Robust Guarantees for Perception-Based Control

Sarah Dean, Nikolai Matni, Benjamin Recht, Vickie Ye
University of California Berkeley

Abstract

Motivated by vision based control of autonomous vehicles, we consider the problem of controlling a known linear dynamical system for which partial state information, such as vehicle position, can only be extracted from high-dimensional data, such as an image. Our approach is to learn a perception map from high-dimensional data to partial-state observation and its corresponding error profile, and then design a robust controller. We show that under suitable smoothness assumptions on the perception map and generative model relating state to high-dimensional data, an affine error model is sufficiently rich to capture all possible error profiles, and can further be learned via a robust regression problem. We then show how to integrate the learned perception map and error model into a novel robust control synthesis procedure, and prove that the resulting perception and control loop has favorable generalization properties. Finally, we illustrate the usefulness of our approach on a synthetic example and on the self-driving car simulation platform CARLA.

1 Introduction

Incorporating insights from rich, perceptual sensing modalities such as cameras remains a major challenge in controlling complex autonomous systems. While such sensing systems clearly have the potential to convey more information than simple, single output sensor devices, interpreting and robustly acting upon the high-dimensional data streams remains difficult. For this type of sensing, one can view the design space of algorithms available to practitioners as lying between two extremes: at one extreme, there are purely data-driven approaches that attempt to learn an optimized map from perceptual inputs directly to low-level control decisions. Such approaches have seen tremendous success in accomplishing sophisticated tasks that were once thought to be well beyond the realm of autonomous systems, although critical gaps in understanding their robustness and safety still remain [36]. At the other extreme, there are methods rooted in classical system identification and robust control, wherein an intricate and explicit model of the underlying system and its environment is characterized, and subsequently used inside of a feedback control loop. Such methods have provided strong and rigorous guarantees of robustness and safety in domains such as aerospace and process control, but they have thus far had limited impact in domains with highly complex systems and environments, such as agile robotics and autonomous vehicles.

In this paper, we attempt to bridge the gap between these two camps, proposing a methodology for using perceptual information in complex control loops. Whereas much recent work has been devoted to proving safety and performance guarantees for learning-based controllers applied to systems with unknown dynamics [1, 2, 4, 5, 12, 16, 21, 22, 31, 44, 58, 60, 62], we focus on the practical scenario where the underlying dynamics of a system are well understood, and it is instead the interaction with a perceptual sensor that is the limiting factor. Specifically, we consider controlling a known linear dynamical system for which partial state information can only be extracted from high-dimensional observations. Our approach is to design a *virtual sensor* by learning a perception map, i.e., a map from high-dimensional observations to a subset of the state, and crucially to quantify its error profile. We show that under suitable smoothness assumptions, a linear parameterization of the error profile is valid within a neighborhood of the training data. This linear model of uncertainty is then used

Preprint. Under review.

to synthesize a robust controller that ensures that the system does not deviate too far from states visited during training. Finally, we show that the resulting perception and robust control loop is able to robustly generalize under adversarial noise models. To the best of our knowledge, this is the first such guarantee for a vision based control system.

1.1 Related work

Vision based estimation, planning, and control There is a rich body of work, spanning several research communities, that integrate high-dimensional sensors, specifically cameras, into estimation, planning, and control loops. The robotics community has focussed mainly on integrating camera measurements with inertial odometry via an Extended Kalman Filter (EKF) [30, 32, 33]. Similar approaches have also been used as part of Simultaneous Localization and Mapping (SLAM) algorithms in both ground [40] and aerial [39] vehicles. We note that these works focus solely on the estimation component, and do not consider downstream use of state estimates in control loops. In contrast, the papers [37, 38, 57] all demonstrate techniques that use camera measurements to aid inertial position estimates to enable aggressive control maneuvers in unmanned aerial vehicles.

The machine learning community has taken a more data-driven approach. The earliest such example is likely [49], in which a 3-layer neural-network is trained to infer road direction from images. Modern approaches to vision based planning, typically relying on deep neural networks, include learning maps from image to trail direction [27], learning Q-functions for indoor navigation using 3D CAD images [53], and using images to specify waypoints for indoor robotic navigation [11]. Moving from planning to low-level control, end-to-end learning for vision based control has been achieved through imitation learning from training data generated via human [15] and model predictive control [47]. The resulting policies map raw image data directly to low-level control tasks. In [18], higher level navigational commands, images, and other sensor measurements are mapped to control actions via imitation learning. Similarly, in [62] and related works, image and inertial data is mapped to a cost landscape, that is then optimized via a path integral based sampling algorithm. More closely related to our approach is [35], where a deep neural network is used to learn a map from image to system state – we note that this perception module is naturally incorporated into our proposed pipeline. To the best of our knowledge, none of the aforementioned results provide safety or performance guarantees.

Learning, robustness, and control Our theoretical contributions are similar in spirit to those of the online learning community, in that we provide generalization guarantees under adversarial noise models [6, 7, 29, 34, 63]. Similarly, [4] shows that adaptive disturbance feedback control of a linear system under adversarial process noise achieves sublinear regret – we note that this approach assumes full state information. We also draw inspiration from recent work that seeks to bridge the gap between linear control and learning theory. These assume a linear time invariant system, and derive finite-time guarantees for system identification [21, 25, 26, 28, 46, 48, 54–56, 59], and/or integrate learned models into control schemes with finite-time performance guarantees [1–3, 19, 21, 22, 41, 45, 50, 52].

1.2 Notation

We use letters such as x and A to denote vectors and matrices, and boldface letters such as $\boldsymbol{x}$ and $\boldsymbol{\Phi}$ to denote infinite horizon signals and linear convolution operators. For $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}$, we have by definition that $y_k = \sum_{t=0}^{k} \Phi_t x_{k-t}$. We write $x_{0:t} = \{x_0, x_1, \dots, x_t\}$ for the history of signal $\boldsymbol{x}$ up to time t. For a function $x_k \mapsto f_k(x_k)$, we write $\boldsymbol{f}(\boldsymbol{x})$ to denote the signal $\{f_k(x_k)\}_{k=0}^{\infty}$. We overload the norm $\|\cdot\|$ so that it applies equally to elements x_k, signals $\boldsymbol{x}$, and linear operators $\boldsymbol{\Phi}$, and assume that it satisfies: (i) $\|x_k\| \leq \|y_k\| + \|z_k\| \implies \|\boldsymbol{x}\| \leq \alpha(\|\boldsymbol{y}\| + \|\boldsymbol{z}\|)$ for $\alpha > 0$, and (ii) $\|\boldsymbol{\Phi}\| = \sup_{\|\boldsymbol{w}\| \leq 1} \|\boldsymbol{\Phi}\boldsymbol{w}\|$. The triple $(\|x_k\|_\infty, \|\boldsymbol{x}\|_\infty, \|\boldsymbol{\Phi}\|_{\mathcal{L}_1})$ satisfies these properties with $\alpha = 1$, as does the triple $(\|x_k\|_2, \|\boldsymbol{x}\|_{pow}, \|\boldsymbol{\Phi}\|_{\mathcal{H}_\infty})$ with $\alpha = \sqrt{2}$ (see Appendix A). As $\|\boldsymbol{\Phi}\|$ is an induced norm, it satisfies the sub-multiplicative property $\|\boldsymbol{\Phi}\boldsymbol{\Psi}\| \leq \|\boldsymbol{\Phi}\|\|\boldsymbol{\Psi}\|$. We let $[x]_+ = \max(x, 0)$.

2 Problem setting

Consider the LTI dynamical system

$$x_{k+1} = Ax_k + Bu_k + Hw_k \;, \tag{1}$$

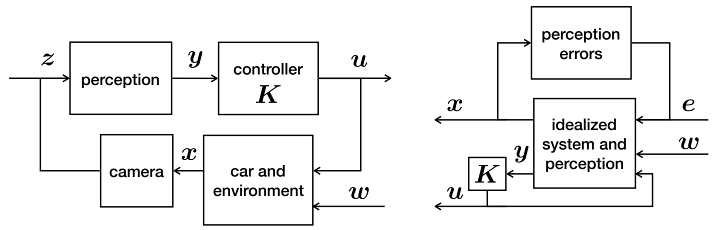

Figure 1: Sketch of proposed pipeline and conceptual robust-control rearrangement permitted through our perception error characterization.

with system state $x \in \mathbb{R}^n$, control input $u \in \mathbb{R}^m$, disturbance $w \in \mathbb{R}^w$, and known matrices (A, B, H). Without loss of generality, we assume that $\|H\| = 1$. Further assume that system (1) induces a corresponding high-dimensional process

$$z_k = q(x_k) + \Delta_{q,k}(x_k) + v_k \ , \tag{2}$$

where q is an unknown *generative model*, with time-varying nuisance variable components $\Delta_{q,k}(x_k)$ and v_k satisfying $\max_{\|x\|\leq 1} \|\Delta_{q,k}(x)\| \leq \varepsilon_q$, and $\|v_t\| \leq \varepsilon_v$, respectively. We typically assume that $N \gg n$. As an example, consider a camera affixed to the dashboard of a car tasked with driving along a road. Here, the high-dimensional $\{z_k\}$ are the captured images and the map q generates these images as a function of position and velocity. Nuisance variables such as lighting variations and occlusions are captured both by $\Delta_{q,k}(x_k)$ and v_k. Motivated by such a vision based control system, our goal is to solve the following optimal control problem

$$\begin{array}{rl} \text{minimize}_{\{\gamma_k\}} & c(\boldsymbol{x}, \boldsymbol{u}) \\ \text{subject to} & \text{dynamics (1) and measurement (2)},\ u_k = \gamma_k(z_{0:k}), \end{array} \tag{3}$$

where here $c(\boldsymbol{x}, \boldsymbol{u})$ is a suitably chosen cost function (see Appendix A), and γ_k is a measurable function of the image history $z_{0:k}$. This problem is made challenging by the high-dimensional, nonlinear, time-varying, and unknown generative model (2).

Suppose instead that there exists a *perception map* p such that $p(z_k) = Cx_k + e_k$ for $C \in \mathbb{R}^{\ell \times n}$ a known matrix, and $e_k \in \mathbb{R}^\ell$ an error term with known statistics. Here, the matrix C enforces that only partial state information can be extracted from a single observation. In the autonomous driving example, we might expect to predict position from a single image, but not velocity. Using this map, we define a new measurement model in which the map p plays the role of a noisy sensor:

$$y_k = p(z_k) = Cx_k + e_k. \tag{4}$$

This allows us to reformulate problem (3) as a *linear* optimal control problem, where now the measurements are defined by (4) and the control law $u_k = \pi(y_{0:k})$ is a *linear* function of the outputs of past measurements $y_{0:k}$. Linear optimal control problems are widely studied, and for a variety of cost functions and noise models, their solutions are well understood. Perhaps the most well known is the combination of *Kalman filtering* with static state feedback, which arises as the solution to the linear quadratic Guassian (LQG) problem. Different control costs and assumptions give rise to different estimation and control strategies: a brief summary of these is given in Appendix A.

In light of this discussion, we can now decompose our problem into two tasks. First, collect training data pairs $\{x_{0:T}, z_{0:T}\}$ and learn a perception map p and corresponding error profile e_k such that the measurement model (4) is valid. Second, compute a robust controller that mitigates the effects of the measurement error e_k. We illustrate the resulting control architecture in Figure 1, and highlight that in contrast to standard certainty equivalent approaches in which an extended or unscented Kalman Filter is used with a state-feedback control law, we explicitly quantify perception and sensing error from data, and use this error characterization to synthesize a robust controller. In the following, we show that under suitable Lipschitz assumptions on the generative model (2) and perception map p, we can successfully accomplish these two tasks using linear error models and robust control.

3 Learning a perception map and its error model

We revisit the measurement model (4), and show that under suitable assumptions, an affine error model is completely general. We then build on this observation to formulate a novel training method that simultaneously learns a perception map and its corresponding error model, and show that it robustly generalizes in a way that depends on the smoothness of the underlying generative process.

Affine error model We assume that there exists an idealized perception map $p^\star$ such that $p^\star(q(x_k)) = Cx_k$,[1] and that the maps p and q are L_p and L_q Lipschitz. We then rewrite

$$y = p(z_k) = Cx_k + \Delta_{C,k}(x_k) + \eta_k, \tag{5}$$

where

$$\begin{aligned}\Delta_{C,k}(x_k) &:= p(q(x_k)) - p^\star(q(x_k)) + p(q(x_k) + \Delta_{q,k}(x_k) + v_k) - p(q(x_k) + v_k)\\ \eta_k &:= p(q(x_k) + v_k) - p(q(x_k)).\end{aligned} \tag{6}$$

Here we have used the generative model (2) and the idealized perception assumption. Note that $\Delta_{C,k}$ is composed of two terms: the first captures the error in our perception map p with respect to the idealized $p^\star$, whereas the second captures the effects of state-dependent nuisance variables $\Delta_{q,k}(x_k)$.

We now make two observations that will motivate our training procedure. First, notice that without loss of generality we can take $\Delta_{C,k}$ to be a time-varying linear operator: for any desired error process $\{\Delta_{C,k}(x_k)\} = \{\nu_k\}$, it suffices to set $\Delta_{C,k} = \nu_k x_k^\top (x_k^\top x_k)^{-1}$. Second, under the Lipschitz assumptions on the maps p, q, and $\Delta_{q,k}$, it follows immediately that: (1) $\Delta_{C,k}$ is a uniformly L_Δ-Lipschitz map, with $L_\Delta \leq L_p(L_q + \varepsilon_q) + \|C\|$, and (2) η_k is a norm bounded perturbation satisfying $\|\eta_k\| \leq L_p \varepsilon_v$. Thus, we parameterize our error model e_k as being an affine function of the state x_k,

$$e_k = \Delta_{C,k} x_k + \eta_k, \tag{7}$$

and seek to find the smallest perturbations $\Delta_{C,k} x_k$ and η_k such that the discrepancies of our perception map on the training data is captured. The generative model that we present is important only insofar as it suggests the error decomposition and ultimate reduction to an affine model. Our results apply to any observation process that gives rise to an error model as in (7).

Training To learn a perception map and an error model, we consider the supervised learning setting. We assume access to perfect measurements $y_k = Cx_k$, and note that so long as the pair (A, C) is observable, these measurements allow for the state to be computed exactly, albeit with a delay. Therefore, during training, we record state observation pairs $\{x_k, z_k\}$. As will become clear in the next section, further details on the distribution or generation of this data need only be specified in relation to the desired closed-loop behavior of the system.

Consider the case that the perception map p is provided, and thus our goal is reduced to fitting an affine error model. We begin by observing that for any error model that is valid on the training data, i.e., for any error model satisfying $p(z_k) - Cx_k = \Delta_{C,k} x_k + \eta_k$ for all sampled points, it immediately follows that

$$\|p(z_k) - Cx_k\| \leq \left(\max_t \|\Delta_{C,t}\|\right) \|x_k\| + \max_t \|\eta_t\| \;\forall k.$$

In particular, this observation means that in $(\|x\|, \|p(z_k) - Cx_k\|)$ space, all pointwise errors lie below the line with slope ε_C and intercept ε_η, where ε_C and ε_η are such that $\|\Delta_{C,k}\| \leq \varepsilon_C, \; \|\eta_k\| \leq \varepsilon_\eta$ for all times k. In fact, an error model (7) bounded by $(\varepsilon_C, \varepsilon_\eta)$ exists for a perception map p if and only if for all pairs (x, z) in the dataset, $\|p(z) - Cx\| \leq \varepsilon_C \|x\| + \varepsilon_\eta$ (see Proposition 7 in Appendix C).

With this discussion in mind, we propose fitting the error profile by solving:

$$\begin{array}{rl}\text{minimize}_{\varepsilon_C, \varepsilon_\eta} & \varepsilon_C M + \varepsilon_\eta \\ \text{subject to} & \|p(z_k) - Cx_k\| \leq \varepsilon_C \|x_k\| + \varepsilon_\eta \;\; \forall\, k\end{array} \tag{8}$$

where with $M := \frac{1}{T}\sum_{k=1}^T \|x_k\|$. Thus we minimize an upper bound on the average perception error.

This formulation is equally applicable when the perception map must be learned from data. Here, we add an additional minimization over p and augment the objective of optimization problem (8) with a regularizer $R(p)$ to enforce smoothness:

$$\begin{array}{rl}\text{minimize}_{\varepsilon_C, \varepsilon_\eta, p} & \varepsilon_C M + \varepsilon_\eta + \lambda R(p) \\ \text{subject to} & \|p(z_k) - Cx_k\| \leq \varepsilon_C \|x_k\| + \varepsilon_\eta \;\; \forall\, k\end{array} \tag{9}$$

This optimization problem seeks to jointly find a small error profile and smooth perception map that perfectly explain the training data. We illustrate these concepts on a simple linear model.

[1] Though we restrict the exposition to this idealized setting, our results extend naturally if the equivalence holds only approximately.

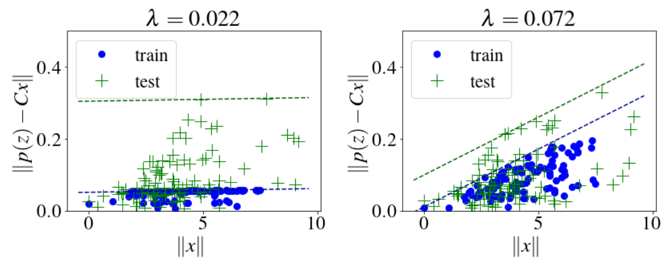

Figure 2: Plotting the perception errors $\|p(x) - Cx\|_\infty$ as a function of the state norm $\|x\|_\infty$ illustrates an affine error profile. Larger regularization parameter λ (right) leads to a smaller gap between the train and test sets.

Example 1 (Linear generative model)**.** Consider the linear time varying generative model

$$z_k = (G_0 C + \Delta_{G,k})x_k + \nu_k \; , \tag{10}$$

with $\|G_0 C\|_{\mathcal{L}_1} = 1$, and at each timestep k, $\|\Delta_{G,k}\|_{\mathcal{L}_1} \leq 0.5$ and $\|\nu_k\|_\infty \leq 0.05$. Figure 2, shows the error profiles for linear perception functions $p(x) = Px$ trained using (9) with different regularization parameters and $R(p) = \|P\|_{\mathcal{L}_1}$. We use $z_k \in \mathbb{R}^{500}$ and training and test trajectories of length $T = 100$ generated by the 2D double integrator system described in Section 5.

As we have assumed that the perception and generative maps are Lipschitz, we can immediately bound the generalization error of a learned model within a neighborhood of the training data.

Lemma 1 (Closeness implies generalization)**.** *Let L_Δ denote the Lipschitz constant of the true state-dependent error term* (6)*. Then for any new state and observation (x, z) and any training data state x_d*

$$\|p(z) - Cx\| \leq \varepsilon_C \|x\| + \varepsilon_\eta + (L_\Delta + \varepsilon_C)\|x - x_d\| + 2L_p \varepsilon_v.$$

The proof is presented in Appendix C. Returning to our interpretation of the error model as a line in $(\|x\|, \|p(z) - Cx\|)$ space, Lemma 1 says that for unseen states x, it suffices to shift the y-intercept of the learned error model line up by a term which depends on its distance from the training data. Figure 2 illustrates this idea on simulated data. We emphasize that our approach is parameterization agnostic, and can also be used to characterize error profiles for existing vision systems.

4 Analysis and synthesis of perception-based controllers

The local generalization result in Lemma 1 is useful only if the system remains close to states visited during training. To this end, we show in Lemma 2 that we can remain close to training data if the error model generalizes well. By then enforcing that the composition of the bounds in Lemmas 1 and 2 is a contraction, a natural notion of controller robustness emerges that guarantees favorable behavior and generalization. To do so, we adopt an adversarial noise model and exploit that we can design system behavior to bound how far the system deviates from states visited during training.

Robust control for generalization Once the control input to dynamical system (1) is defined to be a linear function of the measurement (4), the closed-loop behavior is determined entirely by the process noise $\boldsymbol{w}$ and the measurement noise $\boldsymbol{e}$ (as in Figure 1). For any controller that is a linear function of the history of system outputs, we can write the system state and input directly as a linear function of the noise

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi_{xx}} & \boldsymbol{\Phi_{xy}} \\ \boldsymbol{\Phi_{ux}} & \boldsymbol{\Phi_{uy}} \end{bmatrix} \begin{bmatrix} H\boldsymbol{w} \\ \boldsymbol{e} \end{bmatrix} . \tag{11}$$

In what follows, we will state results in terms of these system response variables. The connection between these maps and a feedback control law $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{y}$ that achieves the response (11) is formalized in the *System Level Synthesis* (SLS) framework. Roughly, SLS states that for any system response $\{\boldsymbol{\Phi_{xx}}, \boldsymbol{\Phi_{xy}}, \boldsymbol{\Phi_{ux}}, \boldsymbol{\Phi_{uy}}\}$ constrained to lie in an affine space defined by the system dynamics, there exists a linear feedback controller $\boldsymbol{K}$ that achieves the response (11). The SLS parametrization thus makes explicit the effects of errors $\boldsymbol{e}$ on system behavior – details are found in Appendix B.

Let $(p, \varepsilon_C, \varepsilon_\eta)$ denote the optimal solution to the robust learning problem (9). For a state-observation pair (x, z) define the generalization error as

$$\delta := Cx - p(z) - \Delta_C x - \eta, \tag{12}$$

where Δ_C and η are set to minimize the norm of δ. For any (x_d, z_d) in the training error, we will have $\delta = 0$. Rewriting expression (12) as $e = Cx - p(z) = \Delta_C x + \eta + \delta$ makes clear that we can view the generalization error δ as introducing additional additive noise to the error model.

While the additive η and δ can handled with standard linear control methods, the state dependent errors can be viewed as time varying perturbations $\Delta_{C,k}$ to the sensing matrix C, and must be handled more carefully. In Appendix B, we show how this can be done using a robust version of the SLS parameterization. The analysis relies on a small-gain like condition on the uncertainty introduced by $\Delta_{C,k}$ into $\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}$, the *nominal* map from additive measurement error to state $\boldsymbol{x}$, designed to lie in the affine space defined by the dynamics (A, B, C). This results in the robust stability constraint

$$\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\| < \frac{1}{\varepsilon_C}\,. \tag{13}$$

We now show how such a robustly stabilizing controller can be used to bound deviations of states seen at test time from those visited during training as a function of the generalization error norm $\|\boldsymbol{\delta}\|$.

Lemma 2 (Generalization implies closeness)**.** *Let $(p, \varepsilon_C, \varepsilon_\eta)$, $\Delta_{C,k}$, η_k, and $\boldsymbol{\delta}$ be as above, let $\{\widehat{\mathbf{\Phi}}_{\boldsymbol{xx}}, \widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}, \widehat{\mathbf{\Phi}}_{\boldsymbol{ux}}, \widehat{\mathbf{\Phi}}_{\boldsymbol{uy}}\}$ lie in the affine space defined by dynamics (A, B, C) and satisfy the robust stability constraint* (13)*, and let $\widehat{\boldsymbol{K}}$ be the associated controller. Then the state trajectory $\boldsymbol{x}$ achieved by the control law $\boldsymbol{u} = \widehat{\boldsymbol{K}}p(\boldsymbol{z})$ and driven by noise process $\boldsymbol{w}$, satisfies*

$$\|\boldsymbol{x} - \boldsymbol{x}_d\| \le \frac{G_x + \varepsilon_C\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|\|\boldsymbol{x}_d\| + \|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|\|\boldsymbol{\delta}\|}{1 - \varepsilon_C\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|} \tag{14}$$

for $\boldsymbol{x}_d$ a trajectory populated with training states x_d, and G_x any constant satisfying

$$G_x \ge \|\widehat{\mathbf{\Phi}}_{\boldsymbol{xx}} H\boldsymbol{w} + \widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\boldsymbol{\eta} - \boldsymbol{x}_d\|\,. \tag{15}$$

The terms in the numerator of the bound (14) capture different generalization properties. The first, G_x, is a measure of nominal similarity of behavior between training and test time. If we plan to visit states during operation that are similar to those seen during training, this term will be small, and indeed in Propositions 4 and 5, we give explicit training and testing scenarios under which this holds true. The third term, $\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|\|\boldsymbol{\delta}\|$, is a measure of the robustness of our nominal system to additional sensor error introduced by the generalization error $\boldsymbol{\delta}$. Finally, the middle term $\varepsilon_C\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|\|\boldsymbol{x}_d\|$ and denominator capture the robustness of our system to mis-specifications in the sensing matrix C.

We are now in a position to state the main result of the paper, which shows that under an additional robustness condition, Lemmas 1 and 2 combine to define an invariant set around the training neighborhood within which we can bound the generalization error $\boldsymbol{\delta}$.

Theorem 3. *Let the assumptions of Lemmas 1 and 2 hold. Then as long as*

$$\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\| < \frac{1}{\varepsilon_C + \alpha(L_\Delta + \varepsilon_C)}, \tag{16}$$

we have that all trajectories $(\boldsymbol{x}, \boldsymbol{z})$ remain close to training states:

$$\|\boldsymbol{x} - \boldsymbol{x}_d\| \le \frac{G_x + (\varepsilon_C\|\boldsymbol{x}_d\| + 2\alpha L_p\varepsilon_v)\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|}{1 - \|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|(\varepsilon_C + \alpha(L_\Delta + \varepsilon_C))} \tag{17}$$

and are well approximated by the learned perception map and error model:

$$\min_{\substack{\|\Delta_{C,k}\| \le \varepsilon_C,\\ \|\eta_k\| \le \varepsilon_\eta}} \|p(\boldsymbol{z}) - (C\boldsymbol{x} + \boldsymbol{\Delta}_C(\boldsymbol{x}) + \boldsymbol{\eta})\| \le \frac{G_x + 2\alpha L_p\varepsilon_v + \varepsilon_C\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|(\|\boldsymbol{x}_d\| - 2\alpha L_p\varepsilon_v)}{1 - \|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|(\varepsilon_C + \alpha(L_\Delta + \varepsilon_C))}. \tag{18}$$

Theorem 3 shows that bound (16) should be used during controller synthesis to ensure generalization. In Appendix D, we present a robust controller synthesis problem, and bound the cost of a system operating under these uncertainties. Feasibility depends on the controllability and observability of the nominal system (A, B, C), which impose limits on how small $\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\|$ can be made to be, and on the size of the error model, as captured by ε_C. We now describe modes of system operation and corresponding training strategies that suggest additional controller synthesis constraints.

Dense sampling We specialize to the $\ell_\infty/\mathcal{L}_1$ norms for this result only, but note that the argument can be extended to the power norm at the expense of a $\sqrt{n}$ factor.

Proposition 4 (Dense sampling)**.** *Suppose that the training data states $\mathcal{X}_d := \{x_d\}$ form an ε_d-net over the norm ball of radius R,[2] such that*

$$\min_{x_d\in\mathcal{X}_d} \|x_d - x\|_\infty \leq \varepsilon_d \ \forall \ \|x\|_\infty \leq R. \tag{19}$$

Then under the assumptions of Theorem 3, we achieve the bounds (17) *and* (18) *with*

$$G_x = \left[\|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H\|_{\mathcal{L}_1} + \varepsilon_\eta \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|_{\mathcal{L}_1} - R\right]_+ + \varepsilon_d. \tag{20}$$

A constraint on the term (20) is easily added to the synthesis problem, and therefore this proposition states that so long as we operate within a well-sampled subset of the state-space, we generalize well.

Imitation learning Next, we instead consider a scenario in which a collection of periodic tasks is specified at training time. Each task has an associated reference trajectory specified by a disturbance sequence driving the system, $w_{0:T-1}^{(s)} := \{w_0^{(s)}, \dots, w_T^{(s)}\}$, where $w_0^{(s)} = w_T^{(s)}$, and the bound $\|w_k^{(s)}\| \leq \varepsilon_w$ describes the how rapidly the reference trajectory can vary (see Appendix A). We may also define $w_{0:T-1}^{(s)}$ to include unknown but bounded process noise. Then we define $\boldsymbol{w}^{(s)} = \{w_{0:T-1}^{(s)}, w_{0:T-1}^{(s)}, \dots\}$. With this imitation learning-like scenario in mind, our exploration strategy is to fix a stabilizing controller $\boldsymbol{K}$ and corresponding system response $\{\boldsymbol{\Phi}_{\boldsymbol{xx}}, \boldsymbol{\Phi}_{\boldsymbol{xy}}, \boldsymbol{\Phi}_{\boldsymbol{ux}}, \boldsymbol{\Phi}_{\boldsymbol{uy}}\}$, and to drive the system with the disturbances $w_{0:T-1}^{(s)}$ to generate training trajectories $\{x_{0:T}^{(s)}, z_{0:T}^{(s)}\}$.

Proposition 5 (Imitation learning)**.** *Let the training data be generated as describe above. Then for any task specified by $\mathbf{w}$, with task similarity $\|\boldsymbol{w}^{(s)} - \boldsymbol{w}\| \leq \varepsilon_r$ for some $\boldsymbol{w}^{(s)}$ in the training set, controlling the system with $\widehat{\boldsymbol{K}}$ satisfying the assumptions of Theorem 3 achieves the bounds* (17) *and* (18) *with*

$$G_x = \varepsilon_r \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H\| + \varepsilon_w \alpha \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H - \boldsymbol{\Phi}_{\boldsymbol{xx}} H\| + \varepsilon_\eta \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|. \tag{21}$$

Thus we generalize well to periodic tasks similar to those performed during training if the controller is similar to that used during training. This suggests using a training controller $\boldsymbol{K}$ with small $\|\boldsymbol{\Phi}_{\boldsymbol{xy}}\|$, such that it may (nearly) satisfy the constraint (16). Further, although the training tasks are finite, their periodicity allows us to guarantee performance over an infinite time horizon.

Standard generalization is hard We remark that standard notions of statistical generalization are challenging to adapt to the problem considered here. First note that if we collect data using one controller, and then use this data to build a new controller, there will be a distribution shift in the observations seen between the two controllers. Any statistical generalization bounds on performance must necessarily account for this shift. Second, from a more practical standpoint, most generalization bounds require knowing instance specific quantities governing properties of the class of functions we use to fit a predictor. Hence, they will include constants that are not measurable in practice. This issue can perhaps be mitigated using some sort of bootstrap technique for post-hoc validation. However, we note that the sort of bounds we aim to bootstrap are worst case, not average case. Indeed, the bootstrap typically does not even provide a consistent estimate of the maximum of independent random variables, see for instance [13], and Ch 9.3 in [17]. Other measures such as conditional value at risk [51] require billions of samples to guarantee five 9s of reliability. We highlight these issues simply to point out that adapting statistical generalization to robust control remains an active area with many open challenges to be considered in future work.

5 Experiments

All code needed to reproduce our experimental results will be publicly released, and can currently be downloaded at `http://robust-vision-control.s3.amazonaws.com/public.zip`. We demonstrate our results in an imitation learning context using a synthetic example and a complex

[2] It is standard that $O(1/\varepsilon_d^n)$ such points suffice. This dependence can be reduced if a subset of the states are known to remain within pre-specified ranges (e.g., if velocity is regulated around a constant value).

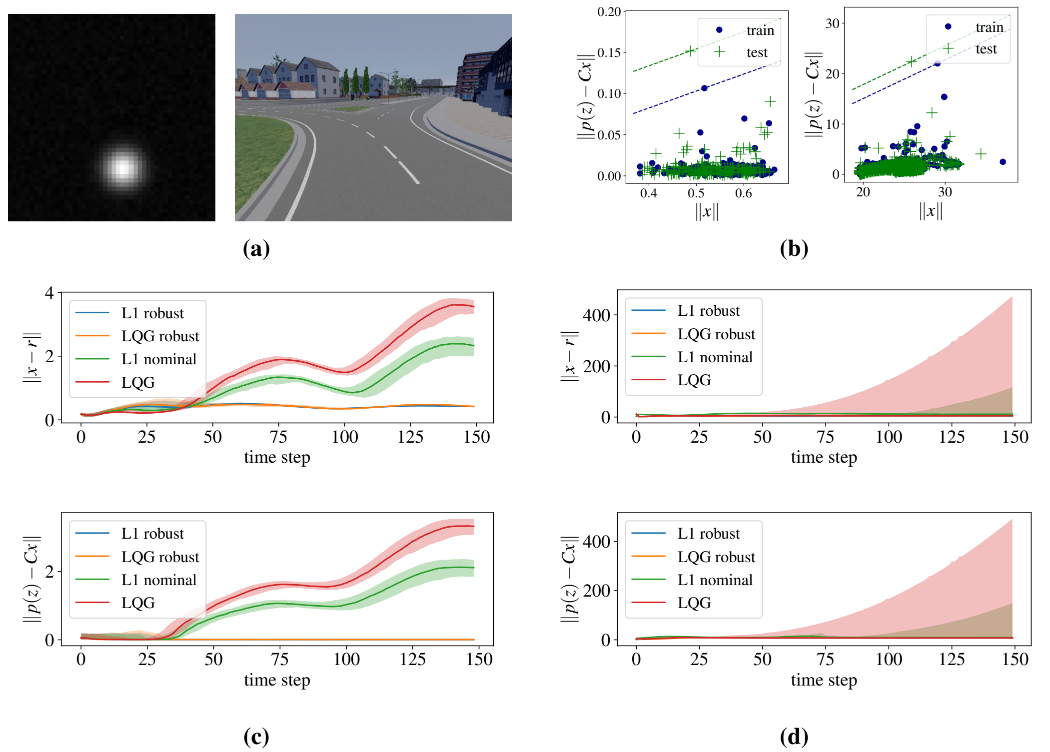

Figure 3: Experimental setup and results: (a) Visual inputs $\{z_t\}$ for the synthetic (left) and CARLA (right) examples, (b) Error model fits in $(\|x\|_\infty, \|p(x) - Cx\|_\infty)$ space on train and test trajectories for the synthetic (left) and simulated vehicle (right) examples, (c-d) Median, upper, and lower quartiles of ℓ_∞ tracking and estimation error for (c) 200 rollouts of the synthetic and (d) 100 rollouts of the CARLA examples.

simulation-based example. In particular, we compare the behavior of using a *nominal controllers* which do not take into account sensitivity to the nonlinearity in the measurement model. For the synthetic example, we consider generated images of a moving blurry white circle on a black background; the goal is to move the system in a circle of radius 1. We also consider an example using dashboard camera images from a vehicle simulated using the CARLA platform [23] and the goal is to drive around a track. Figure 3a shows representative images seen by the controllers.

For both systems, we set the underlying dynamics to be two dimensional double integrators, where the x and y dimensions move independently, i.e., for each dimension $i = 1, 2$, we set

$$x_{k+1}^{(i)} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} x_k^{(i)} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k^{(i)} ,$$

and the full state is then given by $x_k^\top = [(x_k^{(1)})^\top \, (x_k^{(2)})^\top]$. For all examples, the sensing matrix C extracts the position of the system, i.e., $Cx_k = [x_{1,k}^{(1)}, x_{1,k}^{(2)}]$. Our training, validation, and controller synthesis procedures are detailed in Appendix E. For the synthetic example, we jointly learn a linear perception map $p(x) = Px$ and error model using optimization problem (9). In CARLA experiments, we use ORB SLAM 2 [43] as a black box perception map, and fit an error model using optimization problem (8). Figures 3b show the learned error profiles for the synthetic (left) and vehicle (right) examples. We note that although the ORB SLAM 2 perception map used in the CARLA simulations may not satisfy the assumptions of Theorem 3 when the feature matching step fails, we nevertheless observe safe system behavior, suggesting that under our robust controller, no such mismatches occur. For both systems, we compare the behavior of naively synthesized LQG and $\mathcal{L}_1$ optimal controllers with that achieved by robust $\mathcal{L}_1$ and LQG controllers designed with our proposed pipeline. LQG is a standard control scheme that explicitly separates state estimation (Kalman Filtering) from control (LQR control), and is emblematic of much of standard control practice. $\mathcal{L}_1$ optimal control minimizes worst case state deviation and control effort by modeling process and sensor errors as ℓ_∞ bounded adversarial processes. Both LQG and $\mathcal{L}_1$ optimal control are described in more detail in Appendix

A. As shown in the top figures of 3c and 3d, nominal controllers are unable to accurately track the reference trajectory and diverge, whereas the robust controllers remain within a bounded distance of the reference trajectory. The bottom figures of 3c and 3d demonstrate the corresponding degradation in accuracy of the perception maps as the systems deviate from the training data.

6 Conclusions

Though standard practice is to treat the output of a perception module as an ordinary signal, we have demonstrated both in theory and experimentally that accounting for the inherent uncertainty of perception based sensors can dramatically improve the performance of the resulting control loop. Moreover, we have shown how to quantify and account for such uncertainties with tractable data-driven safety guarantees. We hope to extend this study to the control of more complex systems, and to apply this framework to standard model-predictive control pipelines which form the basis of much of contemporary control practice.

References

[1] Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26, 2011.

[2] Yasin Abbasi-Yadkori, Nevena Lazic, and Csaba Szepesvári. Model-free linear quadratic control via reduction to expert prediction. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3108–3117, 2019.

[3] Marc Abeille and Alessandro Lazaric. Improved regret bounds for thompson sampling in linear quadratic control problems. In *International Conference on Machine Learning*, pages 1–9, 2018.

[4] Naman Agarwal, Brian Bullins, Elad Hazan, Sham M Kakade, and Karan Singh. Online control with adversarial disturbances. *arXiv preprint arXiv:1902.08721*, 2019.

[5] Anayo K Akametalu, Jaime F Fisac, Jeremy H Gillula, Shahab Kaynama, Melanie N Zeilinger, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control*, pages 1424–1431. IEEE, 2014.

[6] Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. Online learning for time series prediction. In *Conference on learning theory*, pages 172–184, 2013.

[7] Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: price of past mistakes. In *Advances in Neural Information Processing Systems*, pages 784–792, 2015.

[8] James Anderson and Nikolai Matni. Structured state space realizations for sls distributed controllers. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 982–987. IEEE, 2017.

[9] James Anderson, John C Doyle, Steven Low, and Nikolai Matni. System level synthesis. *arXiv preprint arXiv:1904.01634*, 2019.

[10] MOSEK ApS. *MOSEK Optimizer API for Python Release 9.0.88*, 2019. URL `https://docs.mosek.com/9.0/pythonapi.pdf`.

[11] Somil Bansal, Varun Tolani, Saurabh Gupta, Jitendra Malik, and Claire Tomlin. Combining optimal control and learning for visual navigation in novel environments. *arXiv preprint arXiv:1903.02531*, 2019.

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

[13] Peter J Bickel, David A Freedman, et al. Some asymptotic theory for the bootstrap. *The annals of statistics*, 9(6):1196–1217, 1981.

[14] Ross Boczar, Nikolai Matni, and Benjamin Recht. Finite-data performance guarantees for the output-feedback control of an unknown system. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2994–2999. IEEE, 2018.

[15] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[16] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. *arXiv preprint arXiv:1903.08792*, 2019.

[17] Michael R Chernick. *Bootstrap methods: A guide for practitioners and researchers*, volume 619. John Wiley & Sons, 2011.

[18] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[19] Alon Cohen, Tomer Koren, and Yishay Mansour. Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret. *arXiv preprint arXiv:1902.06223*, 2019.

[20] M. Dahleh and J. Pearson. ℓ^1-optimal feedback controllers for MIMO discrete-time systems. *IEEE Transactions on Automatic Control*, 32(4):314–322, April 1987. ISSN 0018-9286. doi: 10.1109/TAC.1987.1104603.

[21] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *arXiv preprint arXiv:1710.01688*, 2017.

[22] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator. In *Advances in Neural Information Processing Systems*, pages 4192–4201, 2018.

[23] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[24] Bogdan Dumitrescu. *Positive Trigonometric Polynomials and Signal Processing Applications*, volume 103. Springer, 2007.

[25] Salar Fattahi and Somayeh Sojoudi. Data-driven sparse system identification. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 462–469. IEEE, 2018.

[26] Salar Fattahi, Nikolai Matni, and Somayeh Sojoudi. Learning sparse dynamical systems from a single sample trajectory. *arXiv preprint arXiv:1904.09396*, 2019.

[27] Alessandro Giusti, Jérôme Guzzi, Dan C Cireşan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2015.

[28] Moritz Hardt, Tengyu Ma, and Benjamin Recht. Gradient descent learns linear dynamical systems. *The Journal of Machine Learning Research*, 19(1):1025–1068, 2018.

[29] Babak Hassibi and Thomas Kaliath. H-∞ bounds for least-squares estimators. *IEEE Transactions on Automatic Control*, 46(2):309–314, 2001.

[30] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.

[31] Lukas Hewing and Melanie N Zeilinger. Cautious model predictive control using gaussian process regression. *arXiv preprint arXiv:1705.10702*, 2017.

[32] Eagle S Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4): 407–430, 2011.

[33] Jonathan Kelly and Gaurav S Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1): 56–79, 2011.

[34] Vitaly Kuznetsov and Mehryar Mohri. Time series prediction and online learning. In *Conference on Learning Theory*, pages 1190–1213, 2016.

[35] Alexander Lambert, Amirreza Shaban, Amit Raj, Zhen Liu, and Byron Boots. Deep forward and inverse perceptual models for tracking and prediction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 675–682. IEEE, 2018.

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

[37] Yi Lin, Fei Gao, Tong Qin, Wenliang Gao, Tianbo Liu, William Wu, Zhenfei Yang, and Shaojie Shen. Autonomous aerial navigation using monocular visual-inertial fusion. *Journal of Field Robotics*, 35(1):23–51, 2018.

[38] Giuseppe Loianno, Chris Brunner, Gary McGrath, and Vijay Kumar. Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu. *IEEE Robotics and Automation Letters*, 2(2):404–411, 2016.

[39] Simon Lynen, Markus W Achtelik, Stephan Weiss, Margarita Chli, and Roland Siegwart. A robust and modular multi-sensor fusion approach applied to mav navigation. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 3923–3929. IEEE, 2013.

[40] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, 2015.

[41] Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalent control of lqr is efficient. *arXiv preprint arXiv:1902.07826*, 2019.

[42] Nikolai Matni, Yuh-Shyang Wang, and James Anderson. Scalable system level synthesis for virtually localizable systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3473–3480. IEEE, 2017.

[43] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *CoRR*, abs/1610.06475, 2016. URL `http://arxiv.org/abs/1610.06475`.

[44] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot. Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4029–4036, May 2014. doi: 10.1109/ICRA.2014.6907444.

[45] Y. Ouyang, M. Gagrani, and R. Jain. Control of unknown linear systems with thompson sampling. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1198–1205, Oct 2017. doi: 10.1109/ALLERTON.2017.8262873.

[46] Samet Oymak and Necmiye Ozay. Non-asymptotic identification of lti systems from a single trajectory. *arXiv preprint arXiv:1806.05722*, 2018.

[47] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile autonomous driving using end-to-end deep imitation learning. *Proceedings of Robotics: Science and Systems. Pittsburgh, Pennsylvania*, 2018.

[48] José Pereira, Morteza Ibrahimi, and Andrea Montanari. Learning networks of stochastic differential equations. In *Advances in Neural Information Processing Systems*, pages 172–180, 2010.

[49] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[50] Anders Rantzer. Concentration bounds for single parameter adaptive control. In *2018 Annual American Control Conference (ACC)*, pages 1862–1866. IEEE, 2018.

[51] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[52] Daniel J. Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *Foundations and Trends on Machine Learning*, 11(1):1–96, July 2018. doi: 10.1561/2200000070.

[53] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

[54] Tuhin Sarkar and Alexander Rakhlin. How fast can linear dynamical systems be learned? *arXiv preprint arXiv:1812.01251*, 2018.

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

[56] Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pages 439–473, 2018.

[57] Sarah Tang, Valentin Wüest, and Vijay Kumar. Aggressive flight with suspended payloads using vision-based control. *IEEE Robotics and Automation Letters*, 3(2):1152–1159, 2018.

[58] Andrew J Taylor, Victor D Dorobantu, Hoang M Le, Yisong Yue, and Aaron D Ames. Episodic learning with control lyapunov functions for uncertain robotic systems. *arXiv preprint arXiv:1903.01577*, 2019.

[59] Anastasios Tsiamis and George J Pappas. Finite sample analysis of stochastic system identification. *arXiv preprint arXiv:1903.09122*, 2019.

[60] Kim P Wabersich and Melanie N Zeilinger. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 7130–7135. IEEE, 2018.

[61] Yuh-Shyang Wang, Nikolai Matni, and John C Doyle. A system level approach to controller synthesis. *arXiv preprint arXiv:1610.04815*, 2016.

[62] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.

[63] Sholeh Yasini and Kristiaan Pelckmans. Worst-case prediction performance analysis of the kalman filter. *IEEE Transactions on Automatic Control*, 63(6):1768–1775, 2018.

[64] Seungil You and Ather Gattami. H infinity analysis revisited. *arXiv preprint arXiv:1412.6160*, 2014.

[65] Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*, volume 40. Prentice Hall, 1996.

A Linear optimal control

This section recalls some basic concepts from linear optimal control in the partially observed setting. In particular we consider the optimal control problem

$$\begin{array}{rl} \min_{\boldsymbol{K}} & c(\boldsymbol{x}, \boldsymbol{u}) \\ \text{subjectto} & \begin{array}{l} x_{k+1} = Ax_k + Bu_k + Hw_k \\ y_k = Cx_k + e_k \\ u_k = \boldsymbol{K}(y_{0:k}), \end{array} \end{array} \tag{22}$$

for x_k the state, u_k the control input, w_k the process noise, e_k the sensor noise, $\boldsymbol{K}$ a linear-time-invariant operator, and $c(\boldsymbol{x}, \boldsymbol{u})$ a suitable cost function.

By modeling the disturbance w and sensor noise e as being drawn from different signal spaces, and by choosing correspondingly suitable cost functions, we can incorporate practical performance, safety, and robustness considerations into the design process. In Table 1, we show several common cost functions that arise from different system desiderata and different classes of disturbances and measurement errors $\nu := (w, e)$. We recall the definition of the power-norm:[3]

$$\|\boldsymbol{x}\|_{pow} := \sqrt{\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T} x_k^2}.$$

The connection between the power-norm and $\mathcal{H}_\infty$ control is well studied (see [64] and references therein).

From Table 1, it is clear that the triple $(\|x_k\|_\infty, \|\boldsymbol{x}\|_\infty, \|\boldsymbol{\Phi}\|_{\mathcal{L}_1})$ satisfies the norm conditions of Section 1.2 with $\alpha = 1$. Further, we have that if $\|x_k\|_2 \leq \|y_k\|_2 + \|z_k\|_2$, then

$$\|\boldsymbol{x}\|_{pow}^2 \leq \lim_{T\to\infty} \sum_{t=0}^{\infty} (\|y_k\|_2 + \|z_k\|_2)^2 \leq 2 \lim_{T\to\infty} \sum_{t=0}^{\infty} (\|y_k\|_2^2 + \|z_k\|_2^2) = 2(\|\boldsymbol{y}\|_{pow}^2 + \|\boldsymbol{z}\|_{pow}^2).$$

Therefore the triple $(\|x_k\|_2, \|\boldsymbol{x}\|_{pow}, \|\boldsymbol{\Phi}\|_{\mathcal{H}_\infty})$ satisfies the norm conditions of Section 1.2 with $\alpha = \sqrt{2}$.

We now recall some familiar examples of cost functions and system dynamics.

Example 2 (Linear Quadratic Regulator)**.** Suppose that the cost function is given by

$$c(\boldsymbol{x}, \boldsymbol{u}) = \mathbb{E}_\nu \left[\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T} x_k^\top Q x_k + u_k^\top R u_k \right],$$

for some user-specified positive definite matrices Q and R, $w_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$, $H = I$, and that the controller is given full information about the system, i.e., that $C = I$ and $e_k = 0$ such that the measurement model collapses to $y_k = x_k$. Then the optimal control problem reduces to the familiar Linear Quadratic Regulator (LQR) problem

$$\begin{array}{rl} \text{minimize}_{\{\pi\}} & \mathbb{E}_w \left[\lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T} x_k^\top Q x_k + u_k^\top R u_k \right] \\ \text{subjectto} & \begin{array}{l} x_{k+1} = Ax_k + Bu_k + w_k \\ u_k = \pi(x_{0:k}), \end{array} \end{array} \tag{23}$$

For stabilizable (A, B), and detectable (A, Q), this problem has a closed-form stabilizing controller based on the solution of the discrete algebraic Riccati equation (DARE) [65]. This optimal control policy is linear, and given by

$$u_k^{\mathrm{LQR}} = -(B^\top P B + R)^{-1} B^\top P A x_k =: K_{\mathrm{LQR}} x_k, \tag{24}$$

where P is the positive-definite solution to the DARE defined by (A, B, Q, R).

[3] The power-norm is a semi-norm on ℓ_∞, as $\|\boldsymbol{x}\|_{pow} = 0$ for all $\boldsymbol{x} \in \ell_2$, and consequently is a norm on the quotient space ℓ_∞/ℓ_2 – this subtlety does not affect our analysis.

Name	Disturbance class	Cost function	Use cases
LQR/$\mathcal{H}_2$	$\mathbb{E}\nu = 0$, $\mathbb{E}\nu^4 < \infty$, ν_k i.i.d.	$\mathbb{E}_\nu \left[\lim_{T\to\infty} \sum_{k=0}^{T} \frac{1}{T} x_k^\top Q x_k + u_k^\top R u_k \right]$	Sensor noise, aggregate behavior, natural processes
$\mathcal{H}_\infty$	$\|\nu\|_{pow} \leq 1$	$\sup_{\|\nu\|_{pow}\leq 1} \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T} x_k^\top Q x_k + u_k^\top R u_k$	Modeling error, energy/power constraints
$\mathcal{L}_1$	$\|\nu\|_\infty \leq 1$	$\sup_{\|\nu\|_\infty \leq 1, k\geq 0} \left\| \begin{matrix} Q^{1/2} x_k \\ R^{1/2} u_k \end{matrix} \right\|_\infty$	Real-time safety constraints, actuator saturation/limits

Table 1: Different noise model classes induce different cost functions, and can be used to model different phenomenon, or combinations thereof. See [20, 65] for more details.

Example 3 (Linear Quadratic Gaussian Control)**.** Suppose that we have the same setup as the previous example, but that now the measurement is instead given by (4) for some C such that the pair (A, C) is detectable, and that $e_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$. Then the optimal control problem reduces to the Linear Quadratic Gaussian (LQG) control problem, the solution to which is:

$$u_k^{\text{LQG}} = K_{\text{LQR}} \hat{x}_k, \tag{25}$$

where $\hat{x}_k$ is the Kalman filter estimate of the state at time k. The steady state update rule for the state estimate is given by

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L_{\text{LQG}}(y_{k+1} - C(A\widehat{x}_k + Bu_k))$$

for filter gain $L_{\text{LQG}} = -PC^\top(CPC^\top + I)^{-1}$ where P is the solution to the DARE defined by $(A^\top, C^\top, I, I)$. This optimal output feedback controller satisfies the *separation principle*, meaning that the optimal controller K_{LQR} is computed independently of the optimal estimator gain L_{LQG}.

These first two examples are widely known due to the elegance of their closed-form solutions and the simplicity of implementing the optimal controllers. However, this optimality rests on stringent assumptions about the distribution of the disturbance and the measurement noise. We now turn to an example for which disturbances are adversarial and the separation principle fails.

Consider a waypoint tracking problem where it is known that both the distances between waypoints and sensor errors are instantaneously ℓ_∞ bounded, and we want to ensure that the system remains within a bounded distance of the waypoints. In this setup, the $\mathcal{L}_1$ optimal control problem is most natural, and our cost function is then

$$c(\boldsymbol{x}, \boldsymbol{u}) = \sup_{\substack{\|r_{k+1} - r_k\|_\infty \leq 1, \\ \|e_k\|_\infty \leq 1, k \geq 0}} \left\| \begin{matrix} Q^{1/2}(x_k - r_k) \\ R^{1/2} u_k \end{matrix} \right\|_\infty ,$$

for some user-specified positive definite matrices $Q = \text{diag}\frac{1}{q_i^2}$ and $R = \text{diag}\frac{1}{r_i^2}$. Then if the optimal cost is less than 1, we can guarantee that $|x_{i,k} - r_{i,k}| \leq q_i$ and $|u_{i,k}| \leq r_i$ for all possible realizations of the waypoint and sensor error processes. Considering the one-step lookahead case,[4] we can define the augmented state $\xi_k = [x_k - r_k; r_k]$ and pose the problem with bounded disturbances $w_k = r_{k+1} - r_k$. We can then formulate the following $\mathcal{L}_1$ optimal control problem

$$\begin{array}{rl} \text{minimize}_{\{\pi\}} & \sup_{\|\nu\|_\infty \leq 1, k\geq 0} \left\| \begin{matrix} \bar{Q}^{1/2} \xi_k \\ R^{1/2} u_k \end{matrix} \right\|_\infty \\ \text{subject to} & \xi_{k+1} = \bar{A}\xi_k + \bar{B}u_k + \bar{H}w_k \,, \;\; y_k = \bar{C}\xi_k + \eta_k \\ & u_k = \pi(y_{0:k}), \end{array} \tag{26}$$

where

$$\bar{A} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}, \;\; \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \;\; \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix}, \;\; \bar{H} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

This optimal control problem is an instance of $\mathcal{L}_1$ robust control [20]. The optimal controller does not obey the separation principle, and as such, there is no clear notion of an estimated state.

[4] A similar formulation exists for any T-step lookahead of the reference trajectory.

B System-level parametrization

In this section, we motivation and introduce a system level parametrization of closed-loop output feedback systems. As an illustrative example, consider the the static feedback law $u_k = K\hat{x}_k$ applied to the linear system (1), where $\hat{x}_k$ is the output of a Kalman filter as in Example 3. By considering the extended state space $[x_k; x_k - \hat{x}_k]$, the state and control inputs can be written as (assuming that $x_0 = 0$ and that the Kalman filter has converged to steady state),

$$\begin{bmatrix} x_k \\ u_k \end{bmatrix} = \sum_{t=1}^{k} \begin{bmatrix} I & 0 \\ K & -K \end{bmatrix} \begin{bmatrix} A+BK & -BK \\ 0 & A-LC \end{bmatrix}^t \begin{bmatrix} I & 0 \\ 0 & -L \end{bmatrix} \begin{bmatrix} Hw_{k-t} \\ e_{k-t} \end{bmatrix}$$

Thus it is clear that cost functions convex in x_k and u_k are non-convex in K and L. As an alternative, we can parametrize the problem in terms of the convolution of the process and sensor noise with the closed-loop system responses,

$$\begin{bmatrix} x_k \\ u_k \end{bmatrix} = \sum_{t=1}^{k} \begin{bmatrix} \Phi_{xw}(t) & \Phi_{x\eta}(t) \\ \Phi_{uw}(t) & \Phi_{u\eta}(t) \end{bmatrix} \begin{bmatrix} Hw_{k-t} \\ e_{k-t} \end{bmatrix} . \tag{27}$$

We note that the expression above is more general than the Kalman filtering example. In particular, it is valid for any linear dynamic controller, i.e. any controller which is a linear function of the system output and its history. As the equation (27) is linear in the system response elements Φ, convex constraints on state and input translate to convex constraints on the system response elements. The *system level synthesis* (SLS) framework shows that for any elements $\{\Phi_{xw}(k), \Phi_{x\eta}(k), \Phi_{uw}(k), \Phi_{u\eta}(k)\}$ constrained to obey, for all $k \geq 1$,

$$\Phi_{xw}(1) = I, \quad [\Phi_{xw}(k+1) \quad \Phi_{x\eta}(k+1)] = A\,[\Phi_{xw}(k) \quad \Phi_{x\eta}(k)] + B\,[\Phi_{uw}(k) \quad \Phi_{u\eta}(k)]\,,$$
$$\begin{bmatrix} \Phi_{xw}(k+1) \\ \Phi_{uw}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi_{xw}(k) \\ \Phi_{uw}(k) \end{bmatrix} A + \begin{bmatrix} \Phi_{x\eta}(k+1) \\ \Phi_{u\eta}(k+1) \end{bmatrix} C\,,$$

there exists a feedback controller that achieves the desired system responses (27). That this parameterization is equivalent is shown in Wang et al. [61], and therefore any optimal control problem over linear systems can be cast as a corresponding optimization problem over system response elements.

To lessen the notational burden of working with these system response elements, we will work with their z transforms, $\boldsymbol{\Phi}(z) = \sum_{k=1}^{\infty} \Phi(k) z^{-k}$ and slightly overloading notation $\boldsymbol{x} = \sum_{k=1}^{\infty} x_k z^{-k}$. This is a particularly useful to keep track of semi-infinite sequences, especially since convolutions in time can now be represented as multiplications, i.e.

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi_{xx}} & \boldsymbol{\Phi_{xy}} \\ \boldsymbol{\Phi_{ux}} & \boldsymbol{\Phi_{uy}} \end{bmatrix} \begin{bmatrix} H\boldsymbol{w} \\ \boldsymbol{e} \end{bmatrix} .$$

The affine *realizability constraints* can be rewritten as

$$[zI - A \quad -B] \begin{bmatrix} \boldsymbol{\Phi_{xx}} & \boldsymbol{\Phi_{xy}} \\ \boldsymbol{\Phi_{ux}} & \boldsymbol{\Phi_{uy}} \end{bmatrix} = [I \quad 0]\,, \quad \begin{bmatrix} \boldsymbol{\Phi_{xx}} & \boldsymbol{\Phi_{xy}} \\ \boldsymbol{\Phi_{ux}} & \boldsymbol{\Phi_{uy}} \end{bmatrix} \begin{bmatrix} zI - A \\ -C \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} , \tag{28}$$

and the corresponding control law $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{y}$ is given by $\boldsymbol{K} = \boldsymbol{\Phi_{uy}} - \boldsymbol{\Phi_{ux}}\boldsymbol{\Phi_{xx}}^{-1}\boldsymbol{\Phi_{xy}}$. This controller can be implemented via a state-space realization as in [8] or as an interconnection of the system response elements $\{\boldsymbol{\Phi_{xx}}, \boldsymbol{\Phi_{xy}}, \boldsymbol{\Phi_{ux}}, \boldsymbol{\Phi_{uy}}\}$, as shown in [61].

In this SLS framework, many control costs (as in Table 1) can be written as system norms, with

$$c(\boldsymbol{x}, \boldsymbol{u}) = \left\| \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi_{xx}} & \boldsymbol{\Phi_{xy}} \\ \boldsymbol{\Phi_{ux}} & \boldsymbol{\Phi_{uy}} \end{bmatrix} \begin{bmatrix} \varepsilon_w H \\ \varepsilon_e I \end{bmatrix} \right\|, \tag{29}$$

where ε_w and ε_e respectively bound the norms of $\boldsymbol{w}$ and $\boldsymbol{e}$.

B.1 Robust SLS

Proposition 6 (Robust Equivalence from [14]). *Suppose that designed system responses satisfy*

$$[zI - A \quad -B] \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \\ \widehat{\boldsymbol{\Phi}}_{\boldsymbol{ux}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{uy}} \end{bmatrix} = [I \quad 0]\,, \quad \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \\ \widehat{\boldsymbol{\Phi}}_{\boldsymbol{ux}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{uy}} \end{bmatrix} \begin{bmatrix} zI - A \\ -C \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} . \tag{30}$$

Then let $\mathbf{\Delta}_C$ denote the z transform of the time-varying perturbation $\Delta_{C,k}$ and define

$$\mathbf{\Delta}_1 = \widehat{\mathbf{\Phi}}_{\boldsymbol{xy}} * \mathbf{\Delta}_C, \quad \mathbf{\Delta}_2 = \widehat{\mathbf{\Phi}}_{\boldsymbol{uy}} * \mathbf{\Delta}_C \,.$$

Assume that $(1+\mathbf{\Delta}_1)^{-1}$ exists and is in $\mathcal{RH}_\infty$. Then the resulting controller is stabilizing when applied to the true system defined by matrices $(A, B, C + \Delta_{C,k})$, and achieves the system responses

$$\begin{aligned}
\mathbf{\Phi}_{\boldsymbol{xx}} &= (1+\mathbf{\Delta}_1)^{-1}\widehat{\mathbf{\Phi}}_{\boldsymbol{xx}}, & \mathbf{\Phi}_{\boldsymbol{xy}} &= (1+\mathbf{\Delta}_1)^{-1}\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}},\\
\mathbf{\Phi}_{\boldsymbol{ux}} &= \widehat{\mathbf{\Phi}}_{\boldsymbol{ux}} - \mathbf{\Delta}_2(1+\mathbf{\Delta}_1)^{-1}\widehat{\mathbf{\Phi}}_{\boldsymbol{xx}}, & \mathbf{\Phi}_{\boldsymbol{uy}} &= \widehat{\mathbf{\Phi}}_{\boldsymbol{uy}} - \mathbf{\Delta}_2(1+\mathbf{\Delta}_1)^{-1}\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}} \,.
\end{aligned}$$

By the small gain theorem, it then follows that it is sufficient to enforce that

$$\|\widehat{\mathbf{\Phi}}_{\boldsymbol{xy}}\| < \frac{1}{\varepsilon_C} \tag{31}$$

to guarantee robust stability. Note that this condition is immediate because we assume that $\|\cdot\|$ is an induced norm, and $\mathbf{\Delta}_C$ is a memoryless operator: hence the $\Delta_{C,k}$ terms can at most amplify inputs by ε_C.

A comment on finite-dimensional realizations Although the constraints (28) and objective function (29) are in fact infinite dimensional, two finite-dimensional approximations have been successfully applied in the literature. The first consists of selecting an approximation horizon T, and enforcing that $\mathbf{\Phi}(T) = 0$ for some appropriately large T, which is always possible for systems that are controllable and observable. When this is not possible, one can instead enforce bounds on the norm of $\mathbf{\Phi}(T)$ and use robustness arguments similar to those in Proposition 6 to show that the sub-optimality gap incurred by this finite dimensional approximation decays exponentially in the approximation horizon T – see [9, 14, 21, 42] for more details. Finally, in the interest of clarity, we always present the infinite horizon version of the optimization problems, with the understanding that in practice, a finite horizon approximation will need to be used.

C Proofs of intermediate results

Proposition 7. *For any (x, z), the following statements are equivalent.*

1. *There exists some $\|\Delta_C\| \le \varepsilon_C$ and $\|\eta\| \le \varepsilon_\eta$ such that $p(z) - Cx = \Delta_C x + \eta$.*
2. *$\|p(z) - Cx\| \le \varepsilon_C\|x\| + \varepsilon_\eta$.*

Proof. That (1) $\implies$ (2) follows immediately from triangle inequality and the definition of the operator norm:

$$\|p(z) - Cx\| \le \|\Delta_C\|\|x\| + \|\eta\| \,.$$

We then show that (2) $\implies$ (1) by construction. Let

$$\Delta_C = \varepsilon_C \frac{(p(z) - Cx)x^\top}{\|p(z) - Cx\|\|x\|}, \;\; \eta = p(z) - Cx - \Delta_C x \,.$$

By the definition of η, these solutions clearly satisfy the equality condition. Further, we have $\|\Delta_C\| = \varepsilon_C$. Thus, it remains only to check the norm bound on η. We see that

$$\eta = p(z) - Cx - \Delta_C x = (p(z) - Cx)\left(\frac{\|p(z) - Cx\| - \varepsilon_C\|x\|}{\|p(z) - Cx\|}\right)$$

so $\|\eta\| = \|p(z) - Cx\| - \varepsilon_C\|x\|$. Using the condition (2), we see that $\|\eta\| \le \varepsilon_\eta$. □

Proof of Lemma 1 Fix an $x_d \in \mathcal{X}_d$, and notice that for $z_k = q(x_k) + \Delta_{q,k}(x_k) + v_k$, $Cx_k = p^\star(q(x_k))$, $z_d = q(x_d) + \Delta_{q,d}(x_d) + v_d$, and $Cx_d = p^\star(q(x_d))$ we have

$$\begin{aligned}
\|p(z_k) - Cx_k\| &\le \|p(q(x_k) + \Delta_{q,k}(x_k) + v_k) - p^\star(q(x_k)) - \\
&\quad [p(q(x_d) + \Delta_{q,d}(x_d) + v_d) - p^\star(q(x_d))]\,\| + \|p(z_d) - Cx_d\| \\
&\le \|p(q(x_k)) - p^\star(q(x_k)) + p(q(x_k) + \Delta_{q,k}(x_k) + v_k) - p(q(x_k) + v_k) - \\
&\quad [p(q(x_d)) - p^\star(q(x_d)) + p(q(x_d) + \Delta_{q,d}(x_d) + v_d) - p(q(x_d) + v_d)]\,\| + \\
&\quad \|p(q(x_k) + v_k) - p(q(x_k))\| + \|p(q_d(x_d) + v_d) - p(q_d(x_d))\| + \|p(z_d) - Cx_d\| \\
&\le L_\Delta\|x - x_d\| + 2L_p\varepsilon_v + \varepsilon_C\|x_d\| + \varepsilon_\eta,
\end{aligned}$$

where the first and second inequalities follow from the triangle inequality, and the final inequality follows from the assumed Lipschitz properties of the map (6) and the learned perception map p, the assumed norm bound on the nuisance variables v, and the fact that on the training data $\|p(z_d) - Cx_d\| \leq \varepsilon_C \|x_d\| + \varepsilon_\eta$.

It then follows immediately that

$$\|p(z) - Cx\| - \varepsilon_C \|x\| - \varepsilon_\eta \leq L_\Delta \|x - x_d\| + 2L_p \varepsilon_v + \varepsilon_C(\|x_d\| - \|x\|),$$

from which the result follows by applying the reverse triangle inequality to bound $\|x_d\| - \|x\| \leq \|x - x_d\|$.

Proof of Lemma 2 Notice that over the course of a trajectory, we have system outputs

$$\boldsymbol{y} = p(\boldsymbol{z}) = C\boldsymbol{x} + (p(\boldsymbol{z}) - C\boldsymbol{x}) = (C + \boldsymbol{\Delta}_C)\boldsymbol{x} + (\boldsymbol{\eta} + \boldsymbol{\delta}). \tag{32}$$

Based on this observation, we then have by Proposition 6 that

$$\boldsymbol{x} \quad = \quad (I + \boldsymbol{\Delta}_1)^{-1} \left(\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H \boldsymbol{w} + \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} (\boldsymbol{\eta} + \boldsymbol{\delta}) \right), \tag{33}$$

for $\boldsymbol{\Delta}_1 = \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} * \boldsymbol{\Delta}_C$. We note that due to the norm assumptions in Section 1.2 and the structure of the operator $\boldsymbol{\Delta}_C$, we have that $\|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} * \boldsymbol{\Delta}_C\| \leq \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \|\Delta_C\| \leq \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \varepsilon_C < 1$.

Then for any $\boldsymbol{x}_d$ as defined in the lemma statement, it holds that

$$\begin{aligned}
\|\boldsymbol{x} - \boldsymbol{x}_d\| &= \|(I + \boldsymbol{\Delta}_1)^{-1} \left(\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H \boldsymbol{w} + \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} (\boldsymbol{\eta}) \right) - \boldsymbol{x}_d + (I + \boldsymbol{\Delta}_1)^{-1} \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \boldsymbol{\delta}\| \\
&\leq \frac{1}{1 - \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|} \left(\|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H \boldsymbol{w} + \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \boldsymbol{\eta} - \boldsymbol{x}_d\| + \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \|\boldsymbol{x}_d\| + \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \|\boldsymbol{\delta}\| \right) \\
&\leq \frac{G_x + \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \|\boldsymbol{x}_d\| + \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \|\boldsymbol{\delta}\|}{1 - \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|}
\end{aligned}$$

where the first inequality follows from the sub-multiplicative property of the norm, the triangle inequality, and robustness condition (13) allowing us to write $(I + \boldsymbol{\Delta}_1)^{-1} = \sum_{n=0}^{\infty} (-\Delta_1)^n$. The final inequality follows from the definition of G_x.

Proof of Theorem 3 Let $\delta_k := p(z_k) - Cx_k - \Delta_{C,k} x_k - \eta_k$, for $\|\Delta_{C,k}\| \leq \varepsilon_C$, $\|\eta_k\| \leq \varepsilon_\eta$ chosen to minimize $\|\delta_k\|$. Then by Lemma 1, we have that $\|\delta_k\| \leq 2L_p \varepsilon_v + (L_\Delta + \varepsilon_C)\|x_k - x_{d,k}\|$. We then immediately have that

$$\|\boldsymbol{\delta}\| \leq 2\alpha L_p \varepsilon_v + \alpha(L_\Delta + \varepsilon_C)\|\boldsymbol{x} - \boldsymbol{x}_d\|,$$

by the α-element wise compatibility of the norm.

From Lemma 2, we can then write

$$\begin{aligned}
\|\boldsymbol{x} - \boldsymbol{x}_d\| &\leq \frac{G_x + \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \|\boldsymbol{x}_d\| + \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \|\boldsymbol{\delta}\|}{1 - \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|} \\
&\leq \frac{G_x + \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \|\boldsymbol{x}_d\|}{1 - \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|} + \frac{\|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\| \left(2\alpha L_p \varepsilon_v + \alpha(L_\Delta + \varepsilon_C)\|\boldsymbol{x} - \boldsymbol{x}_d\|\right)}{1 - \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|}.
\end{aligned}$$

Rearranging gives bound (17). Bound (18) is obtained in a similar fashion.

Proof of Proposition 4 Define $\boldsymbol{x} := \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H \boldsymbol{w} + \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} \eta$, and

$$x_{R,k} := \underset{\|x_{R,k}\|_\infty \leq R}{\operatorname{argmin}} \|x_k - x_{R,k}\|_\infty.$$

Then

$$\begin{aligned}
\|\boldsymbol{x} - \boldsymbol{x}_d\| &= \sup_k \|x_k - x_{d,k}\|_\infty \leq \sup_k \|x_k - x_{R,k}\|_\infty + \|x_{R,k} - x_d\|_\infty \\
&\leq \left[\sup_k \|x_k\|_\infty - R \right]_+ + \varepsilon_d \leq \left[\|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H \boldsymbol{w}\|_{\mathcal{L}_1} + \varepsilon_\eta \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|_{\mathcal{L}_1} - R \right]_+ + \varepsilon_d,
\end{aligned}$$

where the first equality follows from the definition of the ℓ_∞ norm, the first inequality follows from the triangle inequality, the second from the definition of $x_{R,k}$, and the third from the triangle inequality, and that $\sup_k \|x_k\|_\infty = \|\boldsymbol{x}\|_\infty$.

Proof of Proposition 5 Choosing $\boldsymbol{x}_d = \boldsymbol{x}^{(s)} = \{x^{(s)}_{0:T-1}, x^{(s)}_{0:T-1}, \dots\} = \boldsymbol{\Phi_{xx}} H \boldsymbol{w}^{(s)}$, we have

$$\begin{aligned}\|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H\boldsymbol{w} + \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\boldsymbol{\eta} - \boldsymbol{x}_d\| &= \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H\boldsymbol{w} + \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\eta - \boldsymbol{\Phi}_{\boldsymbol{xx}} H\boldsymbol{w}^{(s)}\| \\ &= \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H(\boldsymbol{w} - \boldsymbol{w}^{(s)}) + (\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H - \boldsymbol{\Phi}_{\boldsymbol{xx}} H)\boldsymbol{w}^{(s)} + \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}}\eta\| \\ &\le \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H\|\varepsilon_r + \varepsilon_{\boldsymbol{K}}\varepsilon_w\alpha + \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\eta\|\end{aligned}$$

D Performance guarantees

In this section, we show how the perception errors and generalization errors affect the closed-loop performance, and use this insight to propose a robust control problem.

Proposition 8. *Let the assumptions of Theorem 3, and either Proposition 4 or 5 hold. Further assume that the control cost $c(\boldsymbol{x}, \boldsymbol{u})$ is defined by an induced norm as in* (29)*. Then the performance of the controller $\widehat{\boldsymbol{K}}$ defined by the system responses $\{\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}}, \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}, \widehat{\boldsymbol{\Phi}}_{\boldsymbol{ux}}, \widehat{\boldsymbol{\Phi}}_{\boldsymbol{uy}}\}$ satisfying the SLS constraints* (28) *defined by the matrices (A, B, C) achieves performance bounded by:*

$$c(\boldsymbol{x}, \boldsymbol{u}) \le \left\| \begin{bmatrix} Q^{1/2}\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} \\ R^{1/2}\widehat{\boldsymbol{\Phi}}_{\boldsymbol{ux}} \end{bmatrix} H \right\| + \left(\varepsilon_\eta + \varepsilon_G + \frac{\varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H + \varepsilon_\eta \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|}{1 - \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|} \right) \left\| \begin{bmatrix} Q^{1/2}\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \\ R^{1/2}\widehat{\boldsymbol{\Phi}}_{\boldsymbol{uy}} \end{bmatrix} \right\| .$$

For ε_G specified by the right-hand-side of bound (18)*, and G_x set as in either Proposition 4 or 5.*

Then notice that the first term in this bound is the cost achieved by a system with perfect output measurement. The second term is the additional cost incurred due to the imperfections in the sensing model due to the perception map and the generalization error. We also remark that an analogous result holds for the $\mathcal{H}_2$ cost, where the main subtlety comes from the fact that it is not sub-multiplicative with itself. Instead, the coefficients to ε_C in the expression are replaced with a $\mathcal{H}_\infty$ norm, which is the operator norm analog to $\mathcal{H}_2$.

Theorem 8 therefore suggests the following robust synthesis procedure:

$$\min_{\boldsymbol{\Phi}, \tau, \gamma} \left\| \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{xx}} & \boldsymbol{\Phi}_{\boldsymbol{xy}} \\ \boldsymbol{\Phi}_{\boldsymbol{ux}} & \boldsymbol{\Phi}_{\boldsymbol{uy}} \end{bmatrix} \begin{bmatrix} H \\ (\varepsilon_\eta + \varepsilon_G) I \end{bmatrix} \right\| + \frac{\varepsilon_C(\gamma + (\varepsilon_\eta + \varepsilon_G)\tau)}{1 - \varepsilon_C \tau} \left\| \begin{bmatrix} Q^{1/2}\boldsymbol{\Phi}_{\boldsymbol{xy}} \\ R^{1/2}\boldsymbol{\Phi}_{\boldsymbol{uy}} \end{bmatrix} \right\|$$

subject to (28), $\|\boldsymbol{\Phi}_{\boldsymbol{xy}}\| \le \tau$, $\|\boldsymbol{\Phi}_{\boldsymbol{xx}} H\| \le \gamma$, $\tau < \dfrac{1}{\varepsilon_C + \alpha(L_\Delta + \varepsilon_C)}$,

$$\varepsilon_G = \frac{G_x + 2\alpha L_p \varepsilon_v + \varepsilon_C \tau(\|\boldsymbol{x}_d\| - 2\alpha L_p \varepsilon_v)}{1 - \tau(\varepsilon_C + \alpha(L_\Delta + \varepsilon_C))}.$$

Where in the dense sampling setting,

$$G_x = [\gamma + \tau\varepsilon_\eta - R]_+ + \varepsilon_d,$$

or in the imitation learning case we add the constraint $\|\boldsymbol{\Phi}_{\boldsymbol{xx}} H - \boldsymbol{\Phi}^d_{\boldsymbol{xx}} H\| \le \rho$ and set

$$G_x = \gamma\varepsilon_r + \rho\alpha + \tau\varepsilon_\eta \; .$$

This procedure is a convex program for fixed (γ, τ), so the full problem can then be approximately solved by gridding over (γ, τ). In the imitation learning setting, we may additionally grid over ρ.

Proof of 8. First, we simplify the expression for the actual system response in terms of the designed variables using the result of Proposition 6

$$\begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{xx}} & \boldsymbol{\Phi}_{\boldsymbol{xy}} \\ \boldsymbol{\Phi}_{\boldsymbol{ux}} & \boldsymbol{\Phi}_{\boldsymbol{uy}} \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \\ \widehat{\boldsymbol{\Phi}}_{\boldsymbol{ux}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{uy}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Delta}_1 \\ \boldsymbol{\Delta}_2 \end{bmatrix} (I + \boldsymbol{\Delta}_1)^{-1} \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \end{bmatrix}$$

Plugging this expression into the cost $c(\boldsymbol{x}, \boldsymbol{u})$ as in (29) and applying triangle inequality yields the upper bound (using the shorthand $(\varepsilon_\eta + \varepsilon_G) I = H_y$)

$$\begin{aligned}&\left\| \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \\ \widehat{\boldsymbol{\Phi}}_{\boldsymbol{ux}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{uy}} \end{bmatrix} \begin{bmatrix} H \\ H_y \end{bmatrix} \right\| + \left\| \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}_1 \\ \boldsymbol{\Delta}_2 \end{bmatrix} (I + \boldsymbol{\Delta}_1)^{-1} \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \end{bmatrix} \begin{bmatrix} H \\ H_y \end{bmatrix} \right\| \\ &\quad\le \left\| \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \\ \widehat{\boldsymbol{\Phi}}_{\boldsymbol{ux}} & \widehat{\boldsymbol{\Phi}}_{\boldsymbol{uy}} \end{bmatrix} \begin{bmatrix} H \\ H_y \end{bmatrix} \right\| + \frac{\varepsilon_C}{1 + \varepsilon_C \|\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}}\|} \left\| \begin{bmatrix} Q^{1/2}\widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} \\ R^{1/2}\widehat{\boldsymbol{\Phi}}_{\boldsymbol{uy}} \end{bmatrix} \right\| \left\| \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xx}} H + \widehat{\boldsymbol{\Phi}}_{\boldsymbol{xy}} H_y \right\|\end{aligned}$$

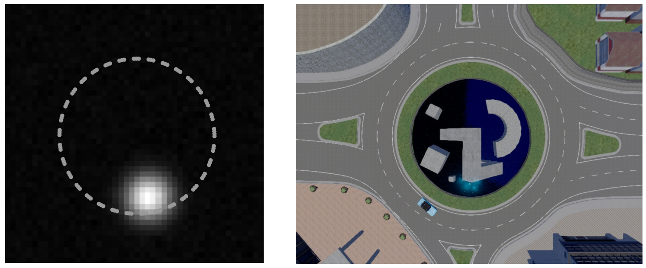

Figure 4: The nominal trajectories that the synthetic (left) and simulated vehicle (right) are trained to follow.

Where the second line uses the sub-multiplicative property of the norm and the robustness condition (13). The desired expression follows from plugging in $H_y = (\varepsilon_\eta + \varepsilon_G)I$, a final triangle inequality, and rearranging terms. $\square$

E Experimental details

E.1 Training

In both the numerically simulated blurry circle and CARLA vehicle reference tracking examples, we drive our system to follow a circle of radius 1. The circular tracks are shown in Figure 4. We first generate training $\{x_{0:T}^{(s)}, z_{0:T}^{(s)}\}_{s=1}^{200}$ and validation $\{x_{0:T}^{(v)}, z_{0:T}^{(v)}\}_{v=1}^{200}$ trajectories by driving the system with an optimal state feedback controller (i.e. where measurement $\boldsymbol{y} = \boldsymbol{x}$) to track a desired reference trajectory $\boldsymbol{w}^{(s)} = \boldsymbol{r} + \boldsymbol{v}^{(s)}$, where $\boldsymbol{r}$ is a nominal reference, and $\boldsymbol{v}^{(s)}$ is a random norm bounded random perturbation satisfying $\|v_k^{(s)}\|_\infty \leq 0.1$. We choose the nominal reference $\boldsymbol{r}$ to be a sequence of waypoints to take the circle/vehicle around the circular tracks in Figure 4.

For the synthetic example, we jointly learn a linear perception map $p(x) = Px$ and error model using optimization problem (9) using regularization $R(p) = \|P\|_\infty$. We use an additional validation set for cross validating the regularization parameter λ.

In CARLA experiments, we use ORB SLAM 2 [43] as a black box perception map (see Figure 5) that gives position estimates of the vehicle. As such, we directly fit the training data to our error model using optimization problem (8). Then, using the learned error model $(\varepsilon_C, \varepsilon_\eta)$, we design an optimal controller that is additionally constrained to satisfy equation (16) with smallest feasible right-hand-side, as we do not have access to the true Lipschitz parameter L_Δ.

E.2 Controller synthesis

As mentioned above, the robust SLS procedure we propose and analyze requires solving a a finite dimensional approximation to an infinite dimensional optimization problem, as $\{\boldsymbol{\Phi}_{\boldsymbol{xx}}, \boldsymbol{\Phi}_{\boldsymbol{xy}}, \boldsymbol{\Phi}_{\boldsymbol{ux}}, \boldsymbol{\Phi}_{\boldsymbol{uy}}\}$ and the corresponding constraints (28) and objective function are infinite dimensional objects. As an approximation, we restrict the system responses $\{\boldsymbol{\Phi}_{\boldsymbol{xx}}, \boldsymbol{\Phi}_{\boldsymbol{xy}}, \boldsymbol{\Phi}_{\boldsymbol{ux}}, \boldsymbol{\Phi}_{\boldsymbol{uy}}\}$ to be finite impulse response (FIR) transfer matrices of length $T = 200$, i.e., we enforce that $\boldsymbol{\Phi}(T) = 0$. We then solve the resulting optimization problem with MOSEK under an academic license [10]. More explicitly, we define $\mathrm{vec}(\boldsymbol{F}) := \begin{bmatrix} F_0^\top & \dots & F_{T-1}^\top \end{bmatrix}^\top$ and $\overline{\mathrm{vec}}(\boldsymbol{F}) := [F_0 \quad \dots \quad F_{T-1}]$, where F_t are the FIR coefficients of the system responses. We further define

$$\boldsymbol{Z} := \begin{bmatrix} Q^{1/2} & \\ & R^{1/2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{xx}} & \boldsymbol{\Phi}_{\boldsymbol{xy}} \\ \boldsymbol{\Phi}_{\boldsymbol{ux}} & \boldsymbol{\Phi}_{\boldsymbol{uy}} \end{bmatrix} \begin{bmatrix} H \\ (\varepsilon_\eta + \varepsilon_G)I \end{bmatrix}. \tag{34}$$

The SLS constraints (28) and FIR condition are then enforced as

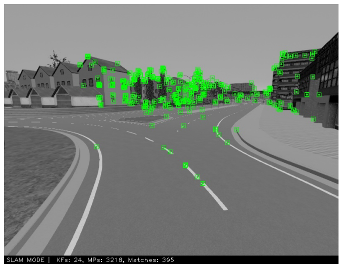

Figure 5: ORB-SLAM annotated screenshot.

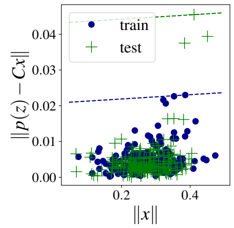

Figure 6: Error model fit in $(\|x\|_\infty, \|p(x) - Cx\|_\infty)$ space for the synthetic example disturbance rejection problem.

$$\begin{aligned}
&[\overline{\mathrm{vec}}(\boldsymbol{\Phi_{xx}}) \quad 0] - [0 \quad A\overline{\mathrm{vec}}(\boldsymbol{\Phi_{xx}})] = [0 \quad B\overline{\mathrm{vec}}(\boldsymbol{\Phi_{ux}}) + \overline{\mathrm{vec}}(\boldsymbol{I})] \\
&[\overline{\mathrm{vec}}(\boldsymbol{\Phi_{xy}}) \quad 0] - [0 \quad A\overline{\mathrm{vec}}(\boldsymbol{\Phi_{xy}})] = [0 \quad B\overline{\mathrm{vec}}(\boldsymbol{\Phi_{uy}})] \\
&\begin{bmatrix} \mathrm{vec}(\boldsymbol{\Phi_{xx}}) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \mathrm{vec}(\boldsymbol{\Phi_{xx}})A \end{bmatrix} = \begin{bmatrix} 0 \\ \mathrm{vec}(\boldsymbol{\Phi_{xy}})C + \mathrm{vec}(\boldsymbol{I}) \end{bmatrix} \\
&\begin{bmatrix} \mathrm{vec}(\boldsymbol{\Phi_{ux}}) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \mathrm{vec}(\boldsymbol{\Phi_{ux}})A \end{bmatrix} = \begin{bmatrix} 0 \\ \mathrm{vec}(\boldsymbol{\Phi_{uy}})C) \end{bmatrix} \\
&\boldsymbol{\Phi_{xx}}(T) = 0,\ \boldsymbol{\Phi_{ux}}(T) = 0,\ \boldsymbol{\Phi_{xy}}(T) = 0,\ \boldsymbol{\Phi_{uy}}(T) = 0.
\end{aligned} \tag{35}$$

We then solve the following optimization problem

$$\begin{aligned}
\underset{\boldsymbol{\Phi}}{\text{minimize}} \quad & \mathrm{cost}(\boldsymbol{Z}) \\
\text{subject to} \quad & (35),\ \ \mathrm{norm}(\boldsymbol{\Phi_{xy}}) \leq \frac{1}{\varepsilon_C + \alpha(L_\Delta + \varepsilon_C)},
\end{aligned}$$

where the $\mathrm{cost}(\cdot)$ and $\mathrm{norm}(\cdot)$ operators are problem dependent.

For the $\mathcal{L}_1$ robust problem, both the cost function and robust norm constraint reduce to the $\ell_\infty \to \ell_\infty$ induced matrix norm for an FIR transfer response $\boldsymbol{F}$ with coefficients in $\mathbb{R}^{n\times m}$,

$$\mathrm{norm}_{\infty\to\infty}(\boldsymbol{F}) = \max_{i=1,\ldots,n} \|\overline{\mathrm{vec}}(\boldsymbol{F})_i\|_1,$$

where $\overline{\mathrm{vec}}(\boldsymbol{F})_i$ denotes the ith row of $\overline{\mathrm{vec}}(\boldsymbol{F})$.

For the robust LQG problem, the cost function reduces to the Frobenius norm for an FIR cost transfer matrix $\boldsymbol{Z}$, i.e.,

$$\mathrm{cost}(\boldsymbol{Z}) = \|\mathrm{vec}(\boldsymbol{Z})\|_F^2 = \sum_{t=0}^{T} \mathrm{Tr} Z_t^\top Z_t.$$

The corresponding robustness constraint is the $\mathcal{H}_\infty$ norm, which in addition to being defined as in Table 1, can also be defined as the $\ell_2 \to \ell_2$ induced norm. This constraint reduces to a compact semidefinite program (SDP) over the system response variables as in Theorem 5.8 of Dumitrescu [24] – this is applied in the state feedback setting in Appendix G.3 of Dean et al. [22], and the output feedback setting in Section 5.1 of Boczar et al. [14]. However, the computational complexity of the resulting SDP scales as $O(T^3)$, which limits the FIR horizon T for which a controller can be computed. To circumvent this issue, we instead implement the norm constraint via an $\ell_1 \to \ell_1$ induced matrix norm, which is equivalent to the $\ell_\infty \to \ell_\infty$ induced matrix norm as applied to the transpose system:

$$\mathrm{norm}_{1\to 1}(\boldsymbol{F}) = \max_{i=1,\ldots,n} \|\overline{\mathrm{vec}}(\boldsymbol{F}^\top)_i\|_1.$$

One can check that $\|\boldsymbol{F}\|_{\mathcal{H}_\infty} \leq \sqrt{n}\|\boldsymbol{F}^\top\|_{\mathcal{L}_1}$, and therefore we are enforcing an upper bound to the desired robustness constraint. The resulting synthesis problem is then a linearly constrained quadratic program, which in practice is much more efficient to solve.

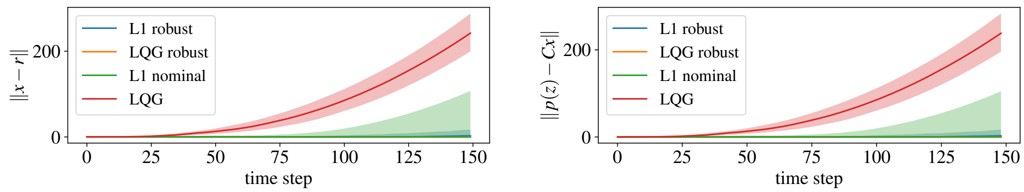

Figure 7: Median ℓ_∞ state deviation (left) and ℓ_∞ estimation error (right) over 200 rollouts for the synthetic example disturbance rejection problem. Error bars are upper and lower quartiles.

E.3 Additional experiments

In this section we demonstrate the dense sampling training scenario by studying a disturbance rejection problem with the synthetic circle example. We generate training data within an ℓ_∞ ball of the origin of radius 0.25 at a resolution of $\varepsilon_d = 0.02$ and learn a perception map. We fit a perception map and error model using optimization problem (9) – the resulting error profile $(\|x\|_\infty, \|p(x) - Cx\|_\infty)$ is shown in Figure 6.

The disturbance rejection control problem consists of keeping the circle within this ℓ_∞ bounded ball in the face of of ℓ_∞ bounded noise ($\|w_k\|_\infty \leq \Delta_w = 0.5$) – we apply a disturbance signal specified by $w_k = \Delta_w \text{sign}(Bu_k)$. We show in Figure 7 that the robustly synthesized LQG and $\mathcal{L}_1$ controllers satisfying constraint (16) maintain the system within the ℓ_∞ norm ball of radius $.25$, whereas naively synthesized controllers lead to systems that are driven far out of the densely sampled region, at which point the perception and control loop completely fails.