Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Data-Driven Quadratic Stabilization of Continuous LTI Systems ⋆

Tianyu Dai * Mario Sznaier ** Biel Roig Solvas ***

* *ECE Dept, Northeastern University, Boston, MA, 02115, USA(e-mail: dai.ti@husky.neu.edu)*
** *ECE Dept, Northeastern University, Boston, MA, 02115, USA(e-mail: msznaier@coe.neu.edu)*
*** *ECE Dept, Northeastern University, Boston, MA, 02115, USA(e-mail: biel@ece.neu.edu)*

Abstract: This paper introduces a simple data-driven quadratic stabilization control (DDQSC) method to design a state feedback controller based solely on experimental measurements while avoiding explicitly identifying the plant. Rather, we seek a controller guaranteed to quadratically stabilize all plants that could have possibly generated the observed data. While in principle this leads to a very challenging non-convex robust optimization problem, our main result provides a convex, albeit infinite-dimensional, necessary and sufficient condition for the existence of such a controller and its associated Lyapunov function. In the second part of the paper, we provide a tractable finite-dimensional convex relaxation of this condition and illustrate its effectiveness with several examples.

Keywords: Data-Driven Control, Robust Control, Quadratic Stability, Semi-Definite Programming

1. INTRODUCTION

Robust control of uncertain systems has been well studied during the past decades, resulting in efficient synthesis methods that guarantee the stability of a set of plants, typically described by a nominal plant and bounded uncertainty (see for instance Sánchez Peña and Sznaier (1998); Zhou and Doyle (1998) and references therein). The traditional design procedure is based on first identifying a nominal plant along with an uncertainty description, using for instance control oriented identification methods (Chen and Gu (2000)), followed by a robust controller synthesis step. However, this two-step approach is typically conservative, since the worst-case uncertainty bounds obtained from the identification steps are usually not tight. This conservativeness can be avoided by pursuing a data-driven control approach, rather than model-based control, which avoids the plant identification step and provides an end-to-end control framework, i.e. design the controller directly based on data. Indeed, as shown in (Formentin et al. (2014)), for a not completely known or high-order system, data-driven methods can statistically outperform model-based ones in terms of control cost. In addition, the recent work in (De Persis and Tesi (2019)) validated the fact that the system model can be replaced by data-dependent matrices.

During the past two decades, several data-driven control approaches have been proposed. These methods include virtual reference feedback tuning (Campi et al. (2002), Bazanella et al. (2011)), correlation-based tuning (Karimi et al. (2004)) and iterative feedback tuning (Hjalmarsson et al. (1998)). 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. (Van Heusden et al. (2011)) provides an asymptotical stability criterion but this is only valid when considering infinitely long data sequence. As an alternative, the recent popularity of neural networks motivated several data-driven methods based on adaptive dynamic programming (Lee and Lee (2005), Zhang et al. (2011)), reinforcement learning (Zhang et al. (2016)) and Koopman eigenfunction (Kaiser et al. (2017), Lusch et al. (2018)). While these methods perform well in simulations, they lack rigorous stability certificates and usually require a fair amount of work tuning the hyper-parameters.

Recently (De Persis and Tesi (2019)) proposed an approach that guarantees closed-loop stability for scenarios where the noise is small enough. Alternatively (Dai and Sznaier (2018)) considered the data-driven control problem for discrete switched LTI systems and showed that a controller guaranteed to stabilize the consistency set, i.e. the set of all plants compatible with the observed data can be synthesized using polyhedral control Lyapunov functions (PCLFs). In principle, the same technique can be applied to design data-driven controllers for non-switching systems. However, the entailed computational complexity is non-trivial, since finding the PCLF requires solving a polynomial optimization problem. In addition, the number

⋆ This work was supported in part by NSF grants CNS–1646121, CMMI–1638234, IIS–1814631 and ECCS–1808381; AFOSR grant FA9550-19-1-0005; and the Alert DHS Center of Excellence under Award Number 2013-ST-061-ED0001.

of faces of the corresponding polyhedral level sets must be fixed a-priori, which constrains the regions where the closed-loop can be placed. Motivated by this success in (Dai and Sznaier (2018)), in this paper we pursue an approach that is similar in spirit: searching for a control Lyapunov function and associated controller guaranteed to stabilize the consistency set. However, to avoid the difficulties associated with using polyhedral Lyapunov functions, here we will seek for quadratic ones, that is, our goal is to find a controller that quadratically stabilizes the consistency set, for scenarios where the noise is not necessarily small. The advantages of seeking quadratically stabilizing controllers are two-fold. On one hand, as shown by the main result of this paper, this approach leads to convex necessary and sufficient conditions for the existence of such a controller; on the other, achieving quadratic stability is a desirable property since it provides uniform bounds on the rate of convergence of the trajectories to the origin and their ℓ_2 bounds, for all plants in the consistency set.

The remainder of the paper is organized as follows. In section 2, we state the problem under consideration and provide some background results on the duality necessary to solve it. The main result is given in section 3 where we show that, by exploiting duality, the original non-convex problem can be reformulated, without conservatism, as a convex, albeit infinite-dimensional, Semi-Definite Program (SDP). Section 4 presents a computationally tractable relaxation of this infinite-dimensional problem. Section 5 presents two academic examples that illustrate the advantages of the proposed method over a naive alternative based on explicitly finding a common Lyapunov for all vertices of the consistency set. Finally, Section 6 summarizes our results and points out directions for further research.

2. PRELIMINARIES

2.1 Notation

We use the standard linear algebra notation. $\mathbb{R}$ and $\mathbb{R}^n$ denote the real numbers and the real n-dimensional vector space, respectively. $\mathbf{x} \in \mathbb{R}^n$ is a vector and $\mathbf{X} \in \mathbb{R}^{m\times n}$ is a matrix. $\mathbf{X} \succeq 0$ indicates a positive semi-definite (PSD) matrix. $\mathbf{Tr}(\mathbf{X})$ is the trace of the matrix. $\mathbf{I}$ is the identity matrix of suitable size. $\mathbf{1}$ represents a vector of 1s.

2.2 Background Results

In this section we recall, for ease of reference, some background results and definitions.

Definition 1. (Khargonekar et al. (1990)) An uncertain continuous time system of the form $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$, with $\mathbf{A} \in \mathcal{A}$ is said to be quadratically stable if there exists a $n \times n$ positive definite matrix $\mathbf{P}$ such that, for any $\mathbf{A} \in \mathcal{A}$, $V = \mathbf{x}^T\mathbf{P}\mathbf{x}$ is a Lyapunov function of the system, e.g. the following holds

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} \prec 0, \quad \forall \mathbf{A} \in \mathcal{A} \tag{1}$$

Similarly, a system of the form $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, with uncertain $\mathbf{A} \in \mathcal{A}$ and $\mathbf{B} \in \mathcal{B}$ is said to be quadratically stabilizable if there exists a state feedback controller $\mathbf{u} = \mathbf{K}\mathbf{x}$ such that, for any pairs $(\mathbf{A} \in \mathcal{A}, \mathbf{B} \in \mathcal{B})$ the resulting closed-loop system is quadratically stable.

Theorem of Alternatives. The following result will play a key role in recasting the data-driven quadratic stabilization problem into a tractable form. Given n concave functions $f_i(\mathbf{x})$, consider the following (primal) feasibility problem:

$$\text{Does there exist } \mathbf{x} \text{ such that } f_i(\mathbf{x}) \geq 0,\ i = 1, \ldots, n? \tag{primal}$$

The dual function associated with the primal problem is:

$$g(\boldsymbol{\mu}) = \sup_{\mathbf{x}} \sum_{i=1}^{k} \mu_i f_i(\mathbf{x}) \tag{2}$$

where μ_i are scalars (the Lagrange multiplier). In terms of $g(.)$ the dual problem is:

$$\text{Does there exist } \boldsymbol{\mu} \geq 0 \text{ such that } g(\boldsymbol{\mu}) < 0? \tag{dual}$$

Theorem 1. The primal and dual problems are strong alternatives, that is, exactly one of them is feasible.

The proof can be found for instance in (Boyd and Vandenberghe (2004)), Chapter 5.

2.3 Problem Statement

Throughout the paper, we consider the following controller design problem:

Problem 1. Consider a continuous LTI system:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \boldsymbol{\eta}(t) \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$ are unknown system matrices, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$ denote the state and input vectors, and $\boldsymbol{\eta} \in \mathbb{R}^n$ denotes ℓ_∞ bounded noise [1], with $||\boldsymbol{\eta}||_\infty \leq \epsilon$. Given measurements $\dot{\mathbf{x}}(t_k)$, $\mathbf{x}(t_k)$, $\mathbf{u}(t_k)$, $k = 1, \ldots, n_s$ representing the sample index, the goal is to find a state feedback controller $\mathbf{u} = \mathbf{K}\mathbf{x}$ guaranteed to quadratically stabilize all pairs $(\mathbf{A}, \mathbf{B})$ that could have generated the observed data.

Note that each measurement $(\dot{\mathbf{x}}(t_k), \mathbf{x}(t_k), \mathbf{u}(t_k))$ yields $2n$ polytopic constraints on the elements of $(\mathbf{A}, \mathbf{B})$:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{A} & -\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t_k) \\ \mathbf{u}(t_k) \end{bmatrix} \leq \begin{bmatrix} \epsilon\mathbf{1} + \dot{\mathbf{x}}(t_k) \\ \epsilon\mathbf{1} - \dot{\mathbf{x}}(t_k) \end{bmatrix} := \mathbf{d}_k \tag{4}$$

Thus, Problem 1 can be recast into the following robust optimization form: Find a positive definite matrix $\mathbf{P}$ and a controller $\mathbf{K}$ such that :

$$\begin{gathered} (\mathbf{A} + \mathbf{B}\mathbf{K})^T\mathbf{P} + \mathbf{P}(\mathbf{A} + \mathbf{B}\mathbf{K}) \prec 0 \\ \text{for all } (\mathbf{A}, \mathbf{B}) \text{ that satisfy (4)} \end{gathered} \tag{5}$$

Note that (5) is bilinear in $\mathbf{K}, \mathbf{P}$. However, using the standard technique of pre/post-multiplying by $\mathbf{Y} \doteq \mathbf{P}^{-1}$ and defining $\mathbf{M} \doteq \mathbf{K}\mathbf{Y}$ yields the Linear Matrix Inequality (LMI)

$$\mathbf{Y}\mathbf{A}^T + \mathbf{A}\mathbf{Y} + \mathbf{M}^T\mathbf{B}^T + \mathbf{B}\mathbf{M} \prec 0 \tag{6}$$

Hence the problem above is equivalent to:

Problem 2. Find matrices $\mathbf{Y} \succ 0$ and $\mathbf{M}$ such that (6) holds for all $(\mathbf{A}, \mathbf{B})$ that satisfy (4).

2.4 A Naive Approach: Vertex LMIs.

Since the constraints (4) define a polytope of $(\mathbf{A}, \mathbf{B})$ in $\mathbb{R}^{n^2} \times \mathbb{R}^{nm}$ a naive approach to solving Problem 2 is to

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

simply find the vertices $(\mathbf{A}_i, \mathbf{B}_i) \doteq \mathbf{V}_i$ of this polytope and search for a common solution $(\mathbf{Y}, \mathbf{M})$ to the set of vertex LMIs. However, due to the exponential growth of the number of vertices with the problem dimension and number of samples, coupled with the limitations of existing LMI solvers (at most $\sim 10^3$ variables), this approach is impractical beyond simple toy problems. For instance, even a scenario with $n = m = 3$ and $n_s = 50$ samples yields 12.7×10^6 vertices. In the next section, we provide a tractable alternative to this naive approach, based on the use of duality.

3. A NECESSARY AND SUFFICIENT CONDITION FOR QUADRATIC STABILIZABILTY

In this section we show that Problem 2 can be recast as a convex (albeit infinite-dimensional) optimization by exploiting duality. The first step towards this goal is, for reasons that will be clear below, to rewrite the constraints (4) in the following form:

$$\mathbf{Tr}(\mathbf{A}\mathbf{Z}^x_{i,k} + \mathbf{B}\mathbf{Z}^u_{i,k}) \le d_{i,k} \tag{7}$$

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

$$\begin{aligned} \mathbf{Z}^x_{i,k} &= [\ldots, \pm\mathbf{x}(t_k), \ldots]_{n\times n} \\ \mathbf{Z}^u_{i,k} &= [\ldots, \pm\mathbf{u}(t_k), \ldots]_{m\times n} \end{aligned} \tag{8}$$

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

$$\begin{aligned} \mathbf{Z}^x_{1,k} &= \begin{bmatrix} x_1(t_k) & 0 \\ x_2(t_k) & 0 \end{bmatrix}, \quad \mathbf{Z}^x_{2,k} = \begin{bmatrix} 0 & x_1(t_k) \\ 0 & x_2(t_k) \end{bmatrix}, \\ \mathbf{Z}^x_{3,k} &= \begin{bmatrix} -x_1(t_k) & 0 \\ -x_2(t_k) & 0 \end{bmatrix}, \mathbf{Z}^x_{4,k} = \begin{bmatrix} 0 & -x_1(t_k) \\ 0 & -x_2(t_k) \end{bmatrix} \end{aligned} \tag{9}$$

with similar expressions for $\mathbf{Z}^u_{i,k}$. In terms of these matrices Problem 2 can be recast as:

Problem 3. Find matrices $\mathbf{Y} \succ 0$ and $\mathbf{M}$ such that (6) holds for all $(\mathbf{A}, \mathbf{B})$ that satisfy (7).

The next result shows that this problem can be solved by solving an infinite-dimensional convex optimization.

Theorem 2. Problem 3 is feasible if and only if there exist matrices $\mathbf{Y} \succ 0$, $\mathbf{M}$ and n_s non-negative (vector) functions $\boldsymbol{\mu}_k(\mathbf{x}) \colon \mathbb{R}^n \to \mathbb{R}^{2n}_+$ such that the following conditions hold for all $\mathbf{x} \in \mathbb{R}^n$:

$$\sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mu_{i,k} d_{i,k} < 0 \tag{10}$$

$$2\mathbf{x}\mathbf{x}^T\mathbf{Y} - \sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mu_{i,k}(\mathbf{x})(\mathbf{Z}^x_{i,k})^T = 0 \tag{11}$$

$$2\mathbf{x}\mathbf{x}^T\mathbf{M}^T - \sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mu_{i,k}(\mathbf{x})(\mathbf{Z}^u_{i,k})^T = 0 \tag{12}$$

In order to prove this Theorem, we need the following preliminary result:

Lemma 1. Given a fixed $\mathbf{x} \in \mathbb{R}^n$ and fixed matrices $\mathbf{Y} \in \mathbb{R}^{n\times n}, \mathbf{M} \in \mathbb{R}^{m\times n}$, consider the following feasibility problem in $(\mathbf{A}, \mathbf{B})$:

$$\begin{aligned} &\mathbf{x}^T(\mathbf{Y}\mathbf{A}^T + \mathbf{A}\mathbf{Y} + \mathbf{M}^T\mathbf{B}^T + \mathbf{B}\mathbf{M})\mathbf{x} \ge 0 \\ &\mathbf{Tr}(\mathbf{A}\mathbf{Z}^x_{i,k} + \mathbf{B}\mathbf{Z}^u_{i,k}) \le d_{i,k} \end{aligned} \tag{13}$$

Then (13) is infeasible if and only if there exist n_s non-negative vectors $\boldsymbol{\mu}_k(\mathbf{x}, \mathbf{Y}, \mathbf{M})$ such that:

$$\begin{aligned} &\sum_{k=1}^{n_s} \boldsymbol{\mu}_k^T \mathbf{d}_k < 0 \\ &2\mathbf{x}\mathbf{x}^T\mathbf{Y} - \sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mu_{i,k}(\mathbf{Z}^x_{i,k})^T = 0 \\ &2\mathbf{x}\mathbf{x}^T\mathbf{M}^T - \sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mu_{i,k}(\mathbf{Z}^u_{i,k})^T = 0 \end{aligned} \tag{14}$$

where for notational simplicity we do not denote the explicit dependence of $\mu_{i,k}$ on $\mathbf{x}$, $\mathbf{Y}$ and $\mathbf{M}$.

Proof: Omitted for space reasons follows by showing that the dual problem of (13) is given by:

$$\begin{aligned} g(\boldsymbol{\mu}) = \sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mu_{i,k} d_{i,k} < 0 \\ \text{subject to } \mu_{i,k} \ge 0, \text{ (11) and (12)} \end{aligned} \tag{15}$$

and applying the Theorem of Alternatives to (13) and its dual (15). □

The detailed proof of Theorem 2 is omitted due to the lack of space. Its main idea is to first use Lemma 1 to show that Problem 3 is feasible if there exist matrices $\mathbf{Y} \succ 0$, $\mathbf{M}$ and n_s vectors $\boldsymbol{\mu}_k(\mathbf{x}, \mathbf{Y}, \mathbf{M}) \ge 0$ such that (14) holds for all $\mathbf{x}$. The proof follows by showing that, since we are interested in finding just one feasible solution, then the vector functions $\boldsymbol{\mu}_k$ can be taken to be independent of $\mathbf{Y}$ and $\mathbf{M}$. □

Remark 1. Theorem 2 provides a convex necessary and sufficient condition, in the form of an SDP in $\mathbf{M}, \mathbf{Y}, \boldsymbol{\mu}_k$, for the existence of a state-feedback controller that quadratically stabilizes the set of all pairs $(\mathbf{A}, \mathbf{B})$ consistent with the observed data. However, this SDP is infinite-dimensional since the constraints (10)-(12) must hold for all $\mathbf{x}$.

4. A TRACTABLE RELAXATION

As noted in the previous section, Theorem 2 provides a solution to Problem 2 in the form of an infinite-dimensional SDP. While of theoretical value, this condition is impractical for designing controllers, even if one wants to discretize it by gridding $\mathbf{x}$, due to the limitation of existing SDP solvers. To address this difficulty, in this section we present a computationally tractable, Sum-of-Squares (SoS) based relaxation.

From (11)-(12) it follows that $\mu_{i,k}(\mathbf{x})$ must be an homogeneous function in $\mathbf{x}$ of degree 2. In order to obtain tractable relaxations, we will approximate these functions by second-order SoS polynomials of the form $\mu_{i,k}(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}_{i,k}\mathbf{x}$ leading to the following feasibility problem:

Problem 4. Find matrices $\mathbf{Y} \succ 0$, $\mathbf{M}$ and $\mathbf{Q}_{i,k} \succeq 0$ such that the following conditions hold:

$$\sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mathbf{Q}_{i,k} d_{i,k} \prec 0 \tag{16}$$

$$2\mathbf{x}\mathbf{x}^T\mathbf{Y} - \sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mathbf{x}^T\mathbf{Q}_{i,k}\mathbf{x}(\mathbf{Z}^x_{i,k})^T = 0 \tag{17}$$

$$2\mathbf{x}\mathbf{x}^T\mathbf{M}^T - \sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mathbf{x}^T\mathbf{Q}_{i,k}\mathbf{x}(\mathbf{Z}^u_{i,k})^T = 0 \tag{18}$$

Note that (17)-(18) each defines $\frac{n(n+1)}{2}$ polynomial constraints that can be solved by simply setting the coefficients of each monomial in the (second-order) polynomial to zero. In the sequel, for notational simplicity, we will denote these coefficients as $k_a(\mathbf{Y}, \mathbf{Q}_{i,k}), k_b(\mathbf{M}, \mathbf{Q}_{i,k})$ leading to the following algorithm:

Algorithm 1 DDQSC

Given n_s measurements $\mathbf{x}, \mathbf{u}, \dot{\mathbf{x}}$, a noise bound ϵ and a small number δ, build $\mathbf{Z}^x_{i,k}, \mathbf{Z}^u_{i,k}, d_{i,k}$, $i = 1, ..., 2n$, $k = 1, ..., n_s$.
Solve:
  minimize 0
  subject to

$$\begin{aligned} k_a(\mathbf{Y}, \mathbf{Q}_{i,k}) &= 0 \\ k_b(\mathbf{M}, \mathbf{Q}_{i,k}) &= 0 \\ -\sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mathbf{Q}_{i,k} d_{i,k} &\succeq \delta\mathbf{I} \\ \mathbf{Q}_{i,k} &\succeq 0 \\ \mathbf{Y} &\succeq \delta\mathbf{I} \end{aligned} \tag{19}$$

Remark 2. As indicated before, the algorithm above is a relaxation of the original problem, in the following sense. Existence of a feasible solution provides a certificate of quadratic stabilizability through the Lyapunov function $\mathbf{x}^T\mathbf{Y}^{-1}\mathbf{x}$ and associated controller $\mathbf{K} = \mathbf{M}\mathbf{Y}^{-1}$. On the other hand, infeasibility of the Algorithm does not rule out the existence of solutions to Problem 1 as there could exist $\boldsymbol{\mu}_k(\mathbf{x}) > 0$ outside the SoS set. Nevertheless consistent numerical experience shows that this relaxation works well in practice. In principle, conservatism can be reduced by considering rational functions of the form

$$\mu_{i,k}(\mathbf{x}) = \frac{N_{i,k}(\mathbf{x})}{D_{i,k}(\mathbf{x})}$$
$$\text{degree}(N_{i,k}(\mathbf{x})) - \text{degree}(D_{i,k}(\mathbf{x})) = 2$$

where $N_{i,k}(\mathbf{x}), D_{i,k}(\mathbf{x})$ can have arbitrarily high degree. However, inserting such functions into (11)-(12), will lead to bilinear constraints involving the products of $\mathbf{Y}\mathbf{P}_{i,k}$ and $\mathbf{M}^T\mathbf{P}_{i,k}$. In principle this problem can also be solved using convex optimization by using the polynomial optimization techniques presented in Lasserre (2009). However, the entailed computational complexity prevents using this approach except for small sized problems.

5. SIMULATION RESULTS

In this section, we illustrate the proposed framework with several academic examples. In all cases the data was generated using the MATLAB function ode45.m (Shampine and Reichelt (1997)) and Algorithm 1 was implemented using YALMIP (Löfberg (2004)). For benchmarking purposes, we also applied (when feasible) the Vertex LMI (VLMI) method described in section 2.4. In this case, we obtained all the vertices of the polytope (4) using the Vertex Enumeration (VE) algorithm introduced in (Avis and Jordan (2018)) and then attempted to find a common Lyapunov function to the corresponding LMIs.

5.1 Second-Order System

We first consider data generated by the second-order system:

$$\mathbf{A} = \begin{bmatrix} 0.9367 & 0.7211 \\ 0.3586 & 0.3974 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.3940 & 0.2038 \\ 0.3513 & 0.7954 \end{bmatrix} \tag{20}$$

$\mathbf{A}$ has eigenvalues (1.2426, 0.0915) and hence is unstable. The input and the initial states are uniformly distributed in $[-1, 1]$. The noise is uniformly distributed in $[-\epsilon, \epsilon]$. For the parameters, we selected $\delta = 0.001, \epsilon = 0.1, s = 25$, i.e. 25 samples with 10% noise, obtained by equally sampling within the time interval [0, 2]. Our goal is to find a data-driven controller $\mathbf{K}$ to stabilize this unstable plant using only experimental data. Applying the VLMI-method yields the following matrix $\mathbf{P}$ and controller $\mathbf{K}$:

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} 18.0627 & 0.7058 \\ 0.7058 & 4.7862 \end{bmatrix} \\ \mathbf{K} &= \begin{bmatrix} -24.5452 & 1.1205 \\ 7.3253 & -3.5479 \end{bmatrix} \end{aligned} \tag{21}$$

As expected, by construction, the controller stabilizes the polytope defined by the experimental data. In particular, the closed-loop poles corresponding to (20) are (-7.0267, -2.2454), hence the system is stable. Applying now Algorithm 1 to the same data yields :

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} 5.3551 & 1.7398 \\ 1.7398 & 1.3681 \end{bmatrix} \\ \mathbf{K} &= \begin{bmatrix} -37.3735 & -10.2398 \\ 1.5504 & -4.6361 \end{bmatrix} \end{aligned} \tag{22}$$

In this case, the closed-loop system corresponding to (20) has poles at (-17.924, -2.4359). Further, it is easy to verify that the pairs $(\mathbf{P}, \mathbf{K})$ satisfy the Lyapunov equation for all the vertices $(\mathbf{A}_i, \mathbf{B}_i)$ found using the VLMI method, certifying that indeed the controller $\mathbf{K}$ quadratically stabilizes it. The initial condition response of the system obtained when closing the loop around (20) with the controllers (21) and (22) are shown in Figs. 1-2. As illustrated there, both methods yield controllers that perform well. Hence, for this choice of parameters, there seems to be no advantage in using either algorithm. However, in many scenarios, guaranteeing feasibility of Problem 1 requires considering a moderately large number of samples, in order to reduce the uncertainty [2], at the cost of extra computational burden. As we show in the next example, the VLMI method is ill equipped to handle these scenarios, becoming quickly intractable as the number of samples grows, even for low dimensional examples.

[2] When the number of samples is relatively low, the consistency set is large and hence there may not exist a single controller that can simultaneously stabilize all the plants in this set.

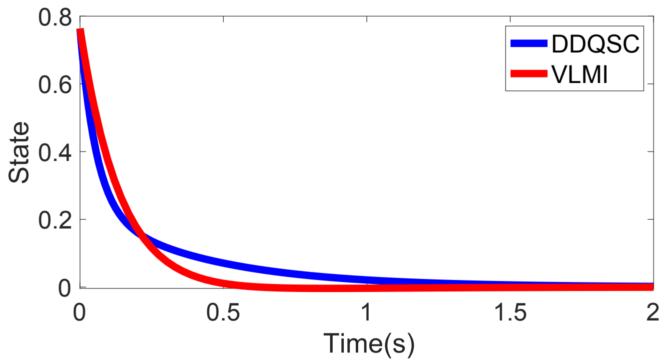

Fig. 1. Closed-Loop Response of Trajectory 1

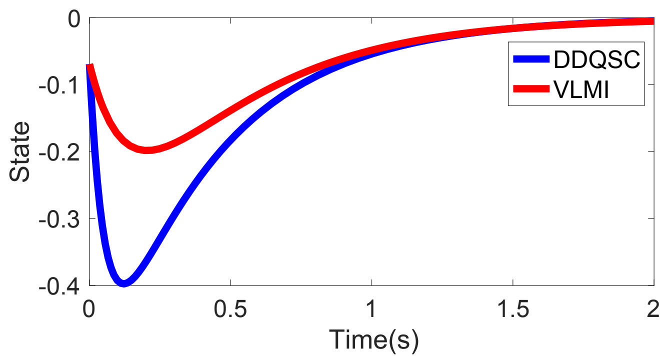

Fig. 2. Closed-Loop Response of Trajectory 2

5.2 Third-Order System

In this example, we illustrate the ability of the proposed approach to handle scenarios beyond the capabilities of the naive VMLI method. The system we used to generate the data is

$$\mathbf{A} = \begin{bmatrix} 0.8757 & 0.1566 & 0.4821 \\ 0.0854 & 0.5821 & 0.0267 \\ 0.2978 & 0.5931 & 0.3061 \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} 0.0525 & 0.1726 & -0.0130 \\ 0.3098 & -0.3037 & 0.1995 \\ -0.3168 & -0.0683 & 0.0252 \end{bmatrix} \tag{23}$$

with the same parameters used for the second-order case except we now use 50 samples, i.e. $s = 50$. As before, $\mathbf{A}$ is unstable, with eigenvalues (1.1372, 0.4798, 0.1469). Applying the VLMI method to this system is impractical. Finding the vertices of the polytope takes only 16 seconds but yields 12.7 million vertices. Solving an SDP with this number of constraints is beyond the ability of existing solvers. On the other hand, applying the algorithm 1 only takes 8.5164 seconds to find the solution:

$$\mathbf{P} = \begin{bmatrix} 14.7586 & 6.5061 & 1.8054 \\ 6.5061 & 3.5464 & 2.7224 \\ 1.8054 & 2.7224 & 8.4384 \end{bmatrix}$$
$$\mathbf{K} = \begin{bmatrix} -743.6 & -202.8 & 500.0 \\ -387.6 & -212.4 & -312.9 \\ 430.9 & -367.6 & -2274.0 \end{bmatrix} \tag{24}$$

The corresponding closed-loop poles are $(-9.369, -183.2 + 136.6i, -183.2 - 136.6i)$. A sample initial condition response is shown in Figs. 3-5.

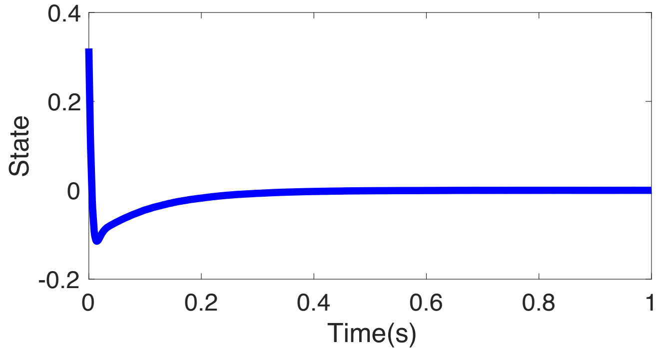

Fig. 3. Closed-Loop Response of Trajectory 1

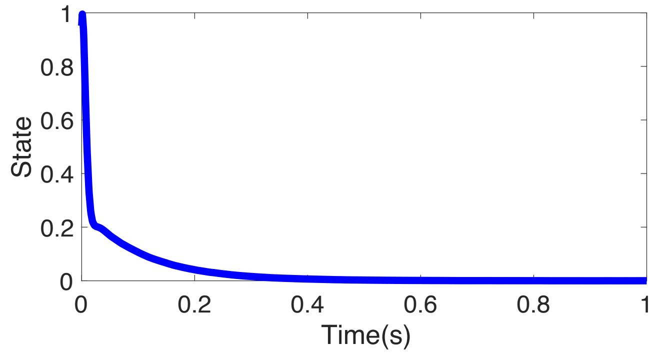

Fig. 4. Closed-Loop Response of Trajectory 2

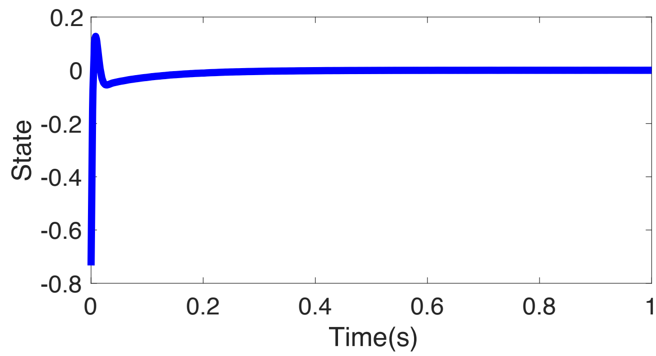

Fig. 5. Closed-Loop Response of Trajectory 3

5.3 Computational Complexity

In this part, we briefly discuss the time complexity of the VLMI method and the DDQSC method by counting the number of SDP constraints. It is well known that for m inequalities involving vectors in $\mathbb{R}^d$, the number of vertices is exponential $\Omega(m^{(\frac{d}{2})})$ in the size of the input. In our case, for an n^{th}-order system with n_s samples, we will have $\Omega(nn_s^{(n)})$ PSD constraints. That is quite a large number and hard to deal with. On the other hand the DDQSC algorithm only contains $(2nn_s + 2)$ PSD constraints and $(\frac{n^2(n+1)(n+m)}{2})$ linear constraints (see (19)). Hence its complexity grows linearly with the number of samples.

6. CONCLUSION

This paper proposes a simple data-driven approach to quadratically stabilize an unknown continuous LTI system, based solely on experimental measurements. While in principle this leads to a challenging non-convex robust optimization problem, our main result shows that a convex necessary and sufficient condition for quadratic stabilization of all plants compatible with the observed experimental data can be obtained via the theorem of alternatives, by rendering the dual of the original problem feasible. Since this convex problem is an infinite-dimensional SDP (due to the fact that the constraints must hold for all $\mathbf{x} \in \mathbb{R}^n$), in the second portion of the paper we provide a tractable finite-dimensional relaxation obtained by limiting the Lagrange multipliers, which in principle are arbitrary non-negative homogeneous functions of degree two, to second-order non-negative polynomials. Remarkably, the computational complexity of this relaxation grows linearly with the number of samples. For comparison, the complexity of a naive approach based on identifying the vertices of the polytope of matrices compatible with the observed data grows exponentially with the number of samples and thus becomes impractical beyond some toy problems.

Perhaps the most serious limitation of the proposed algorithm in its present form is that it only seeks to certify the quadratic stability of the closed-loop system, without taking performance into consideration. Research is currently underway to extend our approach to address performance. Possible venues to accomplishing this include adding a suitable objective to the feasibility problem (19) or replacing the Lyapunov inequalities with Riccati type inequalities, handled via the Bounded Real Lemma. However, these extensions are beyond the scope of the present paper.

REFERENCES

Avis, D. and Jordan, C. (2018). mplrs: A scalable parallel vertex/facet enumeration code. *Mathematical Programming Computation*, 10(2), 267–302.

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

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

Campi, M.C., Lecchini, A., and Savaresi, S.M. (2002). Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8), 1337–1346.

Chen, J. and Gu, G. (2000). *Control Oriented System Identification: an $\mathcal{H}_\infty$ approach.* Wiley & Sons, Inc.

Dai, T. and Sznaier, M. (2018). A moments based approach to designing mimo data driven controllers for switched systems. In *2018 IEEE Conference on Decision and Control (CDC)*, 5652–5657. IEEE.

De Persis, C. and Tesi, P. (2019). On persistency of excitation and formulas for data-driven control. *arXiv preprint arXiv:1903.06842*.

Formentin, S., Van Heusden, K., and Karimi, A. (2014). A comparison of model-based and data-driven controller tuning. *International Journal of Adaptive Control and Signal Processing*, 28(10), 882–897.

Hjalmarsson, H., Gevers, M., Gunnarsson, S., and Lequin, O. (1998). Iterative feedback tuning: theory and applications. *IEEE control systems magazine*, 18(4), 26–41.

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

Karimi, A., Mišković, L., and Bonvin, D. (2004). Iterative correlation-based controller tuning. *International journal of adaptive control and signal processing*, 18(8), 645–664.

Khargonekar, P.P., Petersen, I.R., and Zhou, K. (1990). Robust stabilization of uncertain linear systems: quadratic stabilizability and h/sup infinity/control theory. *IEEE Transactions on Automatic Control*, 35(3), 356–361.

Lasserre, J.B. (2009). *Moments, positive polynomials and their applications*, volume 1. World Scientific.

Lee, J.M. and Lee, J.H. (2005). Approximate dynamic programming-based approaches for input–output data-driven control of nonlinear processes. *Automatica*, 41(7), 1281–1288.

Löfberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD Conference*, volume 3. Taipei, Taiwan.

Lusch, B., Kutz, J.N., and Brunton, S.L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1), 1–10.

Sánchez Peña, R. and Sznaier, M. (1998). *Robust Systems Theory and Applications.* Wiley & Sons, Inc.

Shampine, L.F. and Reichelt, M.W. (1997). The matlab ode suite. *SIAM journal on scientific computing*, 18(1), 1–22.

Van Heusden, K., Karimi, A., and Bonvin, D. (2011). Data-driven model reference control with asymptotically guaranteed stability. *International Journal of Adaptive Control and Signal Processing*, 25(4), 331–351.

Zhang, H., Cui, L., Zhang, X., and Luo, Y. (2011). Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method. *IEEE Transactions on Neural Networks*, 22(12), 2226–2236.

Zhang, H., Jiang, H., Luo, Y., and Xiao, G. (2016). Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method. *IEEE Transactions on Industrial Electronics*, 64(5), 4091–4100.

Zhou, K. and Doyle, J.C. (1998). *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ.