Learning Certifiably Robust Controllers Using Fragile Perception

Dawei Sun
University of Illinois
Urbana, IL 61801, USA
daweis2@illinois.edu

Negin Musavi
University of Illinois
Urbana, IL 61801, USA
nmusavi2@illinois.edu

Geir Dullerud
University of Illinois
Urbana, IL 61801, USA
dullerud@illinois.edu

Sanjay Shakkottai
University of Texas at Austin
Austin, TX 78712, USA
sanjay.shakkottai@utexas.edu

Sayan Mitra
University of Illinois
Urbana, IL 61801, USA
mitras@illinois.edu

Abstract

Advances in computer vision and machine learning enable robots to perceive their surroundings in powerful new ways, but these perception modules have well-known fragilities. We consider the problem of synthesizing a safe controller that is robust despite perception errors. The proposed method constructs a state estimator based on Gaussian processes with input-dependent noises. This estimator computes a high-confidence set for the actual state given a perceived state. Then, a robust neural network controller is synthesized that can provably handle the state uncertainty. Furthermore, an adaptive sampling algorithm is proposed to jointly improve the estimator and controller. Simulation experiments, including a realistic vision-based lane keeping example in CARLA, illustrate the promise of the proposed approach in synthesizing robust controllers with deep-learning-based perception.

1 Introduction

Advances in computer vision and machine learning enable robots to perceive their surroundings in powerful new ways, but these perception modules have well-known fragilities. In this paper, we address the problem of designing robust controllers that tolerate and compensate for the fragilities of the perception modules they use. Simply put, *how to design reliable controllers that use unreliable perception?* We study the fundamental control task for a robot to maintain a given *invariant*. For example, a drone has to stay within a geo-fenced area, or a car has to stay within the lanes.

In this paper, the perception module is modeled as a function that takes in the actual state and generates a noisy copy of it, i.e., the perceived state. In order to design a feedback controller that only has access to the perceived state to maintain safety, we utilize Gaussian processes (GP) to construct a state estimator and machine learning to synthesize a certified controller. Furthermore, an adaptive sampling algorithm is proposed to jointly improve the estimator and controller. We evaluate the proposed approach on three benchmark systems. Experimental results clearly verify that with the proposed approach, one can synthesize controllers that are robust to perception errors, and the proposed adaptive sampling algorithm indeed improves the sample efficiency.

Related work Data-driven approaches have been developed for perception-based control [1; 2]. However, such purely data-driven approaches do not provide any safety guarantees. In a series of recent works by Dean et al., the authors studied the robustness guarantees of perception-based control algorithms. In [3], the authors proposed a perception-based controller synthesis approach for linear

NeurIPS 2022 Workshop on Robot Learning: Trustworthy Robotics, Virtual, Virtual

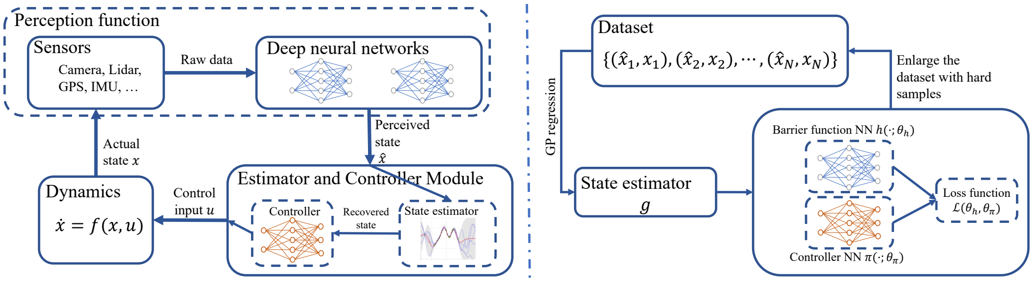
Perception function
Sensors
Camera, Lidar, GPS, IMU, …
Raw data
Deep neural networks
Perceived state $\hat{x}$
Actual state x
Dynamics
$\dot{x} = f(x,u)$
Control input u
Estimator and Controller Module
Controller
Recovered state
State estimator
Dataset
$\{(\hat{x}_1, x_1), (\hat{x}_2, x_2), \cdots, (\hat{x}_N, x_N)\}$
Enlarge the dataset with hard samples
GP regression
State estimator g
Barrier function NN $h(\cdot;\theta_h)$
Loss function $\mathcal{L}(\theta_h, \theta_\pi)$
Controller NN $\pi(\cdot;\theta_\pi)$

Figure 1: Overview of the system and the learning algorithm.

systems and provided a theoretical analysis. In [4], the authors proposed robust barrier functions for synthesizing safety-critical controllers under uncertainty of the state. In our approach, such a robust barrier function is used as a component of the estimation and control pipeline.

2 Preliminaries and problem setup

Dynamical systems. We consider dynamical systems of the form $\dot{x} = f(x, u)$, where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state and $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input.

Imperfect perception functions. We study systems equipped with sensors and perception modules, which together constitute a *perception function* $s : \mathcal{X} \mapsto \mathcal{X}$. That is, it takes in the actual state $x \in \mathcal{X}$ and produces a noisy observation $\hat{x} := s(x)$ which is called the *perceived state*. The perception function s is a complex composite of the environment, the sensor, and the perception module, for instance, a deep convolutional neural network, and we treat it as a black box function.

Invariant sets. Consider an autonomous system $\dot{x} = f(x)$, where $x \in \mathcal{X}$. A set $\mathcal{C} \subseteq \mathcal{X}$ is called an invariant set of it, if starting from any initial state in $\mathcal{C}$, a trajectory always stays in $\mathcal{C}$.

The synthesis problem. As shown in Figure 1, the goal is to synthesize a module that computes control input u for the dynamical system such that after plugging this module into the dynamical system, a user-defined set $\mathcal{S} \subset \mathcal{X}$ is invariant to the closed-loop system. However, different from an ordinary feedback controller, this module does not have access to the actual state x of the system. Instead, it only has access to a noisy version of the actual state, which is the perceived state $\hat{x}$.

3 Design Methodology

Our approach decomposes the desired controller function c into two components, a GP-based *state estimator* $g : \mathcal{X} \mapsto 2^{\mathcal{X}}$ aiming to recover a high-confidence set for the actual state from the perceived state, and a robust controller $\pi : \mathcal{X} \mapsto \mathcal{U}$ (See Fig. 1). As will be shown later, the output of g is an ellipsoid. We call the sequential combination of the two components as the *Estimator and Controller Module (ECM)*: $c(\cdot) := \pi(\mathrm{center}(g(\cdot)))$, where center picks the center of an ellipsoid.

3.1 Constructing the state estimator

The state estimator is designed as $g : \mathcal{X} \mapsto 2^{\mathcal{X}}$ such that for an arbitrary state x, $x \in g(s(x))$ with a high probability. In this paper, we construct such an estimator using Gaussian processes. Here, we view the problem of estimating x from $\hat{x}$ as a regression problem.

Construction of the data set. The state estimator will be constructed from samples of the perception function s. To this end, a data set $\mathcal{D} = \{(\hat{x}_j, x_j)\}_{j=1}^N$ that captures the relationship between the perceived state and the actual state is constructed. Each sample is obtained as follows: first, a state $x_j \in \mathcal{X}$ is sampled, then $\hat{x}_j$ is computed as $\hat{x}_j = s(x_j)$.

Setup of the probabilistic model. In order to apply GP regression, we first set up a probabilistic model to represent the observations in the above data set. As stated earlier, since s might not be invertible, there might be multiple x's that correspond to the same $\hat{x}$ in the above data set. To characterize this property of the data set, we assume there is an input-dependent observation noise.

Specifically, we adopt the following probabilistic model (Along the lines of [5]).

$$x_j = s^+(\hat{x}_j) + w_j,\; j = 1, \cdots, N,$$

where $s^+ : \mathcal{X} \mapsto \mathcal{X}$ approximately inverts s with an input-dependent zero-mean noise $w_j \sim \mathcal{N}\left(0, \mathrm{diag}(\exp(z(\hat{x}_j)))^2\right)$. Here, $z : \mathcal{X} \mapsto \mathbb{R}^n$ is the noise-level function, which characterizes the non-invertibility of the perception function s at $\hat{x}$.

Posterior distribution and construction of the high-confidence set. With the data set and the prior distributions, the posterior distribution of the actual state x corresponding to a query perceived state $\hat{x}$ is also a multivariate Gaussian distribution as in [5]. Then, it is standard to construct an ellipsoid as the high-confidence set. Specifically, given a confidence level δ, we construct the high-confidence set $g(\hat{x})$ as an ellipsoid centered at the mean of the multivariate Gaussian distribution and whose semiaxes are proportional to the standard deviations such that $\Pr\left(x \in g(\hat{x})\right) = \delta$. For more details, please refer to the extended version [6] of this paper.

3.2 Learning the controller

In this section, we elaborate on the process of learning a certified controller π given such a state estimator g. We fix the *target* invariant set $\mathcal{S} \subset \mathcal{X}$ for the control system. In order to prove or certify that $\mathcal{S}$ is indeed invariant with respect to the closed-loop system, we will learn a continuously differentiable function $h : \mathcal{X} \mapsto \mathbb{R}$, the *certificate*, such that the 0-superlevel set of h, $\mathcal{C}_h := \{x \in \mathcal{X} : h(x) > 0\}$ is equal to $\mathcal{S}$. Functions like h are often called barrier functions or barrier certificates. The learning algorithm is designed based on the following theorem inspired by [4].

Theorem 1. *Assume that a state estimator g satisfies that for all x, the set $g(s(x))$ contains x. If $\pi : \mathcal{X} \mapsto \mathcal{U}$ be a differentiable controller such that there exists an extended class $\mathcal{K}_\infty$ function α such that $\forall \hat{x} \in \mathcal{X}$,*

$$\inf_{x \in g(\hat{x})} \left(\frac{\partial h}{\partial x}(x) \cdot f(x, c(\hat{x})) + \alpha(h(x)) \right) \geq 0, \tag{1}$$

where $c(\cdot) := \pi(\mathrm{center}(g(\cdot)))$, then, $\mathcal{C}_h$ is invariant to the closed-loop system, i.e., $\dot{x} = f(x, c \circ s(x))$.

Learning-based synthesis. As shown in Figure 1, we model the controller and the barrier function with two neural networks $\pi(\cdot; \theta_\pi)$ and $h(\cdot; \theta_h)$, where θ_π and θ_h are the parameters. The learning algorithm aims at finding the correct parameters such that π and h satisfy the condition in Theorem 1. In order to train the neural networks on sampled data, we transform the above loss functions into their empirical version. That is, replacing the expectations with empirical averages. To this end, we construct a data set $\mathcal{D}_c$ as follows. We sample M_1 perceived state $\hat{x}$ from $\mathtt{Unif}\;(\mathcal{X})$ and denote them by $\{\hat{x}_i\}_{i=1}^{M_1}$. Then, for each $\hat{x}_i$, we sample M_2 points x from $\mathtt{Unif}\;(g(\hat{x}))$ and denote them by $\{x_i^j\}_{j=1}^{M_2}$. These samples constitute the data set $\mathcal{D}_c := \cup_{i=1}^{M_1}\{(\hat{x}_i, x_i^j)\}_{j=1}^{M_2}$. Then, we train the neural networks with the following loss function $\mathcal{L}$.

$$\begin{aligned}\mathcal{L}(\theta_h, \theta_\pi) = \frac{1}{M_1 M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \Bigg[& \lambda_1 \mathtt{ReLU}\left(- \left(\frac{\partial h}{\partial x}(x_i^j) \cdot f(x_i^j, \pi(\mathrm{center}(g(\hat{x}_i)))) + \alpha(h(x_i^j)) \right) \right) \\ & + \lambda_2 \left(\left(1 - \mathbb{I}_{\mathcal{S}}(x_i^j)\right) h(x_i^j) - \mathbb{I}_{\mathcal{S}}(x_i^j) h(x_i^j) \right) \Bigg],\end{aligned} \tag{2}$$

where the first term is to impose the condition in Theorem 1 and the second term is to impose the condition that $C_h = \mathcal{S}$. Here, $\lambda_1 > 0$ and $\lambda_2 > 0$ are weights that balance two loss terms.

3.3 Adaptive sampling for estimation and control

After training, the loss function $\mathcal{L}$ might remain positive due to the large uncertainty of the Gaussian process at certain points. These points are called *hard samples*. For a hard sample $\hat{x}$, the set $g(\hat{x})$ is too large and satisfying the condition in Theorem 1 might be impossible. To conquer this problem, we sample more data around these hard samples to reduce uncertainty. We collect these samples

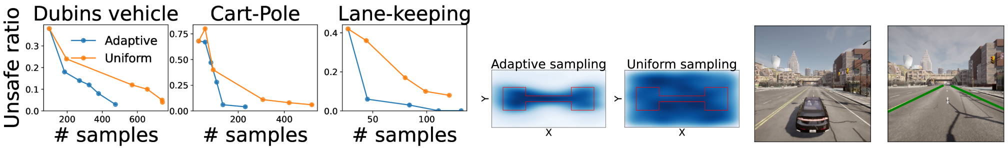

Figure 2: Image 1-3: Comparison of sample efficiency of learning certifiable controller. While both uniform and adaptive sampling approaches learn safety preserving controllers, the latter is significantly more sample efficient. The baseline for the three benchmarks are 0.58, 0.85, and 0.61 respectively. Image 4-5: Distribution of samples in the data set $\mathcal{D}$ constructed by two sampling approaches on Dubins vehicle. Red lines are the boundary of the target invariant set $\mathcal{S}$. Image 6-7: The simulated lane-keeping scenario in CARLA and the detected lanes.

into a set $\mathcal{H}$, and $\mathcal{H}$ is then merged with the current $\mathcal{D}$ to improve the GP. The algorithm returns the control module c if it successfully finds one, otherwise, it returns some debug information to the user such that the perception function can be improved accordingly in a separate procedure. The debug information is simply the set $\mathcal{H}$ in the last iteration.

Algorithm 1: Adaptive sampling.

```
Input: Max number of iteration: $I$; Confidence $\delta$.
Output: $\pi$ and $g$, or $\mathcal{H}$.
Randomly initialize $\mathcal{D} = \{(\hat{x}_1, x_1), \cdots, (\hat{x}_N, x_N)\}$;
$i \leftarrow 0$;
do
    Compute the state estimator $g$ on $\mathcal{D}$;
    Construct $\mathcal{D}_c$ and train $h$ and $\pi$ on $\mathcal{D}_c$;
    Collect hard samples $\mathcal{H}$;
    $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{H}$; $i \leftarrow i + 1$;
while $\mathcal{H} \neq \varnothing$ and $i < I$;
```

4 Experiments

We evaluated the proposed approach on three dynamical systems, two simple systems with synthetic perception error functions, and one realistic lane-keeping task in CARLA with deep-learning-based perception modules [7]. As a performance index for the learned controller, we report the *unsafe ratio*, which empirically measures the fraction of finite-time (10 seconds in our experiments) trajectories that exit the invariant set $\mathcal{S}$.

In Figure 2, we show how the unsafe ratio varies with the number of samples in the data set $\mathcal{D}$. We compared the proposed adaptive sampling approach with uniform sampling, where instead of enlarging the data set $\mathcal{D}$ with hard samples we use data points that are uniformly sampled from the state space, and the baseline, where GP is disabled and the state estimator is set to $g(\hat{x}) = \{\hat{x}\}$. There are several observations from the experiments. (1) Both sampling methods find controllers that are robust to the perception error. (2) Adaptive sampling leads to a lower unsafe ratio than uniform sampling with the same number of samples in $\mathcal{D}$, which empirically confirms our intuition that the proposed controller design approach with adaptive sampling is more sample efficient. To further illustrate the benefit of adaptive sampling, we contrast the data sets $\mathcal{D}$ generated by adaptive and uniform sampling in Figure 2. As can be seen from the figure, adaptive sampling zooms in on relevant areas of the state space, because at those areas, a more accurate state estimator is needed, while uniform sampling evenly distributes its sampling budget.

5 Limitations and future work

One should be careful with interpreting the theoretical guarantee of the proposed approach. The confidence δ characterizes the probability of each standalone state falling into the high-confidence set. Further theoretical treatment is required to boost such a guarantee to one on the safety of trajectories. Furthermore, in order to handle larger data sets, approximate GP such as [8] has to be used.

References

[1] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4693–4700.

[2] F. Sadeghi and S. Levine, “Cad2rl: Real single-image flight without a single real image,” in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017.

[3] S. Dean, N. Matni, B. Recht, and V. Ye, “Robust guarantees for perception-based control,” in *Learning for Dynamics and Control*. PMLR, 2020, pp. 350–360.

[4] S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames, “Guaranteeing safety of learned perception modules via measurement-robust control barrier functions,” *arXiv preprint arXiv:2010.16001*, 2020.

[5] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, “Most likely heteroscedastic gaussian process regression,” in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 393–400.

[6] D. Sun, N. Musavi, G. Dullerud, S. Shakkottai, and S. Mitra, “Learning certifiably robust controllers using fragile perception,” *arXiv preprint arXiv:2209.11328*, 2022.

[7] D. Wu, M. Liao, W. Zhang, X. Wang, X. Bai, W. Cheng, and W. Liu, “Yolop: You only look once for panoptic driving perception,” 2021.

[8] J. Hensman, A. Matthews, and Z. Ghahramani, “Scalable variational gaussian process classification,” in *Artificial Intelligence and Statistics*. PMLR, 2015, pp. 351–360.