \,|\, \mathbf{x} \in \mathcal{S}_0^x\}$.

The goal of the algorithm is to expand this set of states that we can learn about safely. Thus, we need to estimate the region of attraction by certifying that state-action pairs achieve the $-L_{\Delta v} \tau$ decrease condition in Theorem 2 by learning about state-action pairs in $\mathcal{S}$. We can then generalize the gained knowledge to unseen states by exploiting the Lipschitz continuity,

$$R^{\mathrm{dec}}(\mathcal{S}) = \mathcal{S}_0 \cup \big\{ \mathbf{z} \in \mathcal{X}_\tau \times \mathcal{U}_\tau \,|\, \exists (\mathbf{x}, \mathbf{u}) \in \mathcal{S} \colon v(f(\mathbf{x}, \mathbf{u})) - v(\mathbf{x}) + \epsilon + L_{\Delta v} \| \mathbf{z} - (\mathbf{x}, \mathbf{u}) \|_1 < -L_{\Delta v} \tau \big\}, \tag{10}$$

where we use that we can learn $v(f(\mathbf{x}, \mathbf{u}))$ up to ϵ accuracy within $\mathcal{S}$. We specifically include $\mathcal{S}_0$ in this set, to allow for initial policies that are safe, but does not meet the strict decrease requirements of Theorem 2. Given that all states in $R^{\mathrm{dec}}(\mathcal{S})$ fulfill the requirements of Theorem 2, we can estimate the corresponding region of attraction by committing to a control policy $\pi \in \Pi_L$ and estimating the largest safe level set of the Lyapunov function. With $\mathcal{D} = R^{\mathrm{dec}}(\mathcal{S})$, the operator

$$R^{\mathrm{lev}}(\mathcal{D}) = \mathcal{V}\big(\operatorname{argmax} c, \quad \text{such that } \exists \pi \in \Pi_L \colon \forall \mathbf{x} \in \mathcal{V}(c) \cap \mathcal{X}_\tau,\, (\mathbf{x}, \pi(\mathbf{x})) \in \mathcal{D} \big) \tag{11}$$

encodes this operation. It optimizes over safe policies $\pi \in \Pi_L$ to determine the largest level set, such that all state-action pairs $(\mathbf{x}, \pi(\mathbf{x}))$ at discrete states $\mathbf{x}$ in the level set $\mathcal{V}(c) \cap \mathcal{X}_\tau$ fulfill the decrease condition of Theorem 2. As a result, $R^{\mathrm{lev}}(R^{\mathrm{dec}}(\mathcal{S}))$ is an estimate of the largest region of attraction given the ϵ-accurate knowledge about state-action pairs in $\mathcal{S}$. Based on this increased region of attraction, there are more states that we can safely learn about. Specifically, we again use the Lipschitz constant and statistical accuracy ϵ to determine all states that map back into the region of attraction,

$$R_\epsilon(\mathcal{S}) = \mathcal{S} \cup \big\{ \mathbf{z}' \in R_\tau^{\mathrm{lev}}(R^{\mathrm{dec}}(\mathcal{S})) \times \mathcal{U}_\tau \,|\, \exists \mathbf{z} \in \mathcal{S} \colon v(f(\mathbf{z})) + \epsilon + L_v L_f \| \mathbf{z} - \mathbf{z}' \|_1 \leq \max_{\mathbf{x} \in R^{\mathrm{lev}}(R^{\mathrm{dec}}(\mathcal{S}))} v(\mathbf{x}) \big\}, \tag{12}$$

where $R_\tau^{\mathrm{lev}}(\mathcal{D}) = R^{\mathrm{lev}}(\mathcal{D}) \cap \mathcal{X}_\tau$. Thus, $R_\epsilon(\mathcal{S}) \supseteq \mathcal{S}$ contains state-action pairs that we can visit to learn about the system. Repeatedly applying this operator leads the largest set of state-action pairs that any safe algorithm with the same knowledge and restricted to policies in Π_L could hope to reach.

Algorithm 2 Theoretical algorithm

```
**Input:** Initial safe policy $\mathcal{S}_0$, dynamics model $\mathcal{GP}(\mu(\mathbf{z}), k(\mathbf{z}, \mathbf{z}'))$
**for all** $n = 1, \dots$ **do**
    $\mathcal{D}_n = \bigcup_{(\mathbf{x},\mathbf{u}) \in \mathcal{S}_{n-1}} \left\{ \mathbf{z}' \in \mathcal{X}_\tau \times \mathcal{U}_\tau \mid u_n(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}) + L_{\Delta v} \| \mathbf{z}' - (\mathbf{x}, \mathbf{u}) \|_1 < -L_{\Delta v} \tau \right\}$,
    $\pi_n, c_n = \operatorname{argmax}_{\pi \in \Pi_L, c \in \mathbb{R}_{>0}} c$,   such that for all $\mathbf{x} \in \mathcal{V}(c) \cap \mathcal{X}_\tau \colon (\mathbf{x}, \pi(\mathbf{x})) \in \mathcal{D}_n$
    $\mathcal{S}_n = \bigcup_{\mathbf{z} \in \mathcal{S}_{n-1}} \left\{ \mathbf{z}' \in \mathcal{V}(c_n) \cap \mathcal{X}_\tau \times \mathcal{U}_\tau \mid u_n(\mathbf{z}) + L_v L_f \| \mathbf{z} - \mathbf{z}' \|_1 \leq c_n \right\}$
        $= \bigcup_{\mathbf{z} \in \mathcal{S}_{n-1}} \left\{ \mathbf{z}' \in R_\tau^{\text{lev}}(\mathcal{D}_n) \times \mathcal{U}_\tau \mid u_n(\mathbf{z}) + L_v L_f \| \mathbf{z} - \mathbf{z}' \|_1 \leq \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_n)} v(\mathbf{x}) \right\}$
    $(\mathbf{x}_n, \mathbf{u}_n) = \operatorname{argmax}_{(\mathbf{x},\mathbf{u}) \in \mathcal{S}_n} u_n(\mathbf{x}, \mathbf{u}) - l_n(\mathbf{x}, \mathbf{u})$
    $\mathcal{S}_n = \{ \mathbf{z} \in \mathcal{V}(c_n) \times \mathcal{U}_\tau \mid u_n(\mathbf{z}) \leq c_n \}$
    Update GP with measurements $f(\mathbf{x}_n, \mathbf{u}_n) + \epsilon_n$
```

Specifically, let $R_\epsilon^0(\mathcal{S}) = \mathcal{S}$ and $R_\epsilon^{i+1}(\mathcal{S}) = R_\epsilon(R_\epsilon^i(\mathcal{S}))$. Then $\overline{R}_\epsilon(\mathcal{S}) = \lim_{i \to \infty} R_\epsilon^i(\mathcal{S})$ is the set of all state-action pars on the discrete grid that any algorithm could hope to classify as safe without leaving this safe set. Moreover, $R^{\text{lev}}(\overline{R}_\epsilon(\mathcal{S}))$ is the largest corresponding region of attraction that any algorithm can classify as safe for the given Lyapunov function.

Proofs In the following we implicitly assume that the assumptions of Lemma 3 hold and that β_n is defined as specified within Lemma 3. Moreover, for ease of notation we assume that $\mathcal{S}_0^x$ is a level set of the Lyapunov function $v(\cdot)$.

Lemma 4. *$\mathcal{V}(c_n) = R^{\text{lev}}(\mathcal{D}_n)$ and $c_n = \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_n)} v(\mathbf{x})$*

Proof. Directly by definition, compare (3) and (11). □

Remark 4. *Lemma 4 allows us to write the proofs entirely in terms of operators, rather than having to deal with explicit policies. In the following and in Algorithm 2 we replace $\mathcal{V}(c_n)$ and c_n according to Lemma 4. This moves the definitions closer to the baseline and makes for an easier comparison.*

We roughly follow the proof strategy in [18], but deal with the additional complexity of having safe sets that are defined in a more difficult way (indirectly through the policy). This is non-trivial and the safe sets are carefully designed in order to ensure that the algorithm works for general nonlinear systems.

We start by listing some fundamental properties of the sets that we defined below.

Lemma 5. *It holds for all $n \geq 1$ that*

(i) $\forall \mathbf{z} \in \mathcal{X}_\tau \times \mathcal{U}_\tau,\ u_{n+1}(\mathbf{z}) \leq u_n(\mathbf{z})$

(ii) $\forall \mathbf{z} \in \mathcal{X}_\tau \times \mathcal{U}_\tau,\ l_{n+1}(\mathbf{z}) \geq l_n(\mathbf{z})$

(iii) $\mathcal{S} \subseteq \mathcal{R} \implies R^{\text{lev}}(\mathcal{S}) \subseteq R^{\text{lev}}(\mathcal{R})$

(iv) $\mathcal{S} \subseteq \mathcal{R} \implies R^{\text{dec}}(\mathcal{S}) \subseteq R^{\text{dec}}(\mathcal{R})$

(v) $\mathcal{S} \subseteq \mathcal{R} \implies R_\epsilon(\mathcal{S}) \subseteq R_\epsilon(\mathcal{R})$

(vi) $\mathcal{S} \subseteq \mathcal{R} \implies \overline{R}_\epsilon(\mathcal{S}) \subseteq \overline{R}_\epsilon(\mathcal{R})$

(vii) $\mathcal{S}_n \supseteq \mathcal{S}_{n-1} \implies \mathcal{D}_{n+1} \supseteq \mathcal{D}_n$

(viii) $\mathcal{D}_1 \supseteq \mathcal{S}_0$

(ix) $\mathcal{S}_n \supseteq \mathcal{S}_{n-1}$

(x) $\mathcal{D}_n \supseteq \mathcal{D}_{n-1}$

Proof. (i) and (ii) follow directly form the definition of $\mathcal{C}_n$.

(iii) Let $\pi \in \Pi_L$ be a policy such that for some $c > 0$ it holds for all $\mathbf{x} \in \mathcal{V}(c) \cap \mathcal{X}_\tau$ that $(\mathbf{x}, \pi(\mathbf{x})) \in \mathcal{S}$. Then we have that $(\mathbf{x}, \pi(\mathbf{x})) \in \mathcal{R}$, since $\mathcal{S} \subseteq \mathcal{R}$. Thus it follows that with

$$c_s = \operatorname{argmax} c \quad \text{s.t. } \exists \pi \in \Pi_L \colon \forall \mathbf{x} \in \mathcal{V}(c) \cap \mathcal{X}_\tau,\ (\mathbf{x}, \pi(\mathbf{x})) \in \mathcal{S} \tag{13}$$

and

$$c_r = \operatorname{argmax} c \quad \text{s.t. } \exists \pi \in \Pi_L : \forall \mathbf{x} \in \mathcal{V}(c) \cap \mathcal{X}_\tau,\ (\mathbf{x}, \pi(\mathbf{x})) \in \mathcal{R} \tag{14}$$

we have that $c_r \geq c_s$. This implies $\mathcal{V}(c_r) \supseteq \mathcal{V}(c_s)$. The result follows.

(iv) Let $\mathbf{z} \in R^{\text{dec}}(\mathcal{S})$. Then there exists $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}$ such that $v(f(\mathbf{x}, \mathbf{u})) - v(\mathbf{x}) + \epsilon + L_{\Delta v} \|\mathbf{z} - (\mathbf{x}, \mathbf{u})\|_1 < -L_{\Delta v} \tau$. Since $\mathcal{S} \subseteq \mathcal{R}$ we have that $(\mathbf{x}, \mathbf{u}) \in \mathcal{R}$ as well and thus $\mathbf{z} \in R^{\text{dec}}(\mathcal{R})$.

(v) $\mathcal{S} \subseteq \mathcal{R} \implies R^{\text{lev}}(R^{\text{dec}}(\mathcal{S})) \subseteq R^{\text{lev}}(R^{\text{dec}}(\mathcal{R}))$ due to (iii) and (iv). Since $\mathbf{z}' \in R_\epsilon(\mathcal{S})$, there must exist an $\mathbf{z} \in \mathcal{S}$ such that $v(f(\mathbf{z})) + \epsilon + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 \leq \max_{\mathbf{x} \in R^{\text{lev}}(R^{\text{dec}}(\mathcal{S}))} v(\mathbf{x})$. Since $\mathcal{S} \subseteq \mathcal{R}$ it follows that $\mathbf{z} \in \mathcal{R}$. Moreover,

$$\max_{\mathbf{x} \in R^{\text{lev}}(R^{\text{dec}}(\mathcal{S}))} v(\mathbf{x}) \leq \max_{\mathbf{x} \in R^{\text{lev}}(R^{\text{dec}}(\mathcal{R}))} v(\mathbf{x}) \tag{15}$$

follows from $R^{\text{lev}}(R^{\text{dec}}(\mathcal{S})) \subseteq R^{\text{lev}}(R^{\text{dec}}(\mathcal{R}))$, so that we conclude that $\mathbf{z}' \in R_\epsilon(\mathcal{R})$.

(vi) This follows directly by repeatedly applying the result of (v).

(vii) Let $\mathbf{z}' \in \mathcal{D}_n$. Then $\exists (\mathbf{x}, \mathbf{u}) \in \mathcal{S}_{n-1} \colon u_n(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}) + L_{\Delta v} \|\mathbf{z}' - (\mathbf{x}, \mathbf{u})\|_1 < -L_{\Delta v} \tau$. Since $\mathcal{S}_n \supseteq \mathcal{S}_{n-1}$ it follows that $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}_n$ as well. Moreover, we have

$$\begin{aligned} &u_{n+1}(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}) + L_{\Delta v} \|\mathbf{z}' - (\mathbf{x}, \mathbf{u})\|_1 \\ &\leq u_n(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}) + L_{\Delta v} \|\mathbf{z}' - (\mathbf{x}, \mathbf{u})\|_1 < -L_{\Delta v} \tau \end{aligned}$$

since u_{n+1} is non-increasing, see (i). Thus $\mathbf{z}' \in \mathcal{D}_{n+1}$.

(viii) By definition of $\mathcal{C}_0$ we have for all $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}_0$ that $u_0(\mathbf{x}, \mathbf{u}) < v(\mathbf{x}) - L_{\Delta v} \tau$. Now we have that

$$\begin{aligned} &u_1(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}) + L_{\Delta v} \|(\mathbf{x}, \mathbf{u}) - (\mathbf{x}, \mathbf{u})\|_1, \\ &= u_1(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}), \\ &\leq u_0(\mathbf{x}, \mathbf{u}) - v(\mathbf{x}), \quad \text{by Lemma 5 (i)} \\ &< -L_{\Delta v} \tau, \end{aligned}$$

which implies that $(\mathbf{x}, \mathbf{u}) \in \mathcal{D}_1$.

(ix) Proof by induction. We consider the base case, $\mathbf{z} \in \mathcal{S}_0$, which implies that $\mathbf{z} \in \mathcal{D}_1$ by (viii). Moreover, since $\mathcal{S}_0^x$ is a level set of the Lyapunov function v by assumption, we have that $R^{\text{lev}}(\mathcal{S}_0) = \mathcal{S}_0^x$. The previous statements together with (iii) imply that $\mathbf{z} \in R_\tau^{\text{lev}}(\mathcal{D}_1) \times \mathcal{U}_\tau$, since $\mathcal{D}_1 \supseteq \mathcal{S}_0$ by (viii). Now, we have that

$$u_1(\mathbf{z}) + L_v L_f \|\mathbf{z} - \mathbf{z}\|_1 = u_1(\mathbf{z}) \overset{(i)}{\leq} u_0(\mathbf{z}).$$

Moreover, by definition of $\mathcal{C}_0$, we have for all $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}_0$ that

$$u_0(\mathbf{x}, \mathbf{u}) < v(\mathbf{x}) - L_{\Delta v} \tau < v(\mathbf{x}).$$

As a consequence,

$$u_0(\mathbf{x}, \mathbf{u}) \leq \max_{(\mathbf{x}, \mathbf{u}) \in \mathcal{S}_0} v(\mathbf{x}), \tag{16}$$

$$= \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{S}_0)} v(\mathbf{x}), \tag{17}$$

$$\leq \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_1)} v(\mathbf{x}), \tag{18}$$

where the last inequality follows from (iii) and (viii). Thus we have $\mathbf{z} \in \mathcal{S}_1$.

For the induction step, assume that for $n \geq 2$ we have $\mathbf{z}' \in \mathcal{S}_n$ with $\mathcal{S}_n \supseteq \mathcal{S}_{n-1}$. Now since $\mathbf{z}' \in \mathcal{S}_n$ we must have that $\mathbf{z}' \in R_\tau^{\text{lev}}(\mathcal{D}_n) \times \mathcal{U}_\tau$. This implies that $\mathbf{z}' \in R_\tau^{\text{lev}}(\mathcal{D}_{n+1}) \times \mathcal{U}_\tau$, due to Lemma 5 (iii) and (vii) together with the induction assumption of $\mathcal{S}_n \supseteq \mathcal{S}_{n-1}$. Moreover, there must exist a $\mathbf{z} \in \mathcal{S}_{n-1} \subseteq \mathcal{S}_n$ such that

$$u_{n+1}(\mathbf{z}) + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1, \leq u_n(\mathbf{z}) + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1, \tag{19}$$

$$\leq \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_n)} v(\mathbf{x}), \tag{20}$$

$$\leq \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_{n+1})} v(\mathbf{x}), \tag{21}$$

which in turn implies $\mathbf{z} \in \mathcal{S}_{n+1}$. The last inequality follows from Lemma 5 (iii) and (vii) together with the induction assumption that $\mathcal{S}_n \supseteq \mathcal{S}_{n-1}$.

(x) This is a direct consequence of (vii), (viii), and (ix).

□

Given these set properties, we first consider what happens if the safe set $\mathcal{S}_n$ does not expand after collecting data points. We use these results later to conclude that the safe set must either expand or that the maximum level set is reached. We denote by

$$\mathbf{z}_n = (\mathbf{x}_n, \mathbf{u}_n)$$

the data point the is sampled according to (6).

Lemma 6. *For any $n_1 \geq n_0 \geq 1$, if $\mathcal{S}_{n_1} = \mathcal{S}_{n_0}$, then for any n such that $n_0 \leq n < n_1$, it holds that*

$$2\beta_n \sigma_n(\mathbf{z}_n) \leq \sqrt{\frac{C_1 q \beta_n^2 \gamma_n}{n - n_0}}, \tag{22}$$

where $C_1 = 8/\log(1 + \sigma^{-2})$.

Proof. We modify the results for $q = 1$ by [35] to this lemma, but use the different definition for β_n from [34]. Even though the goal of [35, Lemma 5.4] is different from ours, we can still apply their reasoning to bound the amplitude of the confidence interval of the dynamics. In particular, in [35, Lemma 5.4], we have $r_n = 2\beta_n \sigma_{n-1}(\mathbf{z}_n)$ with $\mathbf{z}_n = (\mathbf{x}_n, \mathbf{u}_n)$ according to Lemma 3. Then

$$\begin{aligned}
r_n^2 &= 4\beta_n^2 \sigma_{n-1}^2(\mathbf{z}_n), && (23)\\
&= 4\beta_n^2 \left(\sum_{i=1}^{q} \sigma_{n-1}(\mathbf{z}_n, i) \right)^2, && (24)\\
&\leq 4\beta_n^2 q \sum_{i=1}^{q} \sigma_{n-1}^2(\mathbf{z}_n, i) \qquad \text{(Jensen's ineq.)}, && (25)\\
&\leq 4\beta_n^2 q \sigma^2 C_2 \sum_{i=1}^{q} \log(1 + \sigma^{-2}\sigma_{n-1}^2(\mathbf{z}_n, i)), && (26)
\end{aligned}$$

where $C_2 = \sigma^{-2}/\log(1 + \sigma^{-2})$. The result then follows analogously to [35, Lemma 5.4] by noting that

$$\sum_{j=1}^{n} r_j^2 \leq C_1 \beta_n^2 q \gamma_n \qquad \forall n \geq 1 \tag{27}$$

according to the definition of γ_n in this paper and using the Cauchy-Schwarz inequality. □

The previous result allows us to bound the width of the confidence intervals:

Corollary 2. *For any $n_1 \geq n_0 \geq 1$, if $\mathcal{S}_{n_1} = \mathcal{S}_{n_0}$, then for any n such that $n_0 \leq n < n_1$, it holds that*

$$u_n(\mathbf{z}_n) - l_n(\mathbf{z}_n) \leq L_v \sqrt{\frac{C_1 q \beta_n^2 \gamma_n}{n - n_0}}, \tag{28}$$

where $C_1 = 8/\log(1 + \sigma^{-2})$.

Proof. Direct consequence of Lemma 6 together with the definition of $\mathcal{C}$ and $\mathcal{Q}$. □

Corollary 3. *For any $n \geq 1$ with C_1 as defined in Lemma 6, let N_n be the smallest integer satisfying $\frac{N_n}{\beta_{n+N_n}^2 \gamma n + N_n} \geq \frac{C_1 L_v^2 q}{\epsilon^2}$ and $\mathcal{S}_{n+N_n} = \mathcal{S}_{N_n}$, then, for any $\mathbf{z} \in \mathcal{S}_{n+N_n}$ it holds that*

$$u_n(\mathbf{z}) - l_n(\mathbf{z}) \leq \epsilon. \tag{29}$$

Proof. The result trivially follows from substituting N_n in the bound in Corollary 2. □

Lemma 7. *For any* $n \geq 1$*, if* $\overline{R}_\epsilon(\mathcal{S}_0) \setminus \mathcal{S}_n \neq \emptyset$*, then* $R_\epsilon(\mathcal{S}_n) \setminus \mathcal{S}_n \neq \emptyset$*.*

Proof. As in [18, Lemma 6]. Assume, to the contrary, that $R_\epsilon(\mathcal{S}_n) \setminus \mathcal{S}_n = \emptyset$. By definition $R_\epsilon(\mathcal{S}_n) \supseteq \mathcal{S}_n$, therefore $R_\epsilon(\mathcal{S}_n) = \mathcal{S}_n$. Iteratively applying R_ϵ to both sides, we get in the limit $\overline{R}_\epsilon(\mathcal{S}_n) = \mathcal{S}_n$. But then, by Lemma 5,(vi) and (ix), we get

$$\overline{R}_\epsilon(\mathcal{S}_0) \subseteq \overline{R}_\epsilon(\mathcal{S}_n) = \mathcal{S}_n, \tag{30}$$

which contradicts the assumption that $\overline{R}_\epsilon(\mathcal{S}_0) \setminus \mathcal{S}_n \neq \emptyset$. □

Lemma 8. *For any* $n \geq 1$*, if* $\overline{R}_\epsilon(\mathcal{S}_0) \setminus \mathcal{S}_n \neq \emptyset$*, then the following holds with probability at least* $1 - \delta$*:*

$$\mathcal{S}_{n+N_n} \supset \mathcal{S}_n. \tag{31}$$

Proof. By Lemma 7, we have that $R_\epsilon(\mathcal{S}_n) \setminus \mathcal{S}_n \neq \emptyset$. By definition, this means that there exist $\mathbf{z} \in R_\epsilon(\mathcal{S}_n) \setminus \mathcal{S}_n$ and $\mathbf{z}' \in \mathcal{S}_n$ such that

$$v(f(\mathbf{z}')) + \epsilon + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 \leq \max_{\mathbf{x} \in R^{\text{lev}}(R^{\text{dec}}(\mathcal{S}_n))} v(\mathbf{x}) \tag{32}$$

Now we assume, to the contrary, that $\mathcal{S}_{n+N_n} = \mathcal{S}_n$ (the safe set cannot decrease due to Lemma 5 (ix)). This implies that $\mathbf{z} \in \mathcal{X}_\tau \times \mathcal{U}_\tau \setminus \mathcal{S}_{n+N_n}$ and $\mathbf{z}' \in \mathcal{S}_{n+N_n} = \mathcal{S}_{n+N_n-1}$. Due to Corollary 2, it follows that

$$\begin{aligned}
&u_{n+N_n}(\mathbf{z}') + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 && && (33)\\
&\leq v(f(\mathbf{z}')) + \epsilon + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 && && (34)\\
&\leq \max_{\mathbf{x} \in R^{\text{lev}}(R^{\text{dec}}(\mathcal{S}_n))} v(\mathbf{x}) && \text{by (32)} && (35)\\
&= \max_{\mathbf{x} \in R^{\text{lev}}(R^{\text{dec}}(\mathcal{S}_{n+N_n}))} v(\mathbf{x}) && \text{by (iii), (iv) and (ix)} && (36)
\end{aligned}$$

Thus, to conclude that $\mathbf{z} \in \mathcal{S}_{n+N_n}$ according to (5), we need to show that $R^{\text{lev}}(\mathcal{D}_{n+N_n}) \supseteq R^{\text{lev}}(R^{\text{dec}}(\mathcal{S}_n))$. To this end, we use Lemma 5 (iii) and show that $\mathcal{D}_{n+N_n} \supseteq R^{\text{dec}}(\mathcal{S}_{n+N_n})$. Consider $(\mathbf{x}, \mathbf{u}) \in R^{\text{dec}}(\mathcal{S}_{n+N_n})$, we know that there exists a $(\mathbf{x}', \mathbf{u}') \in \mathcal{S}_{n+N_n} = \mathcal{S}_{n+N_n-1}$ such that

$$\begin{aligned}
-L_{\Delta v}\tau &> v(f(\mathbf{x}', \mathbf{u}')) - v(\mathbf{x}') + \epsilon + L_{\Delta v}\|(\mathbf{x}, \mathbf{u}) - (\mathbf{x}', \mathbf{u}')\|_1, && (37)\\
&\geq u_{n+N_n}(\mathbf{x}', \mathbf{u}') - v(\mathbf{x}') + L_{\Delta v}\|(\mathbf{x}, \mathbf{u}) - (\mathbf{x}', \mathbf{u}')\|_1, && (38)
\end{aligned}$$

where the second inequality follows from Corollary 2. This implies that $(\mathbf{x}, \mathbf{u}) \in \mathcal{D}_n$ and thus $\mathcal{D}_{n+N_n} \supseteq R^{\text{dec}}(\mathcal{S}_{n+N_n})$. This, in turn, implies that $\mathbf{z} \in \mathcal{S}_{n+N_n}$, which is a contradiction. □

Lemma 9. *For any* $n \geq 0$*, the following holds with probability at least* $1 - \delta$*:*

$$\mathcal{S}_n \subseteq \overline{R}_0(\mathcal{S}_0). \tag{39}$$

Proof. Proof by induction. For the base case, $n = 0$, we have $\mathcal{S}_0 \subseteq \overline{R}_0(\mathcal{S}_0)$ by definition.

For the induction step, assume that for some $n \geq 1$, $\mathcal{S}_{n-1} \subseteq \overline{R}_0(\mathcal{S}_0)$. Let $\mathbf{z} \in \mathcal{S}_n$. Then, by definition, $\exists \mathbf{z}' \in \mathcal{S}_{n-1}$ such that

$$u_n(\mathbf{z}') + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 \leq \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_n)} v(\mathbf{x}), \tag{40}$$

which, by Corollary 1, implies that

$$v(f(\mathbf{z}')) + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 \leq \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_n)} v(\mathbf{x}) \tag{41}$$

Now since $\mathbf{z}' \in \overline{R}_0(\mathcal{S}_0)$ by the induction hypothesis, in order to conclude that $\mathbf{z} \in \overline{R}_0(\mathcal{S}_0)$ we need to show that $R^{\text{lev}}(\mathcal{D}_n) \subseteq R^{\text{lev}}(R^{\text{dec}}(\overline{R}(\mathcal{S}_0)))$.

Let $(\mathbf{x}, \mathbf{u}) \in \mathcal{D}_n$, then there exist $(\mathbf{x}', \mathbf{z}') \in \mathcal{S}_{n-1}$ such that

$$u_{n-1}(\mathbf{x}', \mathbf{u}') - v(\mathbf{x}') + L_{\Delta v} \|(\mathbf{x}, \mathbf{u}) - (\mathbf{x}', \mathbf{u}')\|_1 < -L_{\Delta v} \tau, \tag{42}$$

which, by Corollary 1, implies that

$$v(f(\mathbf{x}', \mathbf{u}')) - v(\mathbf{x}') + L_{\Delta v} \|(\mathbf{x}, \mathbf{u}) - (\mathbf{x}', \mathbf{u}')\|_1 < -L_{\Delta v} \tau, \tag{43}$$

which means that $(\mathbf{x}, \mathbf{u}) \in R^{\text{dec}}(\overline{R}_0(\mathcal{S}_0))$ since $\mathcal{S}_{n-1} \subseteq \overline{R}_0(\mathcal{S}_0)$ and therefore $(\mathbf{x}', \mathbf{u}') \in \overline{R}_0(\mathcal{S}_0)$ holds by the induction hypothesis. We use (iii) to conclude that $R^{\text{lev}}(\mathcal{D}_n) \subseteq R^{\text{lev}}(R^{\text{dec}}(\overline{R}(\mathcal{S}_0)))$, which concludes the proof. □

Lemma 10. *Let n^* be the smallest integer, such that $n^* \geq |\overline{R}_0(\mathcal{S}_0)| N_{n^*}$. Then, there exists $n_0 \leq n^*$ such that $\mathcal{S}_{n_0 + N_{n_0}} = \mathcal{S}_{n_0}$ holds with probability $1 - \delta$.*

Proof. By contradiction. Assume, to the contrary, that for all $n \leq n^*$, $\mathcal{S}_n \subset \mathcal{S}_{n+N_n}$. From Lemma 5 (ix) we know that $\mathcal{S}_n \subseteq \mathcal{S}_{n+N_n}$. Since N_n is increasing in n, we have that $N_n \leq N_{n^*}$. Thus, we must have

$$\mathcal{S}_0 \subset \mathcal{S}_{N_{n^*}} \subset \mathcal{S}_{2N_{n^*}} \cdots, \tag{44}$$

so that for any $0 \leq j \leq |\overline{R}_0(\mathcal{S}_0)|$, it holds that $|S_{jT_{n^*}}| > j$. In particular, for $j = |\overline{R}_0(\mathcal{S}_0)|$, we get

$$|\mathcal{S}_{jN_{n^*}}| > |\overline{R}_0(\mathcal{S}_0)|, \tag{45}$$

which contradicts $\mathcal{S}_{jN_{n^*}} \subseteq \overline{R}_0(\mathcal{S}_0)$ from Lemma 9. □

Corollary 4. *Let n^* be the smallest integer such that*

$$\frac{n^*}{\beta_{n^*} \gamma_{n^*}} \geq \frac{C_1 L_v^2 q |\overline{R}_0(\mathcal{S}_0)|}{\epsilon^2}, \tag{46}$$

then there exists a $n_0 \leq n^*$ such that $\mathcal{S}_{n_0 + N_{n_0}} = \mathcal{S}_{n_0}$.

Proof. A direct consequence of Lemma 10 and Corollary 3. □

A.4 Safety and policy adaptation

In the following, we denote the true region of attraction of (1) under a policy π by $\mathcal{R}_\pi$.

Lemma 11. $R^{\text{lev}}(\mathcal{D}_n) \subseteq \mathcal{R}_{\pi_n}$ *for all* $n \geq 0$.

Proof. By definition, we have for all $(\mathbf{x}, \mathbf{u}) \in \mathcal{D}_n$ that the exists $(\mathbf{x}', \mathbf{u}') \in \mathcal{S}_{n-1}$ such that

$$\begin{aligned} -L_{\Delta v} \tau &\geq u_n(\mathbf{x}', \mathbf{u}') - v(\mathbf{x}') + L_{\Delta v} \|(\mathbf{x}, \mathbf{u}) - (\mathbf{x}', \mathbf{u}')\|_1, \\ &\geq v(f(\mathbf{x}', \mathbf{u}')) - v(\mathbf{x}') + L_{\Delta v} \|(\mathbf{x}, \mathbf{u}) - (\mathbf{x}', \mathbf{u}')\|_1, \\ &\geq v(f(\mathbf{x}, \mathbf{u})) - v(\mathbf{x}), \end{aligned}$$

where the first inequality follows from Corollary 1 and the second one by Lipschitz continuity, see Lemma 1.

By definition of R^{lev} in (11), it follows that for all $\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_n) \cap \mathcal{X}_\tau$ we have that $(\mathbf{x}, \pi_n(\mathbf{x})) \in \mathcal{D}_n$. Moreover, $R^{\text{lev}}(\mathcal{D}_n)$ is a level set of the Lyapunov function by definition. Thus the result follows from Theorem 2. □

Lemma 12. $f(\mathbf{x}, \mathbf{u}) \in \mathcal{R}_{\pi_n} \; \forall (\mathbf{x}, \mathbf{u}) \in \mathcal{S}_n$.

Proof. This holds for $\mathcal{S}_0$ by definition. For $n \geq 1$, by defition, we have for all $\mathbf{z} \in \mathcal{S}_n$ there exists an $\mathbf{z}' \in \mathcal{S}_{n-1}$ such that

$$\begin{aligned} \max_{\mathbf{x} \in R^{\text{lev}}(\mathcal{D}_n)} v(\mathbf{x}) &\geq u_n(\mathbf{z}') + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 \\ &\geq v(f(\mathbf{z}')) + L_v L_f \|\mathbf{z} - \mathbf{z}'\|_1 \\ &\geq v(f(\mathbf{z})) \end{aligned}$$

where the first inequality follows from Corollary 1 and the second one by Lipschitz continuity, see Lemma 1. Since $R^{\text{lev}}(\mathcal{D}_n) \subseteq \mathcal{R}_{\pi_n}$ by Lemma 11, we have that $f(\mathbf{z}) \in \mathcal{R}_{\pi_n}$. □

Theorem 4. *Assume σ-sub-Gaussian measurement noise and that the model error $g(\cdot)$ in* (1) *has RKHS norm smaller than B_g. Under the assumptions of Theorem 2, with $\beta_n = B_g + 4\sigma\sqrt{\gamma_n + 1 + \ln(1/\delta)}$, and with measurements collected according to* (6)*, let n^* be the smallest positive integer so that $\frac{n^*}{\beta^2_{n^*}\gamma_{n^*}} \geq \frac{Cq(|\overline{R}(\mathcal{S}_0)|+1)}{L_v^2\epsilon^2}$ where $C = 8/\log(1+\sigma^{-2})$. Let $\mathcal{R}_\pi$ be the true region of attraction of* (1) *under a policy π. For any $\epsilon > 0$, and $\delta \in (0,1)$, the following holds jointly with probability at least $(1-\delta)$ for all $n > 0$:*

(i) $\mathcal{V}(c_n) \subseteq \mathcal{R}_{\pi_n}$ *(ii)* $f(\mathbf{x}, \mathbf{u}) \in \mathcal{R}_{\pi_n}\ \forall (\mathbf{x}, \mathbf{u}) \in \mathcal{S}_n$. *(iii)* $\overline{R}_\epsilon(\mathcal{S}_0) \subseteq \mathcal{S}_n \subseteq \overline{R}_0(\mathcal{S}_0)$.

Proof. See Lemmas 11 and 12 for (i) and (ii), respectively. Part (iii) is a direct consequence of Corollary 4 and Lemma 9. □