Tutorial: Data Driven and Learning Enabled Control

M. Sznaier F. Allgower A. C. B. de Oliveira N. Ozay E. Sontag

***Abstract*— Data-driven control (DDC), that is the design of controllers directly from observed data, has attracted substantial attention in recent years due to its advantages over model-based control. DDC avoids a computationally expensive, potentially conservative model identification step and bypasses practically difficult questions such as model order/class selection. This tutorial paper seeks to offer a sampling of the different approaches that have been recently used to synthesize data driven controllers and filters, covering both analytic approaches and learning enabled ones, indicating the relative strengths of each. A second objective is to provide a key to the rapidly expanding literature in the subject, to help researchers newly interested in this field to quickly come up to speed.**

I. Introduction

The past few years have seen a renewed interest in data-driven control (DDC), motivated both by the increased availability of data and the success of machine learning-inspired methods in finding controllers directly from data, bypassing a potentially conservative model identification step. Indeed, as shown in [1], for a not completely known or high-order system, data-driven methods can statistically outperform model-based ones in terms of control cost. Further, the work in [2] defines a notion of 'data informativity,' demonstrating that the assumptions required for a data-driven stabilization task are less restrictive than what is needed to perform system identification.

Earlier approaches to linear DDC include virtual reference feedback tuning [3], [4]), correlation-based tuning [5] and iterative feedback tuning [6]. All of these methods assume a reference signal\model of the closed-loop system and aim to minimize the error between the reference and true signal. However, these methods either lack closed-loop stability guarantees or consider simplified noise-free scenarios. [7] provides an asymptotical stability criterion but this is only valid when considering infinitely long data sequences. As an alternative, the recent popularity of neural networks motivated several data-driven methods based on adaptive dynamic programming [8], [9]), reinforcement learning [10] and Koopman eigenfunctions [11], [12]. While these methods perform well in simulations, they lack rigorous stability certificates and usually require a fair amount of work tuning the hyper-parameters.

Mario Sznaier, Arthur C. B. de Oliveira, and Eduardo Sontag are with the ECE Department, Northeastern University, Boston, MA, USA. Frank Allgower is with the Institute for Systems Theory and Automatic Control and Mechanical Engineering Department, University of Stuttgart, Germany. Necmiye Ozay is with the EECS and Robotics Departments, University of Michigan, Ann Arbor, MI, USA. A. C. B. de Oliveira, N. Ozay, E. Sontag and M. Sznaier were partially supported by ONR grant N00014-21-1-2431. M. Sznaier was also supported by NSF grant CMMI–2208182.

More recent work has concentrated on guaranteeing closed-loop stability in scenarios where only a finite data record is available. Along these lines, data-driven controllers guaranteed to stabilize the consistency set have been proposed in [13], [14] using a polyhedral control Lyapunov function (PCLF) approach. This approach has the advantage of being able to handle switching dynamics. On the other hand, this ability comes at the price of increased computational complexity, directly tied to the number of faces of the level sets of the PCLF. This difficulty can be addressed by considering quadratic rather than polyhedral Lyapunov functions. For the case of noise-free data, the milestone paper [15] proposed an approach, based on the Willems' fundamental lemma [16], that optimized an $\mathcal{H}_2$ cost and guaranteed closed-loop stability for scenarios where the noise is small enough. The paper [17] substantially advanced the state-of-the-art by exploiting a novel matrix S-lemma that leads to non-conservative controllers, with guaranteed $\mathcal{H}_2$ performance, provided that the noise satisfies a quadratic constraint. [18] considered a scenario where both uncertainty and noise are described using multipliers. Other SDP-based methods for nonconservative DDC under ℓ_2-bounded process noise include Petersen's Lemma [19], through updating uncertanties [20], and Lyapunov-Metzler (bilinear) inequalities for switched systems [21]. Alternatively, a sum-of-squares (SoS) based approach capable of handling the more involved error-in-variables (EiV) case was proposed in [22].

As an alternative to designing a single controller, several approaches follow a model-predictive approach, based on on-line optimization (see [23] for a comprehensive list of references). The paper [24] proposed a data-enabled predictive control (DeePC) algorithm, based on Willem's Lemma, to track a desired trajectory. In the case of LTI systems, this algorithm is shown to be equivalent to classical MPC. Along the same lines, [25] proposes a robust data-driven MPC method, also based on Willems' Fundamental Lemma. The approach addresses bounded additive noise in output measurements by introducing a slack variable with cost regularization in the MPC problem. The work in [26] proposes a method to compute robust backward reachable sets from noisy data for unknown constrained linear systems with bounded disturbances. The approach constructs backward reachable sets using zonotopic inner approximations that align with measurements and state and input constraints, and designs a controller based on the reachable sets. The study by [27] introduces a data-driven receding-horizon control method for chance-constrained output tracking in unknown stochastic LTI systems influenced by both process and measurement noise. The approach constructs an auxiliary

state model directly parameterized by input-output data, enabling the formulation of a stochastic control problem. This method incorporates chance constraints, aligning with a stochastic MPC framework. Finally, [28], the authors consider an unknown discrete LTI system affected by noise and design a general min-max MPC problem with input and state constraints. The considered min-max problem minimizes the worst case cost over the set of system matrices consistent with the measurement data. Alternatively, [29] presents a framework that guarantees robust performance in the presence of bounded disturbances. This approach enforces set-membership constraints and utilizes execution data to iteratively refine a set of compatible systems online.

Non-linear DDC is less developed, but a number of techiques that guarantee closed loop stability are available. Local stabilization can be achieved by estimating the Jacobian of the system around a point combined with LTI data-driven techniques [30]. Data-driven global stabilization of nonlinear systems is a much harder problem. Without assuming some form of nonlinearity the problem of obtaining stability certificates becomes intractable. To address these difficulties, [31] assumed polynomial dynamics and resorted to finding a data-driven density function so that the resulting problem was transformed into linear programs. Similarly, in [32], a positive semidefinite program is derived by using quadratic-like Lyapunov functions to ensure closed-loop stability. Recent work on data-driven safe control of continuous-time non-linear systems includes [33]–[37]. [33] introduced a learning-based approach that iteratively collects data and updates a controller corresponding to a known CBF to ensure ultimate safety. The work in [34] uses density functions and the Theorems of Alternatives to derive a rational controller together with a density-based safety certificate. The work in [37] uses contraction methods to form robust adaptive CBF under a set membership approach, but assumes that the input relation $g(\cdot)$ is known. The approach in [35] finds an ellipsoidal overapproximation of the consistency set and enforces robust invariance to certify safety. The method in [36] performs forward and backward Hamilton-Jacobi reachability analysis on a differential game function and uses data-driven Bayesian inference to construct high probability safety guarantees. Interested readers are referred to [38] and references therein for a comprehensive overview on data-driven safe control using Hamilton-Jacobi reachability, Control Barrier Functions (CBFs) and predictive control-related techniques. Finally, the related problem of finding controllers that render an unknown system contractive has been recently studied in [39], [40] for the case of polynomial dynamics.

Unlike the data-driven safe control of continuous-time non-linear systems that is relatively well understood, its discrete-time analogue is considerably less developed. The approach in [41] models the system as known dynamics plus an additive residual and synthesizes a control law that guarantees safety up to a given risk level. [42] performs formal synthesis of safety for stochastic systems with a desired confidence level, over a finite time horizon, relying on multiple measurements at each datapoint. The method in [43] performs iterative model predictive control for a discrete-time system by constraining state trajectories to always lie in a sampled safe set (using integer programming). The paper [44] proposed an SoS based approach based on a lifting to decouple the current state from the next.

Given the vast amount of work noted above, it is impossible to cover all these approaches within a single tutorial conference paper. Rather, our goal is to offer examples of the different types of techniques. With this in mind, in the rest of the paper we briefly cover examples of the following techniques:

1) Off-line data-driven controller design for linear and non-linear systems
2) Data-driven model predictive control
3) Direct optimization, neural-net based controller design
4) Large Language Models (LLM) techniques.

II. Data Driven LQR Control of LTI Systems via Duality

In this section, we briefly review a framework to design LTI controllers that solve the data-driven linear quadratic regulator (DDLQR) problems for both continuous and discrete-time systems. Given noisy input/state measurements and a few priors, the goal is to find a state feedback controller that minimizes the worst case $\mathcal{H}_2$ cost, over all systems compatible with the a-priori information and the experimental data. In principle, finding such a controller is a non-convex robust optimization problem. The main result of this section, taken from [45] shows that, by exploiting duality, the problem can be recast into a convex, albeit infinite-dimensional, functional Linear Program. In turn, this problem can be solved by solving a sequence of increasingly tight finite dimensional semi-definite relaxations, obtained using sum-of-squares and Putinar's Positivstellensatz arguments.

A. Problem Statement

Throughout this section, we consider the following controller design problem:

Problem 1. *Consider an LTI system:*

$$\begin{aligned} \delta\boldsymbol{\xi} &= \mathbf{A}\boldsymbol{\xi} + \mathbf{B}\mathbf{u} + \boldsymbol{\eta} \\ \mathbf{z} &= \begin{bmatrix} \mathbf{R}^{1/2} & 0 \\ 0 & \mathbf{Q}^{1/2} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\xi} \end{bmatrix} \end{aligned} \tag{1}$$

where $\delta\boldsymbol{\xi}(t) = \dot{\boldsymbol{\xi}}(t)$ for continuous-time and $\delta\boldsymbol{\xi}(t) = \boldsymbol{\xi}(t+1)$ for the discrete case. Here $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$ are unknown system matrices, $\boldsymbol{\xi}(t) \in \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathbb{R}^m$ denote the state and input vectors, and $\boldsymbol{\eta}(t) \in \mathbb{R}^n$ denotes ℓ_∞ bounded noise[1], with $||\boldsymbol{\eta}||_\infty \leq \epsilon$. Given measurements $\delta\boldsymbol{\xi}(t_k)$, $\boldsymbol{\xi}(t_k)$, $\mathbf{u}(t_k)$, where $k = 1, \ldots, N$ denotes the sample index, the goal is to find a state feedback controller $\mathbf{u} = \mathbf{K}\boldsymbol{\xi}(t)$ that minimizes the worst case, over all systems that could have generated the observed data, of the $\mathcal{H}_2$ cost γ from the input w to the output z.

[1]This noise arises for instance from numerical estimation of $\delta\mathbf{x}(t)$ from measurements of $\mathbf{x}(t)$

In order to solve the problem above, we recall the following well known result (see for instance [46]). Given the system (1), the $\mathcal{H}_2$ problem is equivalent to the following semi-definite optimization

$$\begin{aligned} &\text{minimize } \gamma \\ &C(\gamma, \mathbf{Y}, \mathbf{M}, \mathbf{W}) \succ 0 \end{aligned} \tag{2}$$

where $\mathbf{W} \doteq \mathbf{R}^{1/2}\mathbf{M}\mathbf{Y}^{-1}\mathbf{M}^T\mathbf{R}^{1/2}$ and $C(\cdot)$ is

$$\begin{aligned} &C_1(\gamma, \mathbf{Y}, \mathbf{M}, \mathbf{W}) \doteq -\mathbf{Tr}(\mathbf{Q}\mathbf{Y}) - \mathbf{Tr}(\mathbf{W}) + \gamma \\ &C_2(\mathbf{Y}, \mathbf{M}) \doteq \Omega(\mathbf{Y}, \mathbf{M}) + \Omega^T(\mathbf{Y}, \mathbf{M}) + \mathbf{L}(\mathbf{Y}) + \mathbf{L}_o \\ &\Omega(\mathbf{Y}, \mathbf{M}) = \mathbf{H}_L(\mathbf{A}\mathbf{Y} + \mathbf{B}\mathbf{M})\mathbf{H}_R \\ &C_3(\mathbf{Y}, \mathbf{M}, \mathbf{W}) \doteq \begin{bmatrix} \mathbf{W} & \mathbf{R}^{1/2}\mathbf{M} \\ \mathbf{M}^T\mathbf{R}^{1/2} & \mathbf{Y} \end{bmatrix} \\ &C(\gamma, \mathbf{Y}, \mathbf{M}, \mathbf{W}) \doteq \text{diag}(C_1, C_2, C_3) \end{aligned} \tag{3}$$

and:

$$-\mathbf{H}_L = \mathbf{H}_R = \mathbf{I}; \mathbf{L} = \mathbf{0}, \mathbf{L}_o = -\mathbf{I} \text{ (continuous-time)} \tag{4a}$$

or

$$\left.\begin{aligned} &\mathbf{H}_L = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}^T, \mathbf{H}_R = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &\mathbf{L}(\mathbf{Y}) = \begin{bmatrix} \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix}, \mathbf{L}_o = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \end{aligned}\right\} \text{ (discrete-time)} \tag{4b}$$

The corresponding state-feedback controller gain is $\mathbf{K} = \mathbf{M}\mathbf{Y}^{-1}$.

Definition 1. *Given N data triplets $\mathcal{D}_N \doteq \{(\delta\boldsymbol{\xi}(t_k), \boldsymbol{\xi}(t_k), \mathbf{u}(t_k))\}_{k=1}^N$, the consistency set $\mathcal{C}(\mathcal{D}_N)$ is the set of all matrices $(\mathbf{A}, \mathbf{B})$ consistent with the observed data $\mathcal{D}_N$, that is all $(\mathbf{A}, \mathbf{B})$ pairs satisfying the $2N$ polytopic constraints*

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{A} & -\mathbf{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}(t_k) \\ \mathbf{u}(t_k) \end{bmatrix} \leq \begin{bmatrix} \epsilon\mathbf{1} + \delta\boldsymbol{\xi}(t_k) \\ \epsilon\mathbf{1} - \delta\boldsymbol{\xi}(t_k) \end{bmatrix} \doteq \mathbf{d}_k, \; k = 1, \ldots, N \tag{5}$$

Note that each constraint in (5) can be rewritten in the following form:

$$\mathbf{Tr}(\mathbf{A}\mathbf{Z}_{i,k}^{\boldsymbol{\xi}} + \mathbf{B}\mathbf{Z}_{i,k}^{\mathbf{u}}) \leq d_{i,k} \tag{6}$$

where $d_{i,k}$ is the i^{th} entry of the vector $\mathbf{d}_k$. This can be accomplished by defining $4nn_s$ matrices $\mathbf{Z}_{i,k}^{\boldsymbol{\xi}}$ and $\mathbf{Z}_{i,k}^{\mathbf{u}}$ having the data vector $\boldsymbol{\xi}(t_k)(\mathbf{u}(t_k))$ as their i^{th} column and all other entries equal to zero.

$$\begin{aligned} \mathbf{Z}_{i,k}^{\boldsymbol{\xi}} &= \begin{bmatrix} \ldots, \pm\boldsymbol{\xi}(t_k), \ldots \end{bmatrix}_{n\times n} \\ \mathbf{Z}_{i,k}^{\mathbf{u}} &= \begin{bmatrix} \ldots, \pm\mathbf{u}(t_k), \ldots \end{bmatrix}_{m\times n} \end{aligned} \tag{7}$$

For example, for a second-order system there are 8 matrices per sample. In this case we have:

$$\begin{aligned} &\mathbf{Z}_{1,k}^{\boldsymbol{\xi}} = \begin{bmatrix} \xi_1(t_k) & 0 \\ \xi_2(t_k) & 0 \end{bmatrix}, \quad \mathbf{Z}_{2,k}^{\boldsymbol{\xi}} = \begin{bmatrix} 0 & \xi_1(t_k) \\ 0 & \xi_2(t_k) \end{bmatrix}, \\ &\mathbf{Z}_{3,k}^{\boldsymbol{\xi}} = \begin{bmatrix} -\xi_1(t_k) & 0 \\ -\xi_2(t_k) & 0 \end{bmatrix}, \mathbf{Z}_{4,k}^{\boldsymbol{\xi}} = \begin{bmatrix} 0 & -\xi_1(t_k) \\ 0 & -\xi_2(t_k) \end{bmatrix} \end{aligned} \tag{8}$$

with similar expressions for $\mathbf{Z}_{i,k}^{\mathbf{u}}$. In this context, the DDLQR problem can be rewritten as:

Problem 2.

$$\begin{aligned} &\min_{\mathbf{Y}\succ 0, \mathbf{M}, \mathbf{W}} \left\{ \max_{\mathbf{A}, \mathbf{B}} \gamma \right\} \textit{ subject to (3) and} \\ &d_{i,k} - \mathbf{Tr}(\mathbf{A}\mathbf{Z}_{i,k}^{x} + \mathbf{B}\mathbf{Z}_{i,k}^{u}) \geq 0 \end{aligned}$$

As shown below, this problem reduces to a convex optimization problem over polynomials positive in $\|\mathbf{x}\|_2 \leq 1$.

Theorem 1 ([45]). *Problem 2 admits a solution with cost γ if and only if there exist matrices $\mathbf{Y} \in \mathbb{R}^{n\times n} \succ 0$, $\mathbf{M} \in \mathbb{R}^{m\times n}, \mathbf{W} \in \mathbb{R}^{m\times m}$, and a polynomial matrix $\boldsymbol{\Upsilon}(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]^{2n_s\times n} > 0$ in $\|\mathbf{x}\|_2 \leq 1$ such that the following conditions hold:*

$$\boldsymbol{\Xi}\boldsymbol{\Upsilon}(\mathbf{x}) = -2 \begin{bmatrix} \mathbf{Y} \\ \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{H}_R & \mathbf{0} \end{bmatrix} \mathbf{x}\mathbf{x}^T \begin{bmatrix} \mathbf{0} \\ \mathbf{H}_L \\ \mathbf{0} \end{bmatrix} \tag{9a}$$

$$\begin{aligned} &(\gamma - \mathbf{Tr}(\mathbf{Q}\mathbf{Y}) - \mathbf{Tr}(\mathbf{W})) \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}\mathbf{x}^T \begin{bmatrix} 1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\ &+ \mathbf{Tr}\left(\begin{bmatrix} \mathbf{W} & \mathbf{R}^{1/2}\mathbf{M} \\ \mathbf{M}^T\mathbf{R}^{1/2} & \mathbf{Y} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I}_{n+m} \end{bmatrix} \mathbf{x}\mathbf{x}^T \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I}_{n+m} \end{bmatrix} \right) \\ &+ \mathbf{Tr}\left((\mathbf{L}(\mathbf{Y}) + \mathbf{L}_o) \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{x}\mathbf{x}^T \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} \right) - \mathbf{Tr}(\mathbf{D}\boldsymbol{\Upsilon}(\mathbf{x})) > 0 \end{aligned} \tag{9b}$$

for all $1 - \mathbf{x}^T\mathbf{x} \geq 0$

where

$$\begin{aligned} \boldsymbol{\Xi} &\doteq \begin{bmatrix} \boldsymbol{\xi}(t_1) \ldots \boldsymbol{\xi}(t_N) & -\boldsymbol{\xi}(t_1) \ldots - \boldsymbol{\xi}(t_N) \\ \mathbf{u}(t_1) \ldots \mathbf{u}(t_N) & -\mathbf{u}(t_1) \ldots - \mathbf{u}(t_N) \end{bmatrix}, \\ \mathbf{D} &\doteq \begin{bmatrix} \mathbf{d}^+(t_1) \ldots \mathbf{d}^+(t_N) & \mathbf{d}^-(t_1) \ldots \mathbf{d}^-(t_N) \end{bmatrix} \\ \mathbf{d}^+(t_k) &\doteq \epsilon\mathbf{1} + \delta\boldsymbol{\xi}(t_k), \mathbf{d}^-(t_k) \doteq \epsilon\mathbf{1} - \delta\boldsymbol{\xi}(t_k) \end{aligned}$$

Enforcing Putinar's P-satz on $\boldsymbol{\Upsilon}(\mathbf{x})$ and (9b) leads to a convex semi-definite program via SoS arguments [45].

To illustrate the advantages of using duality for scenarios involving ℓ_∞ bounded noise, consider the following discrete-time unstable system [45]:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} -0.1660 & -0.8153 & -0.1616 & 0.3409 & 0.6015 \\ 0.4406 & -0.6275 & 0.3704 & -0.1654 & 0.9365 \\ -0.9998 & -0.3089 & -0.5911 & 0.1174 & -0.3732 \\ -0.3953 & -0.2065 & 0.7562 & -0.7192 & 0.3846 \\ -0.7065 & 0.0776 & -0.9452 & -0.6038 & 0.7528 \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 1.7892 & 0.1701 & 0.0781 & 0.3397 & 1.7563 \\ 0.1967 & 0.8422 & 1.9158 & 1.0663 & 1.3838 \end{bmatrix}^T \end{aligned} \tag{10}$$

For $N = 10$, $\epsilon = 0.05$, the proposed approach leads, in 9.6 seconds, to a robust controller with $\mathcal{H}_2$ cost bounded by $\gamma = 18.7818$ and ground truth $\gamma_0 = 15.1617$. On the other hands, [20] introduces 10^{15} SDP constraints, [47] introduces 227376 variables and [48] leads to around 6.8×10^{12} vertices.

III. Safe Data Driven Control on Nonlinear Systems

In this section we consider the problem of safe data-driven stabilization of a nonlinear system of the form [2]:

$$\begin{aligned}\dot{\mathbf{x}} &= f(\mathbf{x}) + g(\mathbf{x})u + \boldsymbol{\eta} \\ &= \mathbf{F}\boldsymbol{\phi}(\mathbf{x}) + \mathbf{G}\boldsymbol{\gamma}(\mathbf{x})u + \boldsymbol{\eta}\end{aligned} \tag{11}$$

where $\mathbf{x} \in \mathbb{R}^n$, $u \in \mathbb{R}$ and $\boldsymbol{\eta} \in \mathbb{R}^n, \|\boldsymbol{\eta}\|_\infty \leq \epsilon$, represent the state, input and unknown but bounded noise, respectively. Here $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{s_f}, \boldsymbol{\gamma}(\mathbf{x}) \in \mathbb{R}^{s_g}$ denote vectors of known functions of the state $\mathbf{x}$ (the dictionaries), and $\mathbf{F} \in \mathbb{R}^{n\times s_f}$, $\mathbf{G} \in \mathbb{R}^{n\times s_g}$ are unknown system parameter matrices. For example, in the case of two states and second order polynomial dynamics, $\boldsymbol{\phi}(\mathbf{x}) = \boldsymbol{\gamma}(\mathbf{x}) = \begin{bmatrix}1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2\end{bmatrix}^T$, and $\mathbf{F}, \mathbf{G} \in \mathbb{R}^{2\times 6}$ are matrices containing the coefficients of the polynomial vectors $f(\mathbf{x}), g(\mathbf{x})$.

A. Data Driven Stabilization

In principle, our goal is to find a robust control Lyapunov function (CCLF) and associated controller that renders the origin a globally asymptotically stable equilibrium point for all systems in the consistency set. However, this leads to very challenging non-convex problems. Thus, in order to obtain tractable problems, we will consider the slightly weaker condition introduced in Rantzer's dual Lyapunov theory [49] that only requires all trajectories (except those starting in a set of measure zero) to converge to the origin as $t \to \infty$. This leads to the following problem formulation:

Problem 3. *Given N data triplets $\mathcal{D}_N \doteq \{(\dot{\boldsymbol{x}}(t_k), \boldsymbol{x}(t_k), \mathbf{u}(t_k))\}_{k=1}^N$, generated by a system of the form (11) where $\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\gamma}(\mathbf{x})$ are known, find a rational state-feedback control law $u(\mathbf{x}) \doteq \frac{p(\mathbf{x})}{q(\mathbf{x})}$ such that for all $\mathbf{F}, \mathbf{G}$ compatible with the experimental data and almost all initial conditions $\mathbf{x}(0)$, the closed-loop trajectories $\mathbf{x}(t) \to 0$ as $t \to \infty$.*

In order to reformulate Problem 3 as a tractable convex optimization we will exploit the following theorem [49].

Theorem 2. *Given a control-affine system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u(\mathbf{x})$, where $f, g \in \mathbf{C}^1(\mathbb{R}^n, \mathbb{R}^n)$ and $f(\mathbf{0}) = 0$, if there exists $\rho \in \mathbf{C}^1(\mathbb{R}^n\backslash\{0\}, \mathbb{R}^+)$, $u \in \mathbf{C}^1(\mathbb{R}^n, \mathbb{R})$, such that $\rho(f+gu)(\mathbf{x})/\|\mathbf{x}\|_\infty$ is integrable on $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_\infty \geq 1\}$ and*

$$[\nabla \cdot (\rho(f+gu)](\mathbf{x}) > 0 \text{ for almost all } \mathbf{x} \neq 0 \tag{12}$$

then the trajectories $\mathbf{x}(t) \to 0$ as $t \to \infty$, except possibly for those originating in a set of measure 0. Moreover, if $\mathbf{x} = 0$ is a Lyapunov stable equilibrium point, $\mathbf{x}(t) \to 0$ for almost every initial condition, even if ρ takes negative values.

Next, we present a convex reformulation of the Problem 3. The main idea, originally presented in [31] is to consider two sets in parameter space: a) an uncertainty set containing all possible plants compatible with the noisy data and b) the set of plants where there exists a density ρ and associated control action satisfying Theorem 2. These two sets will be related through the Farkas lemma to obtain a condition for the existence of a data-driven controller guaranteed to stabilize all plants in the uncertainty set. Then, this condition will be relaxed to a semi-definite optimization subject to SoS constraints and reduced to a convex semi-definite program. Start by noting that the consistency set $\mathcal{C}(\mathcal{D}_N)$ is given by

$$\mathcal{C}(\mathcal{D}_N) \doteq \{\mathbf{F}, \mathbf{G} : \|\dot{\mathbf{x}} - \mathbf{F}\boldsymbol{\phi} - \mathbf{G}\boldsymbol{\gamma}u\|_\infty \leq \epsilon,\ k = 1 \dots N\} \tag{13}$$

or, equivalently:

$$\mathcal{C}(\mathcal{D}_N) \doteq \{\mathbf{f}, \mathbf{g} : \begin{bmatrix}\mathbf{A} & \mathbf{B} \\ -\mathbf{A} & -\mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{f} \\ \mathbf{g}\end{bmatrix} \leq \begin{bmatrix}\epsilon\mathbf{1} + \boldsymbol{\xi} \\ \epsilon\mathbf{1} - \boldsymbol{\xi}\end{bmatrix}\} \tag{14}$$

where $\mathbf{f} = \text{vec}(\mathbf{F}^T)$, $\mathbf{g} = \text{vec}(\mathbf{G}^T)$,

$$\mathbf{A} \doteq \begin{bmatrix}\mathbf{I} \otimes \boldsymbol{\phi}^T(1) \\ \vdots \\ \mathbf{I} \otimes \boldsymbol{\phi}^T(N)\end{bmatrix}, \mathbf{B} \doteq \begin{bmatrix}\mathbf{I} \otimes u(1)\boldsymbol{\gamma}^T(1) \\ \vdots \\ \mathbf{I} \otimes u(N)\boldsymbol{\gamma}^T(N)\end{bmatrix}, \boldsymbol{\xi} \doteq \begin{bmatrix}\dot{\mathbf{x}}(1) \\ \vdots \\ \dot{\mathbf{x}}(N)\end{bmatrix} \tag{15}$$

and where $\otimes$ denotes the Kronecker product. Next, in order to obtain tractable problems, proceeding as in [50], in the sequel we will consider rational densities $\rho(\mathbf{x})$ and control actions $u(\mathbf{x})$ of the form:

$$\rho(\mathbf{x}) = \frac{a(\mathbf{x})}{b(\mathbf{x})^\lambda},\ u(\mathbf{x})\rho(\mathbf{x}) = \frac{c(\mathbf{x})}{b(\mathbf{x})^\lambda}, b(\mathbf{x}) > 0 \tag{16}$$

where $a(\mathbf{x}), b(\mathbf{x}), c(\mathbf{x})$ are polynomials and λ is chosen to satisfy the integrability condition in Theorem 2. Note that for plants of the form (11) and densities of the form (16), (12) is equivalent to:

$$b\nabla \cdot (\mathbf{F}\boldsymbol{\phi}a + \mathbf{G}\boldsymbol{\gamma}c) - \lambda\nabla b \cdot (\mathbf{F}\boldsymbol{\phi}a + \mathbf{G}\boldsymbol{\gamma}c) > 0 \tag{17}$$

where we explicitly used the fact that $b(\mathbf{x}) > 0$. Given (a, b, c), let $\mathcal{P}(a, b, c)$ denote the set of all plants of the form (11) that can be stabilized by a control action of the form (16), that is

$$\begin{aligned}\mathcal{P}(a, b, c) \doteq \{\mathbf{F}, \mathbf{G} : &\text{(17) holds for given} \\ &\text{polynomials } a, b, c\ \}\end{aligned} \tag{18}$$

In this context, Problem 3 reduces to

Problem 4. *Find polynomials $a(\mathbf{x}), b(\mathbf{x}), c(\mathbf{x})$ such that $\mathcal{C}(\mathcal{D}_n) \subseteq \mathcal{P}(a, b, c)$.*

The first step towards enforcing the inclusion above is to obtain a representation of $\mathcal{P}$ in terms of $\mathbf{f}, \mathbf{g}$. To this effect, define $a = \mathbf{c}_a^T\mathbf{m}$, $b = \mathbf{c}_b^T\mathbf{m}$, $c = \mathbf{c}_c^T\mathbf{m}$, where $\mathbf{c}_a, \mathbf{c}_b, \mathbf{c}_c$ are coefficient vectors and

$$\mathbf{m} \doteq \begin{bmatrix}1 & x_1 \dots (x_1^{\alpha_1}..x_n^{\alpha_n}) \dots x_n^m\end{bmatrix}^T \tag{19}$$

is a vector of monomials ordered in a graded reversed (grevlex) order of the corresponding size. Substituting these

[2] For notational simplicity we consider single input systems but the results here generalize trivially to the multiple inputs case.

definitions in (17) leads to:

$$\begin{aligned}\mathbf{c}_b^T\mathbf{m}\nabla\cdot(\mathbf{c}_a^T\mathbf{m}\mathbf{F}\boldsymbol{\phi}+\mathbf{c}_c^T\mathbf{m}\mathbf{G}\boldsymbol{\gamma})-\\ \alpha\nabla(\mathbf{c}_b^T\mathbf{m})\,(\mathbf{c}_a^T\mathbf{m}\mathbf{F}\boldsymbol{\phi}+\mathbf{c}_c^T\mathbf{m}\mathbf{G}\boldsymbol{\gamma})>0\end{aligned} \tag{20}$$

Explicitly computing the first term in (20) yields:

$$\begin{aligned}\mathbf{c}_b^T\mathbf{m}[\mathbf{c}_a^T\frac{\partial\mathbf{m}\boldsymbol{\phi}^T}{\partial x_1}\dots\mathbf{c}_a^T\frac{\partial\mathbf{m}\boldsymbol{\phi}^T}{\partial x_n}\\ \mathbf{c}_c^T\frac{\partial\mathbf{m}\boldsymbol{\gamma}^T}{\partial x_1}\dots\mathbf{c}_c^T\frac{\partial\mathbf{m}\boldsymbol{\gamma}^T}{\partial x_n}]\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix}\doteq\mathbf{d}_1(\mathbf{x})\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix}\end{aligned} \tag{21}$$

Similarly, the second term in (20) is given by:

$$\begin{aligned}-\lambda[\mathbf{c}_b^T\frac{\partial\mathbf{m}}{\partial x_1}\mathbf{c}_a^T\mathbf{m}\boldsymbol{\phi}^T\dots\mathbf{c}_b^T\frac{\partial\mathbf{m}}{\partial x_n}\mathbf{c}_a^T\mathbf{m}\boldsymbol{\phi}^T\\ \mathbf{c}_b^T\frac{\partial\mathbf{m}}{\partial x_1}\mathbf{c}_c^T\mathbf{m}\boldsymbol{\gamma}^T\dots\mathbf{c}_b^T\frac{\partial\mathbf{m}}{\partial x_n}\mathbf{c}_c^T\mathbf{m}\boldsymbol{\gamma}^T]\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix}\doteq\mathbf{d}_2(\mathbf{x})\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix}\end{aligned} \tag{22}$$

Rewriting $\mathcal{P}$ in terms of these expressions leads to:

$$\mathcal{P}\doteq\{\mathbf{f},\mathbf{g}:-\left[\mathbf{d}_1(\mathbf{x})+\mathbf{d}_2(\mathbf{x})\right]\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix}<0\} \tag{23}$$

Note that the inequalities above are functional inequalities that must hold for all $\mathbf{x}$ and some coefficient vectors $\mathbf{c}_a,\mathbf{c}_b,\mathbf{c}_c$. Using the characterization above and the extended Farkas' Lemma [51] to enforce that $\mathcal{C}(\mathcal{D}_N)\subseteq\mathcal{P}$(a,b,c) leads to the following result showing that Problem 4 reduces to a (functional) Linear Program:

Theorem 3. *[31] Given the experimental data $\mathcal{D}_N$ collected from trajectories of (11), form the corresponding matrices $\mathbf{A},\mathbf{B},\boldsymbol{\xi}$, defined in (15). Then, there exist polynomials $a,b>0,c$ such that $\mathcal{C}(\mathcal{D}_N)\subseteq\mathcal{P}$(a,b,c) if and only if there exists a vector function $\mathbf{Y}(\mathbf{x})\in\mathbb{R}^{1\times 2NT}\geq 0$ such that the following (functional) set of affine constraints is feasible:*

$$\mathbf{Y}(\mathbf{x})\mathbf{M}=\mathbf{d}(\mathbf{x})\ \textit{and}\ \mathbf{Y}(\mathbf{x})\mathbf{e}<0 \tag{24}$$

where for notational simplicity we defined

$$\begin{aligned}\mathbf{M}\doteq\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ -\mathbf{A} & -\mathbf{B}\end{bmatrix},\ \mathbf{e}=\begin{bmatrix}\epsilon\mathbf{1}+\boldsymbol{\xi}\\ \epsilon\mathbf{1}-\boldsymbol{\xi}\end{bmatrix}\\ \mathbf{d}\doteq-\left[\mathbf{d}_1(\mathbf{x})+\mathbf{d}_2(\mathbf{x})\right],\end{aligned} \tag{25}$$

Further, (i) if $\boldsymbol{\phi}(\mathbf{x}),\boldsymbol{\gamma}(\mathbf{x})$ are continuous functions, then $\mathbf{Y}(\mathbf{x})$ can be chosen to be continuous, and (ii) if the trajectories of the closed-loop system stay in a compact region $\mathcal{D}$ and $\boldsymbol{\phi},\boldsymbol{\gamma}$ are polynomial, then $\mathbf{Y}(\mathbf{x})$ can be taken to be polynomial.

In the case of polynomial dynamics, Theorem 3 provides a necessary & sufficient condition for the existence of a rational density and associated control action that stabilizes the unknown plant, given in terms of existence of positive polynomials satisfying a set of linear (in)equalities. However, certifying positivity of a polynomial in more than 2 variables is NP hard[3] [52]. A tractable relaxation can be obtained as follows. First relax $\mathbf{Y}(\mathbf{x})$ to a vector $\mathbf{Y}_{SoS}(\mathbf{x})=[y_1(\mathbf{x}),\dots,y_i(\mathbf{x}),\dots,y_{2NT}(\mathbf{x})]$ where $y_i(\mathbf{x})\doteq\mathbf{c}_{y_i}^T\mathbf{m}$ is SoS. In this scenario (24) reduces to:

$$\mathbf{k}_l=\mathbf{k}_r,\text{and}-\mathbf{Y}\mathbf{e}\in\Sigma[\mathbf{x}] \tag{26}$$

where $\mathbf{k}_l,\mathbf{k}_r$ are the coefficients of the polynomials $\mathbf{Y}(\mathbf{x})\mathbf{M}$ and $\mathbf{d}(\mathbf{x})$. Since the entries of $\mathbf{k}_l$ are linear functions of $\mathbf{c}_{y_i}^T$ and the entries of $\mathbf{k}_r$ are bilinear in $\mathbf{c}_b,\mathbf{c}_c$ and $\mathbf{c}_b,\mathbf{c}_a$, the problem above is a Quadratically Constrained Quadratic Program (QCQP). In turn, this QCQP can be relaxed to an SDP using tools from semi-algebraic optimization (see [31] for details). To illustrate these results, consider the following system taken from [50]:

$$\begin{aligned}\dot{x}_1&=x_2-x_1^3+x_1^2+\eta_1\\ \dot{x}_2&=u+\eta_2\end{aligned} \tag{27}$$

The system was excited for 5 seconds with a random signal uniformly distributed in [-1,1], starting from the initial condition [1; -1]. The data record $\mathcal{D}_N$ consists of N = 500 samples, corrupted by noise uniformly distributed in [-0.05, 0.05]. As in [50], we chose $\lambda=4$. We used a quadratic function: $b(\mathbf{x})=\mathbf{x}^T\mathbf{P}\mathbf{x}$ and selected $d_f=3,d_g=0,d_a=0,d_c=3,l=3,h=1\text{e}{-4}$. The proposed approach led, in 112s, to a third-order controller:

$$\begin{aligned}u(\mathbf{x})=&-0.0293x_1^3-0.452x_1^2x_2-0.0614x_1^2-0.0328x_1x_2^2\\ &+0.11x_1x_2-1.25x_1-0.378x_2^3-0.336x_2^2-0.955x_2\end{aligned}$$

The phase plot of the corresponding closed-loop system is shown in Fig. 1.

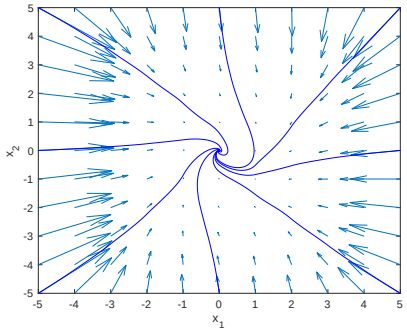


Fig. 1. Phase plot of the closed-loop system for the example above.

B. Adding Safety Constraints

In many scenarios of practical interest, the goal is not only to drive the system to a given target set, but also to avoid an unsafe region while doing so. Formally, given an initial set $\mathcal{X}_i$, an unsafe set $\mathcal{X}_{\text{uns}}$, and target set $\mathcal{X}_r$, the system (11):

1) Is *safe with respect to* $\mathcal{X}_{\text{uns}}$ if there exists $\mathbf{u}:\mathcal{X}\to\mathcal{U}$ such that for all $\mathbf{x}_0\in\mathcal{X}_i$, $\forall t\in[0,\infty)$, $x(t)\notin\mathcal{X}_{\text{uns}}$;
2) *Eventually reaches* $\mathcal{X}_r$ if there exists $\mathbf{u}:\mathcal{X}\to\mathcal{U}$ such that for all $\mathbf{x}_0$ in $\mathcal{X}_i$, $\exists T\in[0,\infty)$, $x(T)\in\mathcal{X}_r$;
3) Is *safe with respect to* $\mathcal{X}_{uns}$ *until it eventually reaches* $\mathcal{X}_r$ if there exists $\mathbf{u}:\mathcal{X}\to\mathcal{U}$ such that for all $\mathbf{x}_0$ in $\mathcal{X}_i$, $\exists T\in[0,\infty)$, $x(T)\in\mathcal{X}_r$ and $\forall t\in[0,T]$ $x(T)\notin\mathcal{X}_{\text{uns}}$.

A shown in [53], [54] a DDC that guarantees robust safe eventuality can be obtained by adding constraints to the basic formulation discussed in Theorem 2.

[3]Except in the case of quartic polynomials in two variables.

Theorem 4 ([54]). *Consider the dynamical system (11). Let $V \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}(.))$, $V(x) > 0\ \forall x \notin \mathcal{X}_r$ and $\alpha \geq 1$ be a given function and scalar, respectively. Then, if there exist a density function $\bar{\rho} : \mathbb{R}^n \to \mathbb{R}$ and a vector field $\bar{\boldsymbol{\psi}} : \mathbb{R}^n \to \mathbb{R}^n$ continuously differentiable such that the following conditions hold*

$$V(\mathbf{x}) > 0,\ \forall \mathbf{x} \notin \mathcal{X}_r, \tag{28}$$

$$\bar{\rho}(\mathbf{x}) \geq 0,\ \forall \mathbf{x} \in \mathcal{X}_i, \tag{29}$$

$$\bar{\rho}(\mathbf{x}) < 0,\ \forall \mathbf{x} \in \mathcal{X}_{uns}, \tag{30}$$

$$\begin{aligned} &\nabla \cdot (\bar{\rho}(f+\eta) + g\bar{\boldsymbol{\psi}}) > \\ &\quad \alpha \frac{\nabla V}{V} \cdot (\bar{\rho}(f+\eta) + g\bar{\boldsymbol{\psi}}), \\ &\qquad \forall \|\eta\|_\infty \leq \epsilon \text{ and } \forall \mathbf{x} \notin \mathcal{X}_r), \end{aligned} \tag{31}$$

then, the control input $\mathbf{u} = \frac{\bar{\boldsymbol{\Psi}}}{\bar{\rho}}$ enforces the weak eventuality *properties on (11) in a robust manner i.e. for almost all $\mathbf{x}_0 \in \mathcal{X}_i$, $\exists T \geq 0\ x(T) \in \mathcal{X}_r$ and $\forall t \in [0, T]\ x(t) \notin \mathcal{X}_{uns}$ for all disturbances $\eta, \|\eta\|_\infty \leq \epsilon$.*

As before, for the case of polynomial dynamics and semi-algebraic initial, target and unsafe sets, selecting polynomial $\bar{\rho}$ and $\bar{\boldsymbol{\psi}}$ leads to a semi-algebraic optimization problem that can be relaxed to an SDP (see [54] for details).

IV. Data Driven Model Predictive Control

In this section, we present a recently developed data-driven min-max model predictive control (MPC) schemes in the data informativity framework to design a controller that robustly stabilizes the system and is able to handle ellipsoidal input and state constraints. The proposed data-driven min-max MPC scheme is developed for unknown linear time-invariant (LTI) systems using noisy input-state data for controller synthesis [28], [55]. Later, the scheme has been extended to other system classes, including linear parameter-varying (LPV) systems [56] and bilinear systems [57]. Over all kind of system setups, the data-driven min-max MPC problem with input and state constraints is reformulated to a semidefinite program (SDP) that yields a time-varying state-feedback control law. We show that the proposed data-driven min-max MPC guarantees closed-loop properties including recursive feasibility, constraint satisfaction and (robust) stability.

Notations: Let $\mathbb{I}_{[a,b]}$ denote the set of integers in the interval $[a, b]$ and $\mathbb{I}_{\geq 0}$ denote the set of nonnegative integers. For a matrix P, we write $P \succ 0$ if P is positive definite and $P \succeq 0$ if P is positive semi-definite. For a vector x and a matrix $P \succ 0$, we write $\|x\|_P = \sqrt{x^\top P x}$.

A. A Data-driven Min-Max MPC Approach for LTI Systems

First, we present the data-driven min-max MPC scheme for LTI systems from [28], [55] in this section. We consider an unknown discrete-time LTI system

$$x_{t+1} = A_s x_t + B_s u_t + \omega_t, \tag{32}$$

where $x_t \in \mathbb{R}^n$ denotes the state, $u_t \in \mathbb{R}^m$ denotes the input, and $\omega_t \in \mathbb{R}^n$ denotes the unknown noise for $t \in \mathbb{I}_{\geq 0}$. The matrices $A_s \in \mathbb{R}^{n \times n}$ and $B_s \in \mathbb{R}^{n \times m}$ are assumed to be unknown. The noise ω_t is assumed to satisfy the following assumption.

Assumption 1. For all $t \in \mathbb{I}_{\geq 0}$, the noise $\omega_t \in \mathbb{R}^n$ satisfies $\|\omega_t\|_G \leq 1$ for a known matrix $G \succ 0$.

We define a sequence of input and corresponding state measurements, which is denoted by

$$U_f = \begin{bmatrix} u_0^f & u_1^f & \dots & u_{T_f-1}^f \end{bmatrix}, X_f = \begin{bmatrix} x_0^f & x_1^f & \dots & x_{T_f}^f \end{bmatrix}.$$

The data-driven parameterization method employed in this work was introduced in [17]. It characterizes a set of system matrices for LTI systems that are consistent with a given sequence of input and state measurements, along with specified assumptions on the noise. The set of system matrices consistent with the sequence of offline input-state measurements (U_f, X_f) is parameterized by the following quadratic matrix inequality (QMI)

$$\mathcal{C}_f = \left\{ (A, B) \;\middle|\; \begin{bmatrix} I & A & B \end{bmatrix} \Pi_f(\tau) \begin{bmatrix} I & A & B \end{bmatrix}^\top \succeq 0, \atop \forall \tau = (\tau_0, \dots, \tau_{T_f-1}), \tau_i \geq 0, i \in \mathbb{I}_{[0,T_f-1]} \right\}, \tag{33}$$

where

$$\Pi_f(\tau) = \sum_{i=0}^{T_f-1} \tau_i \begin{bmatrix} I & x_{i+1}^f \\ 0 & -x_i^f \\ 0 & -u_i^f \end{bmatrix} \begin{bmatrix} G^{-1} & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} I & x_{i+1}^f \\ 0 & -x_i^f \\ 0 & -u_i^f \end{bmatrix}^\top. \tag{34}$$

Our objective is to stabilize the origin for the unknown LTI system (32), while the closed-loop input and state satisfy given ellipsoidal constraints, i.e., $\|u_t\|_{S_u} \leq 1, \|x_t\|_{S_x} \leq 1, \forall t \in \mathbb{I}_{\geq 0}$. A quadratic stage cost function $l(u, x) = \|u\|_R^2 + \|x\|_Q^2$ with $R, Q \succ 0$ is defined to stabilize the origin. Although the exact system dynamics are unknown, we know that the true system matrices (A_s, B_s) lie inside the set $\mathcal{C}_f$. Therefore, we focus on designing a controller that robustly stabilizes all systems characterized by matrices inside the set $\mathcal{C}_f$.

At time t, given offline input-state measurements (U_f, X_f) and an initial state x_t, the data-driven min-max MPC optimization problem is formulated as follows:

$$J_\infty^*(x_t) := \min_{\bar{u}(t)} \max_{(A,B) \in \mathcal{C}_f} \sum_{k=0}^{\infty} l(\bar{u}_k(t), \bar{x}_k(t)) \tag{35a}$$

$$\text{s.t.} \quad \bar{x}_{k+1}(t) = A\bar{x}_k(t) + B\bar{u}_k(t), \tag{35b}$$

$$\bar{x}_0(t) = x_t. \tag{35c}$$

$$\|\bar{u}_k(t)\|_{S_u} \leq 1, \forall k \in \mathbb{I}_{\geq 0}, \tag{35d}$$

$$\|\bar{x}_k(t)\|_{S_x} \leq 1, \forall (A, B) \in \mathcal{C}_f, k \in \mathbb{I}_{\geq 0}, \tag{35e}$$

The objective function aims to minimize the worst-case value of the sum of infinite stage costs among all consistent system matrices in $\mathcal{C}_f$ by adapting the control input $\bar{u}_k(t), \forall k \in \mathbb{I}_{\geq 0}$. In the optimization problem, $\bar{x}_k(t)$ and $\bar{u}_k(t)$ are the predicted state and control input at time $t + k$ based on the measurement at time t. We use the nominal system dynamics (35b) with $(A, B) \in \mathcal{C}_f$ for future state prediction in constraint (35b). In constraint (35c), we initialize $\bar{x}_0(t)$

as the state measurement at time t. In constraints (35d) and (35e), the predicted input and state satisfy the ellipsoidal constraints for any nominal system dynamics (35b) with $(A, B) \in \mathcal{C}_f$.

The data-driven min-max MPC problem (35) is intractable because of the min-max formulation and the constraints for all possible (A, B) within a set characterized by data. To effectively address problem (35) and derive a tractable solution, we focus on finding a time-varying state-feedback control law of the form $\bar{u}_k(t) = F_t \bar{x}_k(t)$, where $F_t \in \mathbb{R}^{m\times n}$ will later be determined online in an MPC fashion. In the following, we formulate an SDP to derive an upper bound on the optimal cost over the set $\mathcal{C}_f$ and to determine a state-feedback gain that minimizes this upper bound.

At time t, given offline input-state measurements (U_f, X_f), an initial state $x_t \in \mathbb{R}^n$ and a constant $c > \lambda_{\min}(Q)$, the SDP is formulated as follows:

$$\text{minimize}_{\gamma>0, H\in\mathbb{R}^{n\times n}, L\in\mathbb{R}^{m\times n}, \tau\in\mathbb{R}^{T_f}} \gamma \tag{36a}$$

$$\text{s.t.} \begin{bmatrix} 1 & x_t^\top \\ x_t & H \end{bmatrix} \succeq 0, \tag{36b}$$

$$\begin{bmatrix} \begin{bmatrix} -H + \frac{\gamma}{c} I & 0 \\ 0 & 0 \end{bmatrix} + \Pi_f(\tau) & \begin{bmatrix} 0 \\ H \\ L \end{bmatrix} & 0 \\ \begin{bmatrix} 0 & H & L^\top \end{bmatrix} & -H & \Phi^\top \\ 0 & \Phi & -\gamma I \end{bmatrix} \prec 0, \tag{36c}$$

$$\tau = (\tau_0, \ldots, \tau_{T_f-1}), \tau_i \geq 0, \forall i \in \mathbb{I}_{[0,T_f-1]}, \tag{36d}$$

$$\begin{bmatrix} H & L^\top \\ L & S_u^{-1} \end{bmatrix} \succeq 0, \begin{bmatrix} H & H \\ H & S_x^{-1} \end{bmatrix} \succeq 0. \tag{36e}$$

where $\Phi = \begin{bmatrix} M_R L \\ M_Q H \end{bmatrix}$, $M_R^\top M_R = R$ and $M_Q^\top M_Q = Q$. The optimal solution of problem (36) depends on the current measured state x_t. For a given state x_t, the optimal solution of problem (36) is denoted by $\gamma^\star_{x_t}, H^\star_{x_t}, L^\star_{x_t}, \tau^\star_{x_t}$. The corresponding optimal state-feedback gain is given by $F^\star_{x_t} = L^\star_{x_t}(H^\star_{x_t})^{-1}$ and we define $P^\star_{x_t} = \gamma^\star_{x_t}(H^\star_{x_t})^{-1}$.

Problem (36) provides an upper bound on the optimal cost of problem (35) and a corresponding state-feedback control gain. Additionally, the constraints (36e) ensure that the input and state constraints are satisfied for the closed-loop system and the derived state-feedback gain. We solve the SDP problem (36) in a receding-horizon fashion to iteratively compute an optimal state-feedback gain. Once the state enters a neighborhood of the origin characterized by the robust positively invariant (RPI) set, a fixed state-feedback control law is implemented. Outside this neighborhood, the receding-horizon strategy continues to solve the SDP problem (36) at each time step to update the state-feedback control law.

In Theorem 2 in [55], we show that if the optimization problem (36) is feasible at initial time $t = 0$ and $\|x_0\|^2_{P^\star_{x_0}} \geq c_{RPI}$ with $c_{RPI} = \frac{c^2}{\lambda_{\min}(Q)\lambda_{\min}(G)}$, then the closed-loop trajectory converges robustly and exponentially to the RPI set for the closed-loop system with any $(A, B) \in \mathcal{C}_f$. We show that the optimal cost decays exponentially when the state is outside the RPI set. Once the state enters the RPI set, we prove robust positive invariance of the RPI set under a suitable state-feedback law defined by the optimal solution at one time step. We show that the input and state constraints are satisfied for the closed-loop system.

The data-driven min-max MPC scheme employs only an offline input-state data sequence to characterize consistent systems matrices for prediction. During online operation, the collection of additional online data can improve performance especially when offline data are inadequate. An adaptive data-driven min-max MPC scheme is proposed to use both offline and online data in [55]. Similar closed-loop guarantees are established for the adaptive data-driven min-max MPC scheme and numerical examples show that incorporating online data leads to improved closed-loop performance.

B. Data-driven Min-Max MPC: Towards Nonlinearity

The proposed data-driven min-max MPC scheme is originally developed for LTI systems. However, many real-world systems have nonlinear dynamics. Some data-driven controller design methods have been extended to certain classes of nonlinear systems [58]. To improve the scalability and applicability of the proposed data-driven min-max MPC scheme, several extensions have been proposed to account for nonlinearities in the system dynamics [56], [57].

LPV systems are linear systems with time-varying scheduling signals and they provide a powerful framework for addressing nonlinear analysis and control problems using linear methods [59]. In [56], we propose a data-driven min-max MPC scheme for LPV systems using only a sequence of input-state data. We assume that the system matrices and the scheduling signal of the LPV system are unknown, but the latter satisfies a known QMI. Based on this QMI, we characterize all system matrices explaining the available input-state data. The proposed data-driven min-max MPC minimizes a tractable upper bound on the worst-case cost over the consistent system matrices set and all scheduling signals satisfying the QMI. The proposed scheme guarantees that all systems consistent with the data are exponentially stabilizes and satisfy the constraints for all scheduling signals satisfying the QMI.

In [57], we consider a specific class of discrete-time polynomial systems, namely bilinear systems, and derive a controller design method for such systems using no explicit model knowledge but only noisy data. Bilinear systems represent a special class of nonlinear systems, where the nonlinearity arises from the product of the states and input [60]. The proposed scheme is based on characterizing the set of bilinear systems that are consistent with a sequence of noisy input-state data. We formulate an SOS program that minimizes an upper bound on the bilinear data-driven min-max MPC cost and derive a corresponding rational feedback controller. The proposed scheme guarantees that the closed-loop system is robustly stabilized to a RPI set around the origin and satisfies the constraints. In contrast to existing data-driven control approaches for polynomial

systems, which require derivative data, the proposed method only relies on discrete input-state measurements.

While the described extensions for nonlinear systems are still limited to specific subclasses of nonlinear systems, they show that this framework holds promise to extend robust data-driven min-max MPC to a wider class of application relevant systems. The goal of future research is to further investigate and generalize this framework to accommodate a broader class of nonlinear system models.

V. Machine Learning Motivated Approaches

As an alternative to the explicit optimization based approaches described in the previous sections, here we discuss some machine learning motivated approaches both for optimal control and optimal filtering.

A. Policy optimization and overparameterization for LTI optimal control

In a scenario where the model is unknown but one could sample information from the system (either from experiments with the real plant or from nonlinear simulations), reinforcement learning techniques are a popular way to obtain an optimal policy (control law). The study of the linear quadratic regulator (LQR) through the lens of policy optimization (model-free LQR) is popular in the literature [61]–[67] with results guaranteeing convergence and optimality of gradient-based optimization algorithms. Furthermore, in the reinforcement learning literature, deep neural network architectures can be leveraged to improve both the convergence of training and performance of the resulting solution [68], motivating the interest for such architectures in the context of control design. However, despite its popularity, some care is necessary when analyzing the convergence rate of the policy optimization, especially when comparing between continuous- and discrete-time formulations for the LQR, and the possible introduction of spurious equilibria to the optimization landscape in an overparameterized formulation.

To illustrate this, consider the continuous-time LQR problem, which consists of finding $K^* \in \mathbb{R}^{m\times n}$ and $\underline{J} := J(K^*)$ that solve the problem.

$$\begin{aligned} \underset{K\in\mathscr{K}}{\text{minimize}} \quad & J(K) := \operatorname{trace}(P_K) \\ \text{subject to} \quad & (A-BK)^\top P_K + P_K(A-BK) \\ & \quad + K^\top R K + Q = 0 \end{aligned} \tag{37}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the drift and input matrices of the linear dynamics, $R \in \mathbb{S}^m_{++}$ and $Q \in \mathbb{S}^n_{++}$ are the parameters of the optimization, and $\mathscr{K} := \{K \in \mathbb{R}^{m\times n} \mid (A-BK)$ is Hurwitz$\}$. The discrete-time LQR has an analogous formulation, except $\mathscr{K} := \{K \in \mathbb{R}^{m\times n} \mid |\lambda_i(A-BK)| < 1,\ i = 1, 2 \dots n\}$ and P_K is the positive definite solution of

$$(A-BK)^\top P_K (A-BK) - P_K + K^\top R K + Q = 0$$

instead. The policy optimization (gradient flow) of either such a problem is the following dynamical system

$$\dot{K} = -\nabla J(K), \tag{38}$$

whose solution $\phi : \mathbb{R}_+ \times \mathscr{K} \to \mathscr{K}$ can be shown to have a unique steady state at the optimal solution of (37). We, then, are interested in analyzing the convergence of this system in both the form described above and in its overparameterized form to be presented later. The given optimization problem, however, is known to be nonconvex, meaning we need to leverage more general tools to study its convergence.

1) The Polyak-Łojasiewicz inequality and the convergence of gradient methods: The Polyak-Łojasiewicz inequality (PŁI) is a staple in nonconvex optimization, due to guaranteeing optimality and exponential convergence of gradient methods. However, its characterization varies in the literature, even in between the discrete-time and the continuous-time LQR problem. To see that, consider the following definitions.

Definition 2 (global PŁI)**.** *A function J satisfies a* global Polyak-Łojasiewicz inequality *(gPŁI) if there exists a $\mu > 0$ such that*

$$\|\nabla J(K)\| \geq \alpha\left(J(K) - \underline{J}\right). \tag{39}$$

with $\alpha(r) = \sqrt{\mu r}$, for all $K \in \mathscr{K}$.

Definition 3 (Saturated PŁI)**.** *A function J satisfies a* saturated Polyak-Łojasiewicz inequality *(satPŁI) if there exist $a, b > 0$ such that*

$$\|\nabla J(K)\| \geq \alpha\left(J(K) - \underline{J}\right). \tag{40}$$

with $\alpha(r) = \sqrt{ar/(b+r)}$, for all $K \in \mathscr{K}$.

Definition 4 (Semi-global PŁI)**.** *A function J satisfies a* semi-global Polyak-Łojasiewicz inequality *(sgPŁI) if, for every $\epsilon > 0$, there exists a $\mu_\epsilon > 0$ such that*

$$\|\nabla J(K)\| \geq \alpha_\epsilon(J(K) - \underline{J}), \tag{41}$$

with $\alpha_\epsilon(r) = \sqrt{\mu_\epsilon r}$, for all $K \in \mathscr{K}_\epsilon := \{K \in \mathscr{K} \mid J(K) - \underline{J} \leq \epsilon\}$.

It is known that the discrete-time LQR satisfies a *g*PŁI [64], while the continuous-time LQR has *sg*PŁI guarantees [63], [67] with a recent work showing that it also satisfies a *sat*PŁI [69]. Although the inequalities might look very similar at first, some of its differences can be made explicit as in the following lemma.

Lemma 1. *If a function J satisfies a gPŁI, then it also satisfies a sgPŁI. Alternatively, if a function J satisfies a semi-global PŁI but not a global PŁI for any $\mu > 0$, then*

$$\lim_{\epsilon\to\infty} \overline{\mu}_\epsilon = 0, \tag{42}$$

where $\overline{\mu}_\epsilon$ is the largest μ_ϵ that satisfies (41) for a given $\epsilon > 0$.

Notice that as a consequence of Lemma 1, a cost function that is *sg*PŁI but not *g*PŁI then there must be some "unbounded sequence" of points $\{K_i\}$, $K_i \in \mathscr{K}$, with $\lim_{i\to\infty} J(K_i) - \underline{J} = \infty$, for which the $\overline{\mu}_{J(K_i)-\underline{J}}$ goes to zero as i goes to infinity. We show next that as a consequence of this fact we can conclude that the convergence is not purely exponential globally.

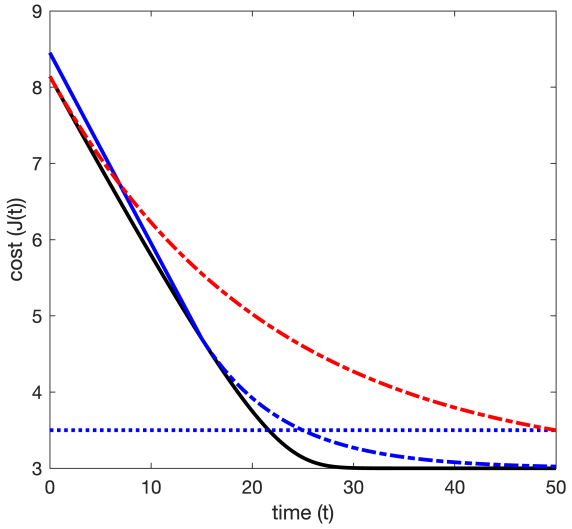


Fig. 2. Numerical simulation for a cost function that is *sat*PŁI and has a globally bounded gradient, illustrating the linear-exponential profile of the solution.

Definition 5 (Global exponential stability)**.** *The gradient flow* (38) *of J is globally exponential stable (GES) in $\mathscr{K}$ if there exists a $\mu > 0$ such that for every $K_0 \in \mathscr{K}$*

$$J(\phi(t, K_0)) - \underline{J} \leq (J(K_0) - \underline{J})e^{-\mu t} \text{ for all } t \geq 0, \quad (43)$$

where $\phi(\cdot, K_0)$ is the solution of (38) *initialized at K_0.*

Definition 6 (global linear-exponential stability)**.** *The gradient flow* (38) *of J is globally linear-exponential stable (GLES) if there exists an $m > 0$ such that for every $K_0 \in \mathscr{K}$ and every $\underline{t} > 0$ there exists a $\mu_{K_0,\underline{t}} > 0$ for which*

$$(J(\phi(t,K_0)) - \underline{J}) \quad (44)$$
$$\leq \begin{cases} (J(\phi(\underline{t}, K_0)) - \underline{J}) - m(t - \underline{t}) & \text{if } t \leq \underline{t} \\ (J(\phi(\underline{t}, K_0)) - \underline{J})e^{-\mu_{K_0,\underline{t}}(t-\underline{t})} & \text{if } t > \underline{t} \end{cases},$$

for all $t \in \mathbb{R}$.

Lemma 2. *The gradient flow* (38) *of J is GES in $\mathscr{K}$ if and only if J satisfies a global PŁI. Alternatively, J is sgPŁI if and only if for every $\epsilon > 0$ there exists a $\mu_\epsilon > 0$ such that the gradient flow* (38) *is μ_ϵ-GES in $\mathscr{K}_\epsilon := \{K \in \mathscr{K} \mid J(K) \leq \epsilon\}$ (i.e. it is GES in the levelsets). Finally, if J is satPŁI and has a globally upper-bounded gradient, then its gradient flow is GLES.*

From Lemmas 1 and 2 one can conclude that if the cost satisfies only a *sat*PŁI or a *sg*PŁI then, in the worst-case, the rate of convergence becomes sub-exponential the farther from the optimum one initializes the gradient flow (as measured by the cost). That is made explicit by characterizing the convergence of the gradient flow of a *sat*PŁI cost function as globally linear-exponential (with the additional assumption of globally bounded gradient).

We illustrate the linear-exponential convergence property in Fig. 2, where we plot the gradient flow solution for a cost function that is *sat*PŁI and that has a globally bounded gradient. The resulting trajectory can be seen in black in Fig. 2 and can be upper-bounded by both a pure exponential (since it is exponentially convergent in the level-set) or by a linear-exponential curve. Notice, however, that the exponential upper-bound (in red) takes much longer to reach tolerance of 0.5 from the optimum (blue dotted line), while the linear-exponential upper-bound (in blue) does so much sooner.

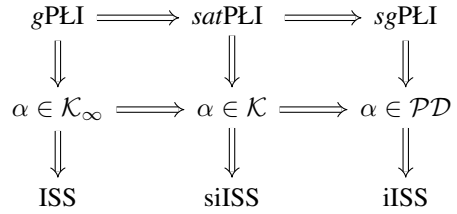


Fig. 3. Diagram of the hierarchy between types of comparison functions and their relationship with the different types of Polyak-Łojasiewicz inequality, and the robustness of the resulting gradient flow.

As mentioned, literature results characterize a *g*PŁI for the discrete-time LQR problem, while only recent results have shown that a *sat*PŁI can be characterized for the continuous-time problem. Although this does not guarantee a priori that the continuous-time LQR does not also satisfy some unknown *g*PŁI, in another recent work [70] we showed that this is indeed the case, *i.e.* the continuous-time LQR can never satisfy a *g*PŁI. Despite that, it can be also shown that this cost function does not have a globally bounded gradient, and as such we cannot characterize global linear-exponential behavior. We refer the reader to [70] for a deeper discussion of this duality in the continuous-time LQR cost, and its consequence to the profile of the cost function.

2) On the robustness of the gradient flow: Beyond concerns regarding the convergence rate of the gradient flow solution, one might be interested in studying guarantees for when the gradient of the cost is known not exactly, but approximately and subject to some disturbance $U : \mathbb{R}_+ \to \mathbb{R}^{n\times m}$. The disturbed gradient flow is defined as

$$\dot{K} = -\nabla J(K) + U(t) \quad (45)$$

and its robustness can be characterized in terms of the class of PŁI it satisfies shown in the following lemma, which consolidates results extracted from [69]

Lemma 3. *Let $\alpha : \mathbb{R} \to \mathbb{R}_+$ be such that for all $K \in \mathscr{K}$, the following holds*

$$\|\nabla J(K)\| \geq \alpha\left(J(K) - \underline{J}\right).$$

If α is a class $\mathcal{K}_\infty$ comparison function, then the gradient flow (45) *of J is input-to-state stable (ISS). Alternatively, if α is a class $\mathcal{K}$ comparison function, then* (45) *is strong integral input-to-state stable (siISS), i.e. it is both ISS with respect to small inputs, and integral ISS. Finally, if α is a positive-definite function, then its disturbed gradient flow* (45) *is integral input-to-state stable (iISS).*

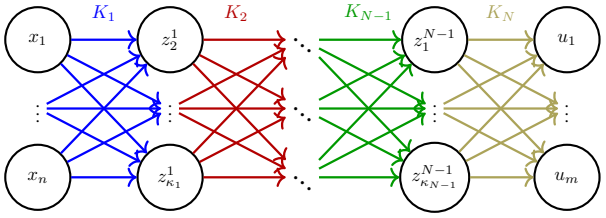


Fig. 4. Graphical representation of a linear feedforward neural network (LFFNN) with input layer $x \in \mathbb{R}^n$, hidden layers $z^i \in \mathbb{R}^{\kappa_i}$ and output layer $u \in \mathbb{R}^m$.

Note that the conditions that α belongs to different classes of comparison function are different from the *g*PŁI, *sat*PŁI , and *sg*PŁI, as defined before. The relationship between the different conditions is illustrated in Fig. 3. All of these conditions are sufficient for asymptotic convergence of the gradient flow to the global minimum, however its convergence rate and robustness to disturbance can vary significantly.

3) Overparameterized state feedback: Alternatively, we can consider the case where the policy (feedback matrix) is computed by a linear feed-forward neural network (LFFNN) with $N-1$ hidden layers $\mathbf{K}(x) = K_N \dots K_1 x$, as depicted in Fig. 4. Notice that by foregoing the nonlinear activation, the resulting neural network can only approximate linear mappings. However the LQR is known to have a linear optimal solution, therefore there is no loss in optimality by this assumption. Furthermore, by considering linear activation functions, the analysis can be more rigorously done, and theoretical guarantees can be obtained for the problem.

The overparameterized LQR consists in finding a LFFNN $\mathbf{K}^*$ and $\underline{J} := J(\mathbf{K}^*)$ that solve

$$\begin{aligned} \underset{\mathbf{K}\in\mathscr{K}}{\text{minimize}} \quad & J(\mathbf{K}) := \text{trace}(P_{\mathbf{K}}) \\ \text{subject to} \quad & (A - BK_N \dots K_1)^\top P_K \\ & + P_K(A - BK_N \dots K_1) \\ & + (K_N \dots K_1)^\top R(K_N \dots K_1) \\ & + Q = 0 \end{aligned} \tag{46}$$

where A, B, Q and R are as in (37), and $\mathbf{K} \in \mathscr{K}$ is a notation overload for $\mathbf{K} \in \{\overline{\mathbf{K}} \mid K_N \dots K_1 \in \mathscr{K}\}$. The policy optimization (gradient flow) for such problem is the collection of N vector fields of the form

$$\dot{K}_i = -\nabla_{K_i} J(\mathbf{K}), \tag{47}$$

for $i = 1, \dots, N$. In our previous work [71], [72], we have established the following properties for this problem

Lemma 4. *For the overparameterized continuous-time LQR policy optimization it can be shown that*

1) *the cost J as a function of the layer weights $(K_1, \dots, K_N)$ does not satisfy a sgPŁI nor is it proper (coercive);*
2) *there is one continuous, path-connected, and non-compact set $\mathcal{T}$ of solutions for (46);*
3) *there are up to 2^p (where $p = \text{rank}K^*$) disjoint sets (each itself continuous, path connected, and non-compact, with exception of one, at the origin of the state-space) of critical points, i.e. points $(K_1, \dots, K_N)$ such that $\nabla_{K_i} J(K_1, \dots, K_N) = 0$ for all $i = 1, \dots, N$;*
4) *all local minima of the problem are a global minimum;*
5) *all solutions initialized at a point $\mathbf{K} \in \mathscr{K}$ converge to a critical point of the cost function;*
6) *if $N = 2$, all initializations but a set of measure-zero converge to a global minimum of the cost function.*

Notice that from a purely qualitatively perspective, employing an overparameterized formulation for the LQR policy optimization is detrimental to the optimization landscape of the problem. The problem which had only a single critical point and global convergence guarantees to the optimum, now has multiple critical points and a non-coercive cost. Although the almost-everywhere convergence to the optimum guarantee mitigates many of these concerns, at this point there is no clear advantage to an overparameterized formulation.

Some possible advantages become clearer if one considers existing literature results on accelerated convergence of overparameterized formulations [73]–[75]. We next show, for synthetic data, that overparameterization does, indeed accelerate the convergence in some scenarios, possibly circumventing the subexponential convergence rate of the solution for initializations far away from the optimum.

4) Numerical simulations on synthetic data: To conclude the discussions from this section of this tutorial paper, we present some numerical simulations demonstrating the subexponential convergence behavior of the continuous-time policy optimization LQR, and the subsequent acceleration that happens when one models the policy through a LFFNN.

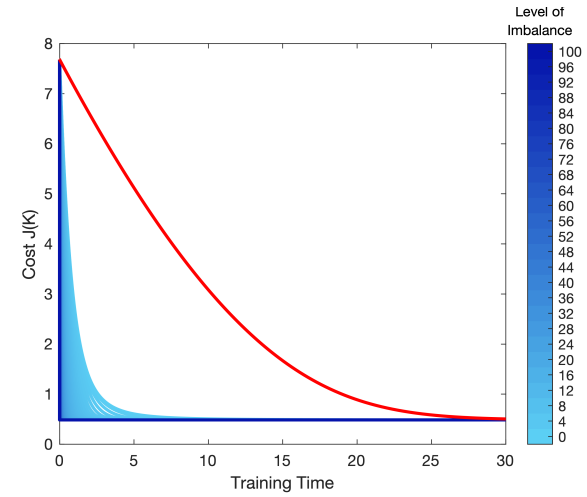
The simulations are done for a single hidden-layer linear neural network with 10 neurons, and for the following values of A and B

$$A = -\begin{bmatrix} 5.2373 & 0.3452 & 0.6653 & 0.6715 & 0.3288 \\ 0.3452 & 5.4889 & 0.8060 & 0.3889 & 0.5584 \\ 0.6653 & 0.8060 & 5.0377 & 0.5735 & 0.5100 \\ 0.6715 & 0.3889 & 0.5735 & 5.3354 & 0.6667 \\ 0.3288 & 0.5584 & 0.5100 & 0.6667 & 5.4942 \end{bmatrix} \tag{48}$$

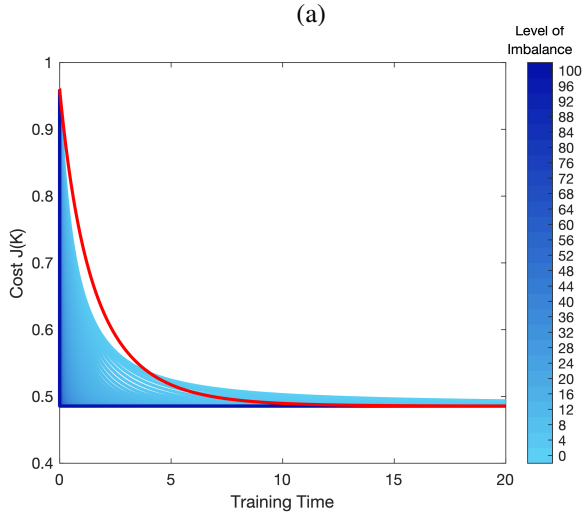
$$B^\top = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{49}$$

We pick an initialization K_0 for the non-overparameterized solution, and select multiple $K_{1,0}$ and $K_{2,0}$ such that $K_{2,0}K_{1,0} = K_0$, resulting in multiple different overparameterized solutions. The overparameterized initializations are selected to have different levels of "imbalance" in some sense, and for a complete overview of the simulation setup, refer to [71].

Notice from Fig. 5a that for, the chosen initialization, the LQR policy-optimization solution has a linear-exponential profile, *i.e.* it decreases linearly while the feedback matrix is far from the optimum, and exponentially when it is near it.



(a)



(b)

Fig. 5. Simulations for the policy optimization solution for the LQR. Red curves are non-overparameterized solutions and the blue curves are the overparameterized ones. In (a) the initialization was picked so that all overparameterized solutions would converge to the optimum while remaining far away from any saddles, while in (b) they were picked so that some overparameterized solutions would pass arbitrarily close to one of the saddle points.

In the same plot, notice that the overparameterized solutions all converge much quicker in comparison.

Alternatively, in Fig. 5b, some of the overparameterized solutions start converging faster than the non-overparameterized ones, but then slow-down as they approach the orbit of a saddle introduced by overparameterization. These two sets of simulations illustrate: the sub-exponential convergence profile of the LQR for some initializations, due to it not satisfying a gPŁI; and both advantages and disadvantages of an overparameterized formulation for the convergence of solutions.

It becomes clear, then, that even the simple optimal linear state feedback problem needs to be tackled with some care by gradient-based methods, even if using common tools from neural-networks literature, such as overparameterization. Furthermore, one might be interested in investigating how the distinction between the continuous- and discrete-time LQR cost functions influence on the convergence rate of Newton-based or natural gradient methods, or of related problem formulation such as a sparse state-feedback through a Lasso regularization. The definitions and results presented serve as basis from which to tackle such problems.

B. Optimal Filtering for Unknown Systems with Transformers

Filtering and prediction are fundamental problems in control systems. For linear systems with Gaussian noise and known dynamics, the celebrated Kalman filter [76] is optimal. For nonlinear systems, while not necessarily optimal, an extended Kalman filter [77] is commonly used because of its ease of implementation. For complicated noise distributions, particle filters [78] provide a way to propagate uncertainty through complex dynamics via sampling. When the system dynamics are unknown, adaptive filters [79] or learning-based variants of Kalman filters [80], [81] are proposed. Motivated by the recent advances in general purpose artificial intelligence models based on transformers [82], in this section, we revisit the filtering and predictions problems and investigate if transformer models can learn a general purpose output predictor that can work with potentially unknown, nonlinear dynamics with correlated non-Gaussian noise. In particular, we summarize some of the positive and negative results from our earlier work in [83]. More empirical work on the use of transformers in filtering and control can be found in [84], [85].

1) Problem Setup: We assume we have an unknown target system $\mathcal{S}_0$ of the form

$$\mathcal{S}_0 : \begin{cases} \mathbf{x}_{0,t+1} = f_0(\mathbf{x}_{0,t}) + \mathbf{w}_{0,t+1} \\ \mathbf{y}_{0,t} = g_0(\mathbf{x}_{0,t}) + \mathbf{v}_{0,t}, \end{cases} \tag{50}$$

where $\mathbf{x}_{0,t} \in \mathbb{R}^n$ and $\mathbf{y}_{0,t} \in \mathbb{R}^m$ are the state and output at time t in the ith system; $f_0(\cdot)$ and $g_0(\cdot)$ are the state dynamics and output functions with $f_0(0) = 0$ and $g_0(0) = 0$; $\mathbf{w}_{0,t} \sim \mathcal{N}(0, \sigma^2_{\mathbf{w},0}\mathbf{I}_n)$ and $\mathbf{v}_{0,t} \sim \mathcal{N}(0, \sigma^2_{\mathbf{v},0}\mathbf{I}_m)$ are the process and output noise, which are mutually independent for all t. For simplicity, it is assumed that the initial state $\mathbf{x}_{0,0} = 0$.

The target system $\mathcal{S}_0$ is assumed to be drawn from some distribution $\mathcal{D}_{sys}$ of systems. At training time, we have access to trajectories generated by M source systems $\{\mathcal{S}_i\}_{i=1}^M$ drawn from the same distribution $\mathcal{D}_{sys}$ and have the same form as the target system:

$$\mathcal{S}_i : \begin{cases} \mathbf{x}_{i,t+1} = f_i(\mathbf{x}_{i,t}) + \mathbf{w}_{i,t+1} \\ \mathbf{y}_{i,t} = g_i(\mathbf{x}_{i,t}) + \mathbf{v}_{i,t}, \end{cases} \tag{51}$$

where state, input, output, and noise terms are defined similarly. We further assume the process and output noise are mutually independent for all i. We emphasize that the target system $\mathcal{S}_0$, while drawn from the same distribution $\mathcal{D}_{sys}$, does not have to be contained within the set of source systems.

We assume that there exists a constant $L_g > 0$ such

that for any i, $\mathbf{x}, \mathbf{x}'$, $\|g_i(\mathbf{x}) - g_i(\mathbf{x}')\| \leq L_g\|\mathbf{x} - \mathbf{x}'\|$. Let $\sigma_{\mathbf{w}} := \max_i \sigma_{\mathbf{w},i}$ and $\sigma_{\mathbf{v}} := \max_i \sigma_{\mathbf{v},i}$. Furthermore, we assume these systems satisfy the following stability condition.

Assumption 2 (Stability)**.** *Let $f_i^{(t)}(\cdot,\cdot)$ denote the t-step state evolution function such that $f_i^{(t)}(\mathbf{x}_{i,\tau}, \mathbf{w}_{i,\tau+1:\tau+t}) = \mathbf{x}_{i,\tau+t}$ for all $\tau \in \mathbb{N}$. Then, there exists constants $\rho \in [0,1)$ and $C_\rho > 0$ such that for any system i and time step t, for any $\mathbf{x}, \mathbf{x}'$ and noise sequence $\mathcal{W} := \{\mathbf{w}_{(\tau)}\}_{\tau\in[t]}$, we have*

$$\|f_i^{(t)}(\mathbf{x}, \mathcal{W}) - f_i^{(t)}(\mathbf{x}', \mathcal{W})\| \leq C_\rho \rho^t \|\mathbf{x} - \mathbf{x}'\|. \tag{52}$$

We define the notation $L_\rho := \frac{C_\rho}{1-\rho}$. When the class of dynamical systems we are sampling from consists of linear systems with $f_i(\mathbf{x}) = \mathbf{A}_i\mathbf{x}$, then Assumption 2 is satisfied when the spectral radius $\rho(\mathbf{A}_i) < 1$ for all i. It is also satisfied by systems that are contracting [86] or exponentially incrementally input-to-state stable [87] with input $\mathbf{w}$.

We seek to predict the output of the target system $\mathcal{S}_0$ using a transformer [82], which is a deep sequence model $\texttt{TF}_\theta(\cdot)$ that maps system output sequences $\mathcal{Y}_t := \mathbf{y}_{0:t}$ to $\hat{\mathbf{y}}_{t+1} := \texttt{TF}_\theta(\mathcal{Y}_t)$, an estimation of the true output $\mathbf{y}_{t+1}$ at time $t+1$. The trainable parameters of the transformer are denoted by $\theta \in \Theta$ for some parameter set Θ. The transformer structure allows the sequence length t to be varying.

2) Learning the Meta-Output-Predictor: Assuming we have access to M length-T output trajectories $\{\mathbf{y}_{i,0:T}\}_{i=1}^M$ generated by each of the M source systems, the goal in this work is to train a transformer model that, at each time t, can predict the output $\mathbf{y}_{0,t+1}$ of the target system S_0 only using the past outputs $\mathbf{y}_{0,0:t}$. Let $\mathcal{Y}_{i,t} := \mathbf{y}_{i,0:t}$ denote the outputs up to time t, which is also known as the *prompt* (to predict $\mathbf{y}_{i,t+1}$), then the transformer is trained by solving

$$\widehat{\texttt{TF}} = \arg\min_{\texttt{TF}\in\mathcal{A}} \frac{1}{MT}\sum_{i=1}^{M}\sum_{t=1}^{T} \ell(\mathbf{y}_{i,t}, \texttt{TF}(\mathcal{Y}_{i,t-1})), \tag{53}$$

where $\mathcal{A} := \{\texttt{TF}_\theta : \theta \in \Theta\}$ and $\ell(\cdot,\cdot) \geq 0$ is the loss function. That is, we solve for the $\theta \in \Theta$ that minimizes the so called emprical loss in (53). To apply $\widehat{\texttt{TF}}(\cdot)$ to the target systems $\mathcal{S}_0$, we simply take $\widehat{\texttt{TF}}(\mathcal{Y}_{0,t})$ as the prediction for $\mathbf{y}_{0,t+1}$.

Our training scheme is depicted in Fig. 6. Training a model as in (53) is related to meta-learning since the transformer is trained using a diverse set of source systems, akin to multiple tasks. Hence, we refer to the obtained transformer $\widehat{\texttt{TF}}$ as meta-output-predictor (MOP). This training mechanism enables the transformer to learn in-context, that is, when faced with a target system at run-time, the transformer can learn to predict the output of the target system at run-time. As a result, similar training schemes are shown to have good generalization performance to unseen but related inputs (target systems, in our context).

3) Performance Guarantees: We first state the following fact that follows from [88, Lemma B.5], see also [83].

Fact 1 (Transformer Robustness)**.** *Consider a trajectory $\{\mathbf{y}_{0:t}\}$ generated by some system i under the noise sequence $\{\mathbf{w}_{0:t}, \mathbf{v}_{0:t}\}$. Let $\{\mathbf{y}'_{0:t}\}$ denote another trajectory under the same noise except that at time τ, $\{\mathbf{w}_\tau, \mathbf{v}_\tau\}$ is replaced by $\{\mathbf{w}'_\tau, \mathbf{v}'_\tau\}$. Let $\mathcal{Y}_t := \mathbf{y}_{0:t}$ and $\mathcal{Y}'_t := \mathbf{y}'_{0:t}$. Suppose the loss function $\ell(\mathbf{y},\cdot)$ is L_ℓ-Lipschitz. Let $\mathcal{B} := \cap_{j=0}^t\{\|\mathbf{w}_j\| \leq \bar{w}, \|\mathbf{v}_j\| \leq \bar{v}\}$ for some $\bar{w}, \bar{v} \geq 0$ and $\mathbb{E}'[\cdot] := \mathbb{E}[\cdot \mid \mathcal{B}]$. Then, for a large class of transformer architectures $\mathcal{A}$, there exist constants $K \geq 0$ such that for any system i, any t and $\{\mathbf{w}_{0:t-1}, \mathbf{v}_{0:t-1}\}$, any $\tau < t$ and $\{\mathbf{w}'_\tau, \mathbf{v}'_\tau\}$, and any $\texttt{TF} \in \mathcal{A}$, we have*

$$\mathbb{E}'_{\mathbf{w}_t,\mathbf{v}_t}\left[\left|\ell(\mathbf{y}_t, \texttt{TF}(\mathcal{Y}_{t-1})) - \ell(\mathbf{y}_t, \texttt{TF}(\mathcal{Y}'_{t-1}))\right|\right] \leq \frac{KL_\ell}{t-\tau}\sum_{j=\tau}^{t-1}\|\mathbf{y}_j - \mathbf{y}'_j\|.$$

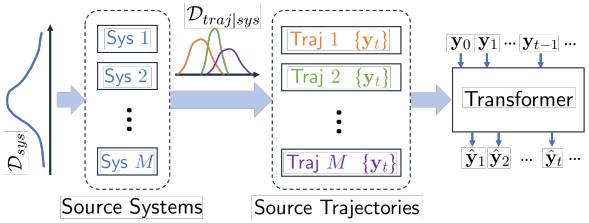


Fig. 6. Training a transformer for dynamical system output prediction

In what follows we assume $\mathcal{A}$ satisfies the above fact. For a transformer $\texttt{TF} \in \mathcal{A}$, we define the following risk to evaluate its performance on the target system S_0 over the time horizon T

$$\mathcal{L}(\texttt{TF}) := \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\ell(\mathbf{y}_{0,t}, \texttt{TF}(\mathcal{Y}_{0,t-1}))\big], \tag{54}$$

where the expectation is over the target system S_0 and noise terms $\{\mathbf{w}_{0,t}, \mathbf{v}_{0,t}\}$. Let $\texttt{TF}^\star \in \mathcal{A}$ denote an optimal transformer that minimizes $\mathcal{L}(\texttt{TF})$. Define the excess risk for $\widehat{\texttt{TF}}$ obtained via minimizing the loss in (53) as

$$\text{Risk}(\widehat{\texttt{TF}}) := \mathcal{L}(\widehat{\texttt{TF}}) - \mathcal{L}(\texttt{TF}^\star). \tag{55}$$

Then, we have the following performance guarantees on $\widehat{\texttt{TF}}$.

Theorem 5. *Suppose Assumption 2 holds, and the loss function $\ell(\mathbf{y},\cdot)$ is L_ℓ-Lipschitz and $\ell(\cdot,\cdot) \leq B$ for some $B \geq 0$. Then, when $MT \geq 3\max(\sqrt{n}, \sqrt{m})$, for all $\epsilon > 0$, with probability at least $1-\delta$,*

$$\textit{Risk}(\widehat{\texttt{TF}}) \leq 12B\delta + 4L_\ell\epsilon + \bar{B}\sqrt{\frac{\log(4\mathcal{N}(\mathcal{A},\mu,\epsilon')/\delta)}{cMT}},$$

where $\bar{B} := 2B + 7KL_\ell\big(L_gL_\rho\sigma_{\mathbf{w}} + \sigma_{\mathbf{v}}\big)\sqrt{\log(4MT/\delta)}\log(T)$; $\epsilon' := \epsilon/((\sigma_{\mathbf{w}} + \sigma_{\mathbf{v}})\sqrt{\log(4MT/\delta)})$; c is some absolute constant; $\mathcal{N}(\cdot,\cdot,\cdot)$ is the covering number of the set $\mathcal{A}$ with respect to a distance metric μ defined on this set.[4]

For fixed failure probability δ and distance ϵ, the upper bound decays with rate $1/\sqrt{MT}$. When the transformer mapping is Lipschitz, the covering number term can be upper bounded by $\log\mathcal{N}(\mathcal{A},\mu,\epsilon') \leq \mathcal{O}(n_\Theta\log(\bar{\Theta}\sqrt{n_\Theta}/\epsilon'))$, where n_Θ and $\bar{\Theta}$ are respectively the dimension and diameter of the

[4]See definitions 1 and 2 in [83] for covering number and the specific distance metric appropriate for this case.

transformer parameter set Θ.

This theorem characterizes how the transformer learned with the algorithm proposed in section V-B.2 performs with respect to the best transformer in the set $\mathcal{A}$ of transformers. One interesting question is whether this set includes any optimal filters. As shown in [89], the elements of this set can indeed represent Kalman filters.

4) Numerical Examples: In this section, we present the experimental results for the transformer-based MOP in different scenarios. In each scenario, during the training, we fix the number of source systems $M = 20000$ and training trajectory length $T = 50$. To evaluate the performance of MOP on different unseen test systems, for each experimental setup, we randomly generate $N = 1000$ systems and record the prediction error $\|\hat{\mathbf{y}}_t - \mathbf{y}_t\|$ over trajectories each with length $T = 50$, where $\hat{\mathbf{y}}_t$ denotes the prediction for $\mathbf{y}_t$. We use GPT-2 [90] architecture with 12 layers, 8 attention heads and 256 embedding dimensions. In each experimental setup, the transformer model is trained for 10000 training steps with batch size 64. The ℓ_2-norm is selected as the training loss function. The code we use to produce the figures and execute our algorithm can be accessed at `https://github.com/haldunbalim/Meta-Output-Predictor`

a) Linear Systems: We first consider the simplest setting with linear systems and i.i.d. Gaussian noise, i.e., $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ and $g(\mathbf{x}) = \mathbf{C}\mathbf{x}$ in (51). The state and output dimensions are $n = 10$, $m = 5$, respectively. For each source and test system, we generate matrix $\mathbf{A}$ with entries sampled uniformly between $[0, 1]$, then scaling so that the largest eigenvalue is 0.95. The $\mathbf{C}$ matrix is generated with entries sampled uniformly between $[0, 1]$. The noise covariance are $\sigma_{\mathbf{w}}^2 = 0.01$ and $\sigma_{\mathbf{v}}^2 = 0.01$. Kalman filter and linear autoregressive predictor are used as baselines, where the latter is given by $\hat{\mathbf{y}}_{t+1} = \boldsymbol{\alpha}_1\mathbf{y}_t + \boldsymbol{\alpha}_2\mathbf{y}_{t-1}$ and the matrix parameters $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ are updated in an online fashion using the recursive least squares (RLS) with initial covariance taken to be identity. The results are presented in Fig. 7(a). We see that after roughly 20 steps, MOP eventually matches the performance of Kalman filter. This lag is because the transformer needs to collect certain amount (in the order of $(n+m)$) of data to implicitly learn the system dynamics, while Kalman filter, having access to the exact dynamics, reaches optimality immediately.

In the next experiment, we consider the case where the noise process $\{\mathbf{w}_t\}$ is non-i.i.d. Specifically, we let $\mathbf{w}_t = \sum_{t'=t-4}^{t} \boldsymbol{\eta}_{t'}$ and $\mathbf{v}_t = \sum_{t'=t-4}^{t} \boldsymbol{\gamma}_{t'}$ where $\boldsymbol{\eta}_t \overset{i.i.d.}{\sim} \mathcal{N}(0, 0.01)$ and $\boldsymbol{\gamma}_t \overset{i.i.d.}{\sim} \mathcal{N}(0, 0.01)$. The Kalman filter is designed disregarding this fact and simply using the variances of $\mathbf{w}_t$ and $\mathbf{v}_t$ for prediction. Fig. 7(b) shows the results for this case. We can observe the advantage of MOP over Kalman filter as Kalman filter is no longer optimal whereas MOP has learned the non-i.i.d noise prior during training.

Next, we evaluate the ability of MOP to adapt to runtime changes in the dynamics. Specifically, when generating the test trajectories, we change the underlying dynamics to a randomly generated new one at time $t = T/2 = 50$. The results are presented in Fig. 7(c). We see that when dynamics changes occur, there are sudden jumps in prediction error for both MOP and the Kalman filter; as we collect more data from the new dynamics, MOP quickly adapts, and achieves the same performance as before at around $t = 100$. The convergence of MOP after dynamics changes is much slower than the one at the beginning because the prompt always contains data from the original system.

b) Planar Quadrotor Systems: We consider the underactuated 6D planar quadrotor systems as in [91] with discrete-time dynamics:

$$\begin{aligned}\mathbf{x}_{t+1} &= f_{(m,l,J)}(\mathbf{x}_t, \mathbf{u}_t) + w_t \\ y_t &= \mathbf{C}\mathbf{x}_t + v_t.\end{aligned}$$

The mass, length and moment of inertia parameters (m, l, J) are chosen uniformly from $[0.5, 2]$. For each system a trajectory is generated by randomly sampled actions. The noise w, v are sampled from $N(0, 0.01)$. The matrix $\mathbf{C} \in \mathbb{R}^{3\times 6}$ has elements uniformly sampled in $[0, 1]$. The results are provided in Fig. 8. We see that MOP significantly outperforms the extended Kalman filter.

c) Effects of Distribution Shifts: In this example, we demonstrate that the performance of MOP degrades if the training distribution is different from the test distribution. In particular, using the same class of systems in Section V-B.4.a, we train the MOP with noise covariances $\sigma_{\mathbf{w}}^2 = 0.1\mathbf{I}_n$, $\sigma_{\mathbf{v}}^2 = 0.1\mathbf{I}_m$ and test on systems subject to a different noise covariance. As shown in Fig. 9, MOP's performance degrades when the target systems are subject to a different noise distribution, especially when the noise covariance increases.

VI. Conclusions

During the past few years there has been substantial research, both in the control and machine learning communities, seeking to design controllers and filters directly from data. The present tutorial paper seeks to provide a key to the relevant literature and illustrate some of the techniques developed in both communities.

References

[1] S. Formentin, K. Van Heusden, and A. Karimi, "A comparison of model-based and data-driven controller tuning," *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 10, pp. 882–897, 2014.

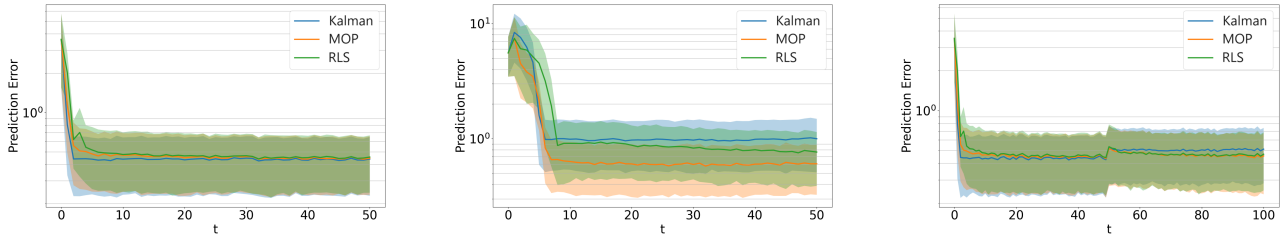
[2] H. J. van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4753–4768, 2020.

[3] M. C. Campi, A. Lecchini, and S. M. Savaresi, "Virtual reference feedback tuning: a direct method for the design of feedback controllers," *Automatica*, vol. 38, no. 8, pp. 1337–1346, 2002.

[4] A. S. Bazanella, L. Campestrini, and D. Eckhard, *Data-driven controller design: the H2 approach*. Springer Science & Business Media, 2011.

[5] A. Karimi, L. Mišković, and D. Bonvin, "Iterative correlation-based controller tuning," *International journal of adaptive control and signal processing*, vol. 18, no. 8, pp. 645–664, 2004.

[6] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: theory and applications," *IEEE control systems magazine*, vol. 18, no. 4, pp. 26–41, 1998.



(a) (b) (c)

Fig. 7. Output predictions for linear systems: (a) with i.i.d. Gaussian noise; (b) with colored (non-i.i.d.) noise; (c) with dynamics changes at $t = T/2$. MOP performs as well as or better than Kalman filter even though it does not have access to the system dynamics.

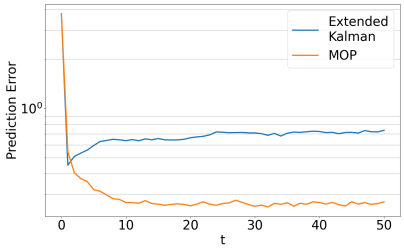


Fig. 8. Output predictions for planar quadrotor systems.

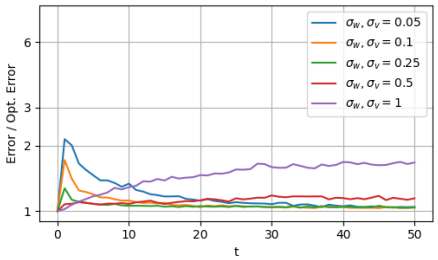


Fig. 9. Performance of MOP compared to Kalman Filter when noise level in test is different than train.

[7] K. Van Heusden, A. Karimi, and D. Bonvin, "Data-driven model reference control with asymptotically guaranteed stability," *International Journal of Adaptive Control and Signal Processing*, vol. 25, no. 4, pp. 331–351, 2011.

[8] J. M. Lee and J. H. Lee, "Approximate dynamic programming-based approaches for input–output data-driven control of nonlinear processes," *Automatica*, vol. 41, no. 7, pp. 1281–1288, 2005.

[9] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2226–2236, 2011.

[10] H. Zhang, H. Jiang, Y. Luo, and G. Xiao, "Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4091–4100, 2016.

[11] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of koopman eigenfunctions for control," *arXiv preprint arXiv:1707.01146*, 2017.

[12] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, pp. 1–10, 2018.

[13] T. Dai and M. Sznaier, "Data Driven Robust Superstable Control of Switched Systems," *IFAC-PapersOnLine*, vol. 51, no. 25, pp. 402–408, 2018.

[14] ——, "A moments based approach to designing mimo data driven controllers for switched systems," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 5652–5657.

[15] C. De Persis and P. Tesi, "Formulas for Data-Driven Control: Stabilization, Optimality, and Robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2019.

[16] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[17] H. J. Vanwaarde, M. K. Camlibel, and M. Mesbahi, "From noisy data to feedback controllers: non-conservative design via a matrix s-lemma," *IEEE Transactions on Automatic Control*, 2020.

[18] J. Berberich, A. Koch, C. W. Scherer, and F. Allgöwer, "Robust data-driven state-feedback design," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1532–1538.

[19] A. Bisoffi, C. De Persis, and P. Tesi, "Data-driven control via Petersen's lemma," *arXiv preprint arXiv:2109.12175*, 2021.

[20] J. Berberich, C. W. Scherer, and F. Allgöwer, "Combining prior knowledge and data for robust controller design," *IEEE Transactions on Automatic Control*, 2022.

[21] M. Bianchi, S. Grammatico, and J. Cortés, "Data-driven stabilization of switched and constrained linear systems," *arXiv preprint arXiv:2208.11392*, 2022.

[22] J. Miller, T. Dai, and M. Sznaier, "Data-driven superstabilizing control under quadratically-bounded errors-in-variables noise," *IEEE Control Systems Letters*, vol. 8, pp. 1655–1660, 2024.

[23] J. Berberich and F. Allgöwer, "An overview of systems-theoretic guarantees in data-driven model predictive control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, 2024.

[24] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 307–312.

[25] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2020.

[26] M. Attar and W. Lucia, "Data-driven robust backward reachable sets for set-theoretic model predictive control," *IEEE Control Systems Letters*, vol. 7, pp. 2305–2310, 2023.

[27] R. Li, J. W. Simpson-Porco, and S. L. Smith, "Stochastic data-driven predictive control with equivalence to stochastic mpc," *arXiv preprint arXiv:2312.15177*, 2023.

[28] Y. Xie, J. Berberich, and F. Allgöwer, "Data-driven min-max mpc for linear systems," in *2024 American Control Conference (ACC)*. IEEE, 2024, pp. 3184–3189.

[29] J. Zheng, S. Kiani, M. Sznaier, and C. M. Lagoa, "Robust data-driven receding horizon control," in *Proc. 2025 IFAC Symposium on Robust Control*, 2025, to appear.

[30] G. Bontempi and M. Birattari, "From linearization to lazy learning: A survey of divide-and-conquer techniques for nonlinear control (invited paper)," *Int. J. of Computational Cognition*, vol. 3, 01 2004.

[31] T. Dai and M. Sznaier, "A semi-algebraic optimization approach to data-driven control of continuous-time nonlinear systems," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 487–492, 2020.

[32] M. Guo, C. De Persis, and P. Tesi, "Learning control for polynomial systems using sum of squares relaxations," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 2436–2441.

[33] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*, 2020, pp. 708–717.

[34] J. Zheng, T. Dai, J. Miller, and M. Sznaier, "Robust Data-Driven Safe Control Using Density Functions," *IEEE Control Systems Letters*, vol. 7, pp. 2611–2616, 2023.

[35] A. Luppi, A. Bisoffi, C. De Persis, and P. Tesi, "Data-driven design of safe control for polynomial systems," *European Journal of Control*, vol. 75, p. 100914, 2024.

[36] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[37] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Robust Adaptive Control Barrier Functions: An Adaptive and Data-Driven Approach to Safety," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1031–1036, 2021.

[38] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

[39] Z. Hu, C. D. Persis, and P. Tesi, "Enforcing contraction via data," 2024. [Online]. Available: https://arxiv.org/abs/2401.07819

[40] A. Oliveira, J. Zheng, and M. Sznaier, "Convex data-driven contraction with riemannian metrics," 2025. [Online]. Available: https://arxiv.org/abs/2412.20283

[41] R. K. Cosner, I. Sadalski, J. K. Woo, P. Culbertson, and A. D. Ames, "Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions," 2023.

[42] A. Salamati, A. Lavaei, S. Soudjani, and M. Zamani, "Data-driven verification and synthesis of stochastic systems via barrier certificates," *Automatica*, vol. 159, p. 111323, 2024.

[43] U. Rosolia and F. Borrelli, "Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1883–1896, 2018.

[44] J. Zheng, J. Miller, and M. Sznaier, "Data-driven safe control of discrete-time non-linear systems," *IEEE Control Systems Letters*, vol. 8, pp. 1553–1558, 2024.

[45] T. Dai and M. Sznaier, "Data-driven quadratic stabilization and LQR control of LTI systems," *Automatica*, vol. 153, p. 111041, 2023.

[46] E. Feron, V. Balakrishnan, S. Boyd, and L. E. Ghaoui, "Numerical methods for h2 related problems," *American Control Conference*, pp. 2921–2922, 1992.

[47] C. W. Scherer and C. W. Hol, "Matrix Sum-of-Squares Relaxations for Robust Semi-Definite Programs," *Mathematical Programming*, vol. 107, no. 1, pp. 189–211, 2006.

[48] M. C. De Oliveira, J. Bernussou, and J. C. Geromel, "A new discrete-time robust stability condition," *Systems & control letters*, vol. 37, no. 4, pp. 261–265, 1999.

[49] A. Rantzer, "A dual to lyapunov's stability theorem," *Systems & Control Letters*, vol. 42, no. 3, pp. 161–168, 2001.

[50] S. Prajna, P. A. Parrilo, and A. Rantzer, "Nonlinear control synthesis by convex optimization," *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 310–314, 2004.

[51] D. Henrion, S. Tarbouriech, and V. Kucera, "Control of linear systems subject to input constraints: a polynomial approach. part i-siso plants," in *IEEE CONFERENCE ON DECISION AND CONTROL*, vol. 3. Citeseer, 1999, pp. 2774–2774.

[52] J.-B. Lasserre, *Moments, positive polynomials and their applications*. World Scientific, 2009, vol. 1.

[53] J. Zhen, F. J. de Ruiter, E. Roos, and D. den Hertog, "Robust optimization for models with uncertain second-order cone and semidefinite programming constraints," *INFORMS Journal on Computing*, vol. 34, no. 1, pp. 196–210, 2022.

[54] M. Bozdag, A. Honarpisheh, and M. Sznaier, "Safe control for pursuit-evasion with density functions," 2025.

[55] Y. Xie, J. Berberich, and F. Allgöwer, "Data-driven min-max mpc for linear systems: Robustness and adaptation," *arXiv:2404.19096*, 2024.

[56] Y. Xie, J. Berberich, F. Brändle, and F. Allgöwer, "Data-driven min-max MPC for LPV systems with unknown scheduling signal," in *Proc. European Control Conference (ECC), Preprint:arXiv:2411.05624*, 2024.

[57] Y. Xie, J. Berberich, R. Strässer, and F. Allgöwer, "Bilinear data-driven min-max mpc: Designing rational controllers via sum-of-squares optimization," *arXiv:2504.04870*, 2025.

[58] T. Martin, T. B. Schön, and F. Allgöwer, "Guarantees for data-driven control of nonlinear systems using semidefinite programming: A survey," *Annual Reviews in Control*, p. 100911, 2023.

[59] R. Tóth, *Modeling and identification of linear parameter-varying systems*. Springer, 2010, vol. 403.

[60] R. R. Mohler, *Bilinear control processes: with applications to engineering, ecology and medicine*. Academic Press, Inc., 1973.

[61] W. Levine and M. Athans, "On the determination of the optimal constant output feedback gains for linear multivariable systems," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 44–48, Feb. 1970. [Online]. Available: http://ieeexplore.ieee.org/document/1099363/

[62] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *International conference on machine learning*. PMLR, 2018, pp. 1467–1476.

[63] H. Mohammadi, A. Zare, M. Soltanolkotabi, and M. R. Jovanovic, "Convergence and Sample Complexity of Gradient Methods for the Model-Free Linear–Quadratic Regulator Problem," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2435–2450, May 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9448427/

[64] B. Hu, K. Zhang, N. Li, M. Mesbahi, M. Fazel, and T. Başar, "Toward a theoretical foundation of policy optimization for learning control policies," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, no. 1, pp. 123–158, 2023.

[65] E. D. Sontag, "Remarks on input to state stability of perturbed gradient flows, motivated by model-free feedback control learning," *Systems & Control Letters*, vol. 161, p. 105138, Mar. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167691122000056

[66] L. Cui, Z.-P. Jiang, and E. D. Sontag, "Small-disturbance input-to-state stability of perturbed gradient flows: Applications to LQR problem," *Systems & Control Letters*, vol. 188, p. 105804, 2024.

[67] I. Fatkhullin and B. Polyak, "Optimizing Static Linear Feedback: Gradient Method," *SIAM Journal on Control and Optimization*, vol. 59, no. 5, pp. 3887–3911, Jan. 2021. [Online]. Available: https://epubs.siam.org/doi/10.1137/20M1329858

[68] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[69] L. Cui, Z.-P. Jiang, and E. D. Sontag, "Small-disturbance input-to-state stability of perturbed gradient flows: Applications to LQR problem," *Systems & Control Letters*, vol. 188, p. 105804, 2024.

[70] A. C. B. de Oliveira, L. Cui, and E. D. Sontag, "Remarks on the polyak-lojasiewicz inequality and the convergence of gradient systems," *arXiv preprint arXiv:2503.23641*, 2025.

[71] A. C. B. de Oliveira, M. Siami, and E. D. Sontag, "Convergence analysis of overparametrized lqr formulations," *arXiv preprint arXiv:2408.15456*, 2024.

[72] ——, "Remarks on the Gradient Training of Linear Neural Network Based Feedback for the LQR Problem," in *2024 63rd IEEE Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 7846–7852.

[73] H. Min, S. Tarmoun, R. Vidal, and E. Mallada, "On the Explicit Role of Initialization on the Convergence and Implicit Bias of Overparametrized Linear Networks," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, Jul. 2021, pp. 7760–7768, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v139/min21c.html

[74] H. Min, R. Vidal, and E. Mallada, "On the convergence of gradient flow on multi-layer linear models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 24 850–24 887.

[75] S. Tarmoun, G. Franca, B. D. Haeffele, and R. Vidal, "Understanding the Dynamics of Gradient Flow in Overparameterized Linear models," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, Jul. 2021, pp. 10 153–10 161, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v139/tarmoun21a.html

[76] R. E. Kalman, "A new approach to recursive filtering and prediction problems," *Trans. ASME*, vol. 82, pp. 35–45, 1960.

[77] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

[78] P. Del Moral, "Nonlinear filtering: Interacting particle resolution," *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, vol. 325, no. 6, pp. 653–658, 1997.

[79] P. S. Diniz, *Adaptive filtering*. Springer, 1997, vol. 4.

[80] E. Hazan, K. Singh, and C. Zhang, "Learning linear dynamical systems via spectral filtering," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[81] A. Tsiamis, N. Matni, and G. Pappas, "Sample complexity of Kalman filtering for unknown systems," in *L4DC*. PMLR, 2020, pp. 435–444.

[82] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, vol. 30, 2017.

[83] Z. Du, H. Balim, S. Oymak, and N. Ozay, "Can transformers learn optimal filtering for unknown systems?" *IEEE Control Systems Letters*, vol. 7, pp. 3525–3530, 2023.

[84] D. Piga, F. Pura, and M. Forgione, "On the adaptation of in-context

learners for system identification," *IFAC-PapersOnLine*, vol. 58, no. 15, pp. 277–282, 2024.
[85] R. Busetto, V. Breschi, M. Forgione, D. Piga, and S. Formentin, "In-context learning of state estimators," *IFAC-PapersOnLine*, vol. 58, no. 15, pp. 145–150, 2024.
[86] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.
[87] D. Angeli, "A Lyapunov approach to incremental stability properties," *IEEE Trans. on Automatic Control*, vol. 47, no. 3, pp. 410–421, 2002.
[88] Y. Li, M. E. Ildiz, D. Papailiopoulos, and S. Oymak, "Transformers as algorithms: Generalization and stability in in-context learning," *International Conference on Machine Learning*, 2023.
[89] G. Goel and P. Bartlett, "Can a transformer represent a kalman filter?" in *6th Annual Learning for Dynamics & Control Conference*. PMLR, 2024, pp. 1502–1512.
[90] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
[91] S. Singh, B. Landry, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust feedback motion planning via contraction theory," *The International Journal of Robotics Research*, 2019.