Automatica 153 (2023) 111041

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Data-driven quadratic stabilization and LQR control of LTI systems ☆

Tianyu Dai, Mario Sznaier *

ECE Department, Northeastern University, Boston, MA, 02115, USA

ARTICLE INFO

Article history:
Received 20 May 2021
Received in revised form 5 December 2022
Accepted 20 March 2023
Available online 24 April 2023

Keywords:
Data-driven control
Robust control
Quadratic stability
Semi-definite programming

ABSTRACT

In this paper, we propose a framework to solve the data-driven quadratic stabilization (DDQS) and the data-driven linear quadratic regulator (DDLQR) problems for both continuous and discrete-time systems. Given noisy input/state measurements and a few priors, we aim to find a state feedback controller guaranteed to quadratically stabilize 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. Our main result shows that, by exploiting duality, the problem can be recast into a convex, albeit infinite-dimensional, functional Linear Program. To address the computational complexity entailed in solving this problem, we show that a sequence of increasingly tight finite dimensional semi-definite relaxations can be obtained using sum-of-squares and Putinar's Positivstellensatz arguments. Finally, we show that these arguments can also be used to find controllers that minimize a worst-case (over all plants in the consistency set) closed-loop $\mathcal{H}_2$ cost. The effectiveness of the proposed algorithm is illustrated through comparisons against existing data-driven methods that handle ℓ_∞ bounded noise.

1. Introduction

Robust control of uncertain systems has been well studied during the past decades, resulting in efficient synthesis methods that guarantee closed-loop stability of a set of plants, typically described by a nominal plant and bounded uncertainty (see for instance Sánchez Peña & Sznaier, 1998; Zhou & 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 & 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. Further, this intermediate identification step can be computationally expensive. These difficulties can be avoided by pursuing a data-driven control approach where a controller is designed directly from the data. Indeed, the recent work (De Persis & Tesi, 2019) showed that data-dependent matrices can be used as a proxy for the system model when designing controllers, while Berberich, Scherer, and Allgöwer (2020), Bisoffi, De Persis, and Tesi (2022) and Vanwaarde, Camlibel, and Mesbahi (2020) proposed procedures to design robust controllers guaranteed to stabilize the set of all plants compatible with noisy observations, under the assumption that the measurement noise admits a multiplier based description.

The techniques described above work well for ℓ_2 bounded noise, but are not well suited for scenarios with ℓ_∞ bounded noise. The latter setting is more desirable than the ℓ_2 norm in many scenarios since it allows for considering noise bounds that are independent of the measurement horizon (Berberich et al., 2020). Thus, data can be added as it becomes available during operation. In addition, these bounds arise naturally when considering discrete-time models that originate from the discretization of a continuous-time system.

In principle, ℓ_∞ noise can be handled using the Matrix Positivstellensatz introduced by Scherer and Hol (2006). However, this approach scales as $\mathcal{O}(n^{2r})$, where n and r denote the system and relaxation order, respectively. Thus, it quickly becomes intractable. Alternatively an approach based on the use of polyhedral control Lyapunov functions (PCLFs) was proposed in Dai and Sznaier (2018a, 2018b). While this approach can handle switching dynamics, this ability comes at the price of increased computational complexity, directly tied to the number of faces of the level sets of the PCLF. The present paper addresses this difficulty by considering quadratic Lyapunov functions, combined with the use of duality, leading to algorithms that scale as $\mathcal{O}(n^r)$, with n and r, and linearly with respect to the number of data

☆ This work was supported in part by U.S. National Science Foundation (NSF) grants CNS–2038493 and CMMI–2208182; AFOSR, United States grant FA9550-19-1-0005; and ONR, United States grant N00014-21-1-2431. The material in this paper was partially presented at the 21st IFAC World Congress (IFAC 2020), July 12–17, 2020, Berlin, Germany. This paper was recommended for publication in revised form by Associate Editor Maria Prandini under the direction of Editor Sophie Tarbouriech.

* Corresponding author.
E-mail addresses: dai.ti@northeastern.edu (T. Dai), msznaier@coe.neu.edu (M. Sznaier).

https://doi.org/10.1016/j.automatica.2023.111041
0005-1098/

points. Specifically, its contributions are:

(1) Non-conservative necessary and sufficient conditions, both for continuous and discrete-time plants, for the existence of a state feedback controller that quadratically stabilizes all plants in the consistency set, under ℓ_∞ bounded measurement noise.
(2) Continuous and discrete-time controller design methods that optimize a bound of the worst-case closed-loop $\mathcal{H}_2$ norm over all plants in the consistency set. These methods involve solving tractable semi-definite programs (SDPs).
(3) Extensive numerical comparisons against existing approaches for different noise scenarios.

A preliminary version of this paper that only considered quadratic stabilization of continuous-time systems was presented in Dai and Sznaier (2020). The present version includes complete proofs for both continuous and discrete-time plants, extends the design process to optimize $\mathcal{H}_2$ performance, and presents extensive comparisons vis-a-vis recently proposed data-driven control methods. The remainder of this paper is organized as follows: In Section 2, we state the problem under consideration and provide some background results necessary to solve it. Section 3 formulates the data-driven quadratic stabilization problem (DDQS) and data-driven linear quadratic regulator problem (DDLQR). Here we show that exploiting duality allows for recasting these problems as infinite-dimensional Linear Programs (LPs). As a follow-up to this result, we present a finite-dimensional semi-definite programming (SDP) based relaxation. Section 4 discusses in detail computational complexity and scaling issues. Section 5 compares the performance of the proposed approach against existing methods. Finally, Section 6 summarizes our results. To facilitate reading, all technical proofs are given in the Appendix.

2. Preliminaries

2.1. 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. $\mathbf{X} > (\geq)\ 0$ is a matrix with positive (non-negative entries). $\mathbf{Tr}(\mathbf{X})$ is the trace of the matrix. $\mathbf{I}$ and $\mathbf{0}$ denote the identity and zero matrices of suitable size. $\mathbf{1}$ represents a matrix of 1s. $\mathbb{R}[\mathbf{x}]_r$ denotes the set of real polynomials of degree up to r in the indeterminate $\mathbf{x}$ and $\mathbb{R}[\mathbf{x}]^{m\times n}$ denotes the set of $m \times n$ polynomial matrices. $\Sigma[\mathbf{x}]_r$ denotes the sub-cone of $\mathbb{R}[\mathbf{x}]$ formed by Sum-of-Squares (SoS) polynomials of order up to $2r$. A polynomial $p(\mathbf{x})$ is SoS if and only if there exists a vector of monomials of degree up to r, $\mathbf{v}_r(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]^s$, $s = \binom{n+r}{r}$, and a matrix $\mathbf{Q} \succeq 0$ (The Gram matrix) such that $p(\mathbf{x}) = \mathbf{v}_r(\mathbf{x})^T\mathbf{Q}\mathbf{v}_r(\mathbf{x})$.

2.2. Background results

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

Definition 1 (*Khargonekar, Petersen, & Zhou, 1990*)**.** Consider an uncertain system of the form $\delta\boldsymbol{\xi}(t) = \mathbf{A}\boldsymbol{\xi}(t)$ with $\mathbf{A} \in \mathcal{A}$, and 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. The system is said to be quadratically stable if there exists $\mathbf{Y} \succ 0$ such that, for any $\mathbf{A} \in \mathcal{A}$, $V = \boldsymbol{\xi}^T\mathbf{Y}^{-1}\boldsymbol{\xi}$ is a Lyapunov function, e.g. the following holds for continuous-time systems

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

and

$$\mathbf{A}\mathbf{Y}\mathbf{A}^T - \mathbf{Y} \prec 0, \quad \forall \mathbf{A} \in \mathcal{A} \tag{2}$$

for the discrete-time case. Similarly, a system of the form $\delta\boldsymbol{\xi}(t) = \mathbf{A}\boldsymbol{\xi}(t) + \mathbf{B}\boldsymbol{\xi}(t)$, with uncertain $\mathbf{A} \in \mathcal{A}$ and $\mathbf{B} \in \mathcal{B}$ is said to be quadratically stabilizable if a state feedback controller $\mathbf{u} = \mathbf{K}\boldsymbol{\xi}(t)$ can be found such that the resulting closed-loop system is quadratically stable for any pairs $(\mathbf{A} \in \mathcal{A}, \mathbf{B} \in \mathcal{B})$.

Strong Duality and the Weak Slater's Conditions
The following result will play a key role in recasting the data-driven quadratic stabilization problem into a tractable form. Given $n+1$ affine functions $f_i(\mathbf{x})$, consider the following (primal) optimization problem:

$$\begin{aligned} p^* \quad = \quad & \min_{\mathbf{x}} f_o(\mathbf{x}) \text{ subject to:} \\ & f_i(\mathbf{x}) \leq 0,\ \ i = 1,\dots,n \end{aligned} \tag{primal}$$

The dual function associated with the primal problem is:

$$g(\boldsymbol{\mu}) = \inf_{\mathbf{x}} f_o(\mathbf{x}) + \sum_{i=1}^{n} \mu_i f_i(\mathbf{x}) \tag{3}$$

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

$$d^* = \max_{\boldsymbol{\mu}\geq 0} g(\boldsymbol{\mu}) \tag{dual}$$

Lemma 1. *If the primal problem is feasible then $p^* = d^*$.*

The proof can be found in Boyd and Vandenberghe (2004), Section 5.2.3.

Semialgebraic Optimization
A basic semialgebraic set $\mathbb{K}$ is defined by a finite collection of polynomial inequalities:

$$\mathbb{K} = \{\mathbf{x}\ |g_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]_{r_g} \geq 0, i = 1\dots n_g\} \tag{4}$$

The set $\mathbb{K}$ is said to be Archimedean if there exists a finite R and SoS polynomials $\{\sigma^i(\mathbf{x})\}_{i=0}^{n_g}$ such that

$$R^2 - \|\mathbf{x}\|_2^2 = \sigma^o(\mathbf{x}) + \sum_{i=1}^{n_g} \sigma^i(\mathbf{x}) g_i(\mathbf{x}). \tag{5}$$

A necessary and sufficient condition for a polynomial $p(x)$ to be positive over an Archimedean semialgebraic set $\mathbb{K}$ is given by Putinar's Positivstellensatz (P-satz) (Putinar, 1993), i.e. there exist $\sigma^o, \sigma^i(\mathbf{x}) \in \boldsymbol{\Sigma}[\mathbf{x}]$ such that

$$p(\mathbf{x}) = \sigma^o(\mathbf{x}) + \sum_{i=1}^{n_g} \sigma^i(\mathbf{x}) g_i(\mathbf{x}) \tag{6}$$

The degree-$2r$ tightening of this condition uses a bounded-degree Putinar's P-satz obtained when σ^o, σ^i are restricted to: $\sigma^o(\mathbf{x}) \in \boldsymbol{\Sigma}[\mathbf{x}]_r$, $\sigma^i(\mathbf{x}) \in \boldsymbol{\Sigma}[\mathbf{x}]_{r-\lfloor \frac{d_g}{2} \rfloor}$.

$\mathcal{H}_2$ Control
Given an LTI system:

$$\delta\boldsymbol{\xi} = \mathbf{A}\boldsymbol{\xi} + \mathbf{B}\mathbf{u} + \boldsymbol{w}$$

$$\mathbf{z} = \begin{bmatrix} \mathbf{R}^{1/2} & 0 \\ 0 & \mathbf{Q}^{1/2} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\xi} \end{bmatrix}$$

the state feedback $\mathcal{H}_2$ control problem seeks to find a control law $\mathbf{u} = \mathbf{K}\boldsymbol{\xi}$ that minimizes the $\mathcal{H}_2$ norm from the input $\boldsymbol{w}$ to the output $\mathbf{z}$. It is well known (see for instance Feron, Balakrishnan, Boyd, & Ghaoui, 1992) that, this problem is equivalent to the following SDP:

$$\begin{aligned} & \text{minimize } \gamma \\ & C(\gamma, \mathbf{Y}, \mathbf{M}, \mathbf{W}) \succ 0 \end{aligned} \tag{7}$$

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

$$\begin{aligned}
&C_1(\gamma, \mathbf{Y}, \mathbf{M}, \mathbf{W}) \doteq -\mathbf{Tr}(\mathbf{QY}) - \mathbf{Tr}(\mathbf{W}) + \gamma\\
&C_2(\mathbf{Y}, \mathbf{M}) \doteq \Omega(\mathbf{Y}, \mathbf{M}) + \Omega^T(\mathbf{Y}, \mathbf{M}) + \mathbf{L}(\mathbf{Y}) + \mathbf{L}_o\\
&\Omega(\mathbf{Y}, \mathbf{M}) = \mathbf{H}_L(\mathbf{AY} + \mathbf{BM})\mathbf{H}_R\\
&C_3(\mathbf{Y}, \mathbf{M}, \mathbf{W}) \doteq \begin{bmatrix} \mathbf{W} & \mathbf{R}^{1/2}\mathbf{M} \\ \mathbf{M}^T\mathbf{R}^{1/2} & \mathbf{Y} \end{bmatrix}\\
&C(\gamma, \mathbf{Y}, \mathbf{M}, \mathbf{W}) \doteq \mathrm{diag}(C_1, C_2, C_3)
\end{aligned} \tag{8}$$

and:

$$-\mathbf{H}_L = \mathbf{H}_R = \mathbf{I};\ \mathbf{L} = \mathbf{0}, \mathbf{L}_o = -\mathbf{I} \ \text{(continuous-time)} \tag{9a}$$

or

$$\left.\begin{aligned} &\mathbf{H}_L = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}^T, \mathbf{H}_R = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\\ &\mathbf{L}(\mathbf{Y}) = \begin{bmatrix}\mathbf{Y} & \mathbf{0}\\ \mathbf{0} & \mathbf{Y}\end{bmatrix}, \ \mathbf{L}_o = \begin{bmatrix}-\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\end{aligned}\right\} \text{(discrete-time)} \tag{9b}$$

Later in Section 3.3 we will use these results to search for data-driven robust $\mathcal{H}_2$ controllers.

2.3. Problem statement

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

Problem 1. Consider an LTI system:

$$\delta\boldsymbol{\xi}(t) = \mathbf{A}\boldsymbol{\xi}(t) + \mathbf{B}\mathbf{u}(t) + \boldsymbol{\eta}(t) \tag{10}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$ 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 $k = 1, \ldots, n_s$ denotes the sample index, the goal is to find a state feedback controller $\mathbf{u} = \mathbf{K}\boldsymbol{\xi}(t)$ guaranteed to quadratically stabilize all pairs $(\mathbf{A}, \mathbf{B})$ that could have generated the observed data.

Problem 2 (*Robust $\mathcal{H}_2$*)**.** Find a control law $\mathbf{u} = \mathbf{K}\boldsymbol{\xi}(t)$ that minimizes the worst case, over all systems in the consistency set, of the $\mathcal{H}_2$ cost γ defined in (7)–(8).

The main result of this paper shows that the problems above can be reformulated as SDPs.

2.4. Alternative characterization of the consistency set

Before proceeding to the main part of the paper, we present an alternative characterization of the consistency set that will be used to reduce Problems 1 and 2 to a tractable SDP. Note that each measurement $(\delta\boldsymbol{\xi}(t_k), \boldsymbol{\xi}(t_k), \mathbf{u}(t_k))$ yields $2n$ polytopic constraints on the elements of $(\mathbf{A}, \mathbf{B})$:

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ -\mathbf{A} & -\mathbf{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{\xi}(t_k)\\ \mathbf{u}(t_k)\end{bmatrix} \leq \begin{bmatrix}\epsilon\mathbf{1} + \delta\boldsymbol{\xi}(t_k)\\ \epsilon\mathbf{1} - \delta\boldsymbol{\xi}(t_k)\end{bmatrix} \doteq \mathbf{d}_k \tag{11}$$

For reasons that will become clear below, we rewrite the constraints (11) in the following form:

$$\mathbf{Tr}(\mathbf{A}\mathbf{Z}^{\boldsymbol{\xi}}_{i,k} + \mathbf{B}\mathbf{Z}^{\mathbf{u}}_{i,k}) \leq d_{i,k} \tag{12}$$

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

$$\begin{aligned}\mathbf{Z}^{\boldsymbol{\xi}}_{i,k} &= \left[\ldots, \pm\boldsymbol{\xi}(t_k), \ldots\right]_{n\times n}\\ \mathbf{Z}^{\mathbf{u}}_{i,k} &= \left[\ldots, \pm\mathbf{u}(t_k), \ldots\right]_{m\times n}\end{aligned} \tag{13}$$

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

$$\begin{aligned}&\mathbf{Z}^{\xi}_{1,k} = \begin{bmatrix}\xi_1(t_k) & 0\\ \xi_2(t_k) & 0\end{bmatrix}, \quad \mathbf{Z}^{\xi}_{2,k} = \begin{bmatrix}0 & \xi_1(t_k)\\ 0 & \xi_2(t_k)\end{bmatrix},\\ &\mathbf{Z}^{\xi}_{3,k} = \begin{bmatrix}-\xi_1(t_k) & 0\\ -\xi_2(t_k) & 0\end{bmatrix}, \mathbf{Z}^{\xi}_{4,k} = \begin{bmatrix}0 & -\xi_1(t_k)\\ 0 & -\xi_2(t_k)\end{bmatrix}\end{aligned} \tag{14}$$

with similar expressions for $\mathbf{Z}^{\mathbf{u}}_{i,k}$.

3. DDQS and DDLQR

3.1. A necessary and sufficient condition for quadratic stabilization

In this section we show that Problem 1 can be recast as a convex (albeit infinite-dimensional) optimization by exploiting duality. It is well known that the control law $\mathbf{K} = \mathbf{M}\mathbf{Y}^{-1}$ quadratically stabilizes the system (10) if and only if the following Linear Matrix Inequality (LMI) in $\mathbf{Y} \succ 0$ and $\mathbf{M}$ is feasible:

$$\Omega(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{Y}) + \Omega^T(\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{Y}) + \mathbf{L}(\mathbf{Y}) \succ 0 \tag{15}$$

where Ω is defined in (8) and $\mathbf{H}_L, \mathbf{H}_R, \mathbf{L}$ are defined in (9). We want this to hold for all $(\mathbf{A}, \mathbf{B})$ constrained by (12) leading to the following version of Problem 1.

Problem 3. Find matrices $\mathbf{Y} \succ 0$ and $\mathbf{M}$ such that (15) holds for all $(\mathbf{A}, \mathbf{B})$ that satisfy (12).

The next result shows that this problem is equivalent to an optimization over positive polynomials.

Theorem 1. *Problem 3 is feasible if and only if there exist matrices $\mathbf{Y} \in \mathbb{R}^{n\times n} \succ 0$, $\mathbf{M} \in \mathbb{R}^{m\times n}$ 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:*

$$\boldsymbol{\Xi}\boldsymbol{\Upsilon}(\mathbf{x}) = -2\begin{bmatrix}\mathbf{Y}\\ \mathbf{M}\end{bmatrix}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L \tag{16a}$$

$$\mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} - \mathbf{Tr}(\mathbf{D}\boldsymbol{\Upsilon}(\mathbf{x})) > 0 \text{ for all } \|\mathbf{x}\|_2 \leq 1 \tag{16b}$$

where

$$\begin{aligned}\boldsymbol{\Xi} &\doteq \begin{bmatrix}\boldsymbol{\xi}(t_1)\ldots\boldsymbol{\xi}(t_{n_s}) & -\boldsymbol{\xi}(t_1)\ldots-\boldsymbol{\xi}(t_{n_s})\\ \mathbf{u}(t_1)\ldots\mathbf{u}(t_{n_s}) & -\mathbf{u}(t_1)\ldots-\mathbf{u}(t_{n_s})\end{bmatrix},\\ \mathbf{D} &\doteq \begin{bmatrix}\mathbf{d}^+(t_1)\ldots\mathbf{d}^+(t_{n_s}) & \mathbf{d}^-(t_1)\ldots\mathbf{d}^-(t_{n_s})\end{bmatrix}\end{aligned}$$

$$\mathbf{d}^+(t_k) \doteq \epsilon\mathbf{1} + \delta\boldsymbol{\xi}(t_k), \mathbf{d}^-(t_k) \doteq \epsilon\mathbf{1} - \delta\boldsymbol{\xi}(t_k)$$

Proof. See Appendix A. □

3.2. A semi-definite reformulation

From Theorem 1 it follows that Problem 3 can be recast as an infinite-dimensional functional Linear Program. Tractable, finite-dimensional relaxations can be obtained by enforcing that each entry of $\boldsymbol{\Upsilon}(\mathbf{x})$ satisfies a Putinar's P-satz associated with positivity over the set $(1 - \mathbf{x}^T\mathbf{x}) \geq 0$:

$$\boldsymbol{\Upsilon}_{i,k}(\mathbf{x}) = \sigma^o_{i,k}(\mathbf{x}) + \sigma^1_{i,k}(\mathbf{x})(1 - \mathbf{x}^T\mathbf{x}) \tag{17}$$

for some SoS polynomials $\sigma^o_{i,k}(\mathbf{x}) \in \boldsymbol{\Sigma}_r, \sigma^1_{i,k}(\mathbf{x}) \in \boldsymbol{\Sigma}_{r-1}$. Thus, one can formulate a sequence of relaxations of increasing order, starting with $r = 1$, with $\sigma^o_{i,k}(\mathbf{x}) = \mathbf{v}_r^T(\mathbf{x})\mathbf{E}^o_{i,k}\mathbf{v}_r(\mathbf{x})$ and $\sigma^1_{i,k}(\mathbf{x}) = \mathbf{v}_{r-1}^T(\mathbf{x})\mathbf{E}^1_{i,k}\mathbf{v}_{r-1}(\mathbf{x})$.

Similarly, from (16b), we have

$$\mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} - \mathbf{Tr}(\mathbf{D}\boldsymbol{\Upsilon}(\mathbf{x})) = \sigma^o(\mathbf{x}) + \sigma^1(\mathbf{x})(1 - \mathbf{x}^T\mathbf{x}) \tag{18}$$

where $\sigma^o(\mathbf{x}) = \mathbf{v}_r^T(\mathbf{x})\mathbf{F}^o\mathbf{v}_r(\mathbf{x})$ and $\sigma^1(\mathbf{x}) = \mathbf{v}_{r-1}^T(\mathbf{x})\mathbf{F}^1\mathbf{v}_{r-1}(\mathbf{x})$. By equating the coefficients of the polynomial matrices on both

[1] This noise arises for instance from numerical estimation of $\delta\xi(t)$ from measurements of $\xi(t)$.

sides, (16a) gives a linear constraint $\mathbf{k}_{eq}(\mathbf{E}^o_{i,k}, \mathbf{E}^1_{i,k}) = \mathbf{k}_{eq}(\mathbf{Y}, \mathbf{M})$, and (18) introduces another linear constraint $\mathbf{k}_{ineq}(\mathbf{E}^o_{i,k}, \mathbf{E}^1_{i,k}, \mathbf{Y}) = \mathbf{k}_{ineq}(\mathbf{F}^o, \mathbf{F}^1)$. Algorithm 1 summarizes the SDP-based procedure to solve (16):

Algorithm 1 DDQS

1: Given n_s measurements $\boldsymbol{\xi}(t_k)$, $\mathbf{u}(t_k)$, $\delta\boldsymbol{\xi}(t_k)$ and a noise bound ϵ, build data matrices Ξ, $\mathbf{D}$ and decide performance matrices $\mathbf{Q}$, $\mathbf{R}$.
2: Solve:
3: minimize 0
4: subject to

$$\mathbf{k}_{eq}(\mathbf{E}^o_{i,k}, \mathbf{E}^1_{i,k}) = \mathbf{k}_{eq}(\mathbf{Y}, \mathbf{M}) \tag{19a}$$

$$\mathbf{k}_{ineq}(\mathbf{E}^o_{i,k}, \mathbf{E}^1_{i,k}, \mathbf{Y}) = \mathbf{k}_{ineq}(\mathbf{F}^o, \mathbf{F}^1) \tag{19b}$$

$$\mathbf{E}^o_{i,k} \succeq 0, \mathbf{E}^1_{i,k} \succeq 0 \tag{19c}$$

$$\mathbf{F}^o \succeq 0, \mathbf{F}^1 \succeq 0 \tag{19d}$$

$$\mathbf{Y} \succ 0 \tag{19e}$$

Remark 1. 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 for a given relaxation order, does not rule out the existence of solutions to Problem 1. While in principle Putinar's P-satz guarantees that if Problem 3 is feasible then there exist some finite r such that the SDP associated with (16) is feasible, since the number of variables scales combinatorially with r, in practice one is limited to relatively low order relaxations. Nevertheless consistent numerical experience shows that the relaxation $r = 1$ works well in practice.

3.3. Achieving robust performance

Next, we show that the framework developed above can be easily extended to minimize an upper bound of the worst case, over the consistency set, of the closed-loop $\mathcal{H}_2$ norm. Specifically, we are interested in solving:

Problem 4.

$$\min_{\mathbf{Y}\succ 0,\mathbf{M},\mathbf{W}} \left\{ \max_{\mathbf{A},\mathbf{B}} \gamma \right\} \text{ subject to (8) and}$$

$$d_{i,k} - \mathbf{Tr}(\mathbf{A}\mathbf{Z}^x_{i,k} + \mathbf{B}\mathbf{Z}^u_{i,k}) \geq 0$$

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

Theorem 2. *Problem 4 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:*

$$\Xi\boldsymbol{\Upsilon}(\mathbf{x}) = -2\begin{bmatrix}\mathbf{Y}\\ \mathbf{M}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{H}_R & \mathbf{0}\end{bmatrix}\mathbf{x}\mathbf{x}^T\begin{bmatrix}\mathbf{0}\\ \mathbf{H}_L\\ \mathbf{0}\end{bmatrix} \tag{20a}$$

$$(\gamma - \mathbf{Tr}(\mathbf{Q}\mathbf{Y}) - \mathbf{Tr}(\mathbf{W}))\begin{bmatrix}1 & \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{x}\mathbf{x}^T\begin{bmatrix}1\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}$$

$$+\mathbf{Tr}\left(\begin{bmatrix}\mathbf{W} & \mathbf{R}^{1/2}\mathbf{M}\\ \mathbf{M}^T\mathbf{R}^{1/2} & \mathbf{Y}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}_{n+m}\end{bmatrix}\mathbf{x}\mathbf{x}^T\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{I}_{n+m}\end{bmatrix}\right)$$

$$+\mathbf{Tr}\left((\mathbf{L}(\mathbf{Y}) + \mathbf{L}_o)\begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{x}\mathbf{x}^T\begin{bmatrix}\mathbf{0}\\ \mathbf{I}\\ \mathbf{0}\end{bmatrix}\right) - \mathbf{Tr}(\mathbf{D}\boldsymbol{\Upsilon}(\mathbf{x})) > 0 \tag{20b}$$

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

Proof. Given in Appendix B □

As before, enforcing Putinar's P-satz on $\boldsymbol{\Upsilon}(\mathbf{x})$ and (20b) and collecting the coefficients of (20a) and (20b) in $\mathbf{k}_{eq}$ and $\mathbf{k}_{ineq}$ leads to the following algorithm:

Algorithm 2 DDLQR

1: Given n_s measurements $\boldsymbol{\xi}(t_k)$, $\mathbf{u}(t_k)$, $\delta\boldsymbol{\xi}(t_k)$ and a noise bound ϵ, build data matrices Ξ, $\mathbf{D}$ and decide performance matrices $\mathbf{Q}$, $\mathbf{R}$.
2: Solve:
3: minimize γ
4: subject to

$$\mathbf{k}_{eq}(\mathbf{E}^o_{i,k}, \mathbf{E}^1_{i,k}) = \mathbf{k}_{eq}(\mathbf{Y}, \mathbf{M}) \tag{21a}$$

$$\mathbf{k}_{ineq}(\mathbf{E}^o_{i,k}, \mathbf{E}^1_{i,k}, \mathbf{Y}, \mathbf{M}, \mathbf{W}, \gamma) = \mathbf{k}_{ineq}(\mathbf{F}^o, \mathbf{F}^1) \tag{21b}$$

$$\mathbf{E}^o_{i,k} \succeq 0, \mathbf{E}^1_{i,k} \succeq 0 \tag{21c}$$

$$\mathbf{F}^o \succeq 0, \mathbf{F}^1 \succeq 0 \tag{21d}$$

$$\mathbf{Y} \succ 0 \tag{21e}$$

$$\mathbf{W} \succ 0 \tag{21f}$$

4. Complexity analysis

In this section we analyze the computational complexity of the proposed approach and compare it against existing techniques that can also handle ℓ_∞ bounded noise: Berberich et al. (2020), Chesi (2010), De Oliveira, Bernussou, and Geromel (1999) and Scherer and Hol (2006). The discussion is limited to the size of the Gram matrices and the corresponding number of constraints since they are the main source of computational complexity.

4.1. Discrete DDLQR

Given $\boldsymbol{\Upsilon}(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]^{2n_s\times n}$, we have $2nn_s$ positive scalar polynomials in an indeterminate $\mathbf{x} \in \mathbb{R}^{3n+m+1}$. Each polynomial contains a Gram matrix $\mathbf{E}^o_{i,k}$ of size $n_o = \binom{n_e+r}{r}$ and a Gram matrix $\mathbf{E}^1_{i,k}$ of size $n_1 = \binom{n_e+r-1}{r-1}$ where $n_e = 3n + m + 1$. In total, the Gram matrices introduce $nn_s(n_o^2 + n_1^2 + n_0 + n_1)$ variables. These also introduce $2nn_s$ PSD constraints of size n_o and $2nn_s$ PSD constraints of size n_1. Asymptotic complexity, given by the size of the largest Gram matrix is roughly $\mathcal{O}(n^r)$.

Remark 2. Note that, due to the projections, $\mathbf{x}$ appears blockwise quadratically in the right hand side of (20a) and in all terms in (20b) except the last. For the case $r = 1$, this motivates a new relaxation where $\boldsymbol{\Upsilon}(\mathbf{x})$ is also taken to be a function of these blocks, or, equivalently, its associated Gram matrices satisfy $\mathbf{E}^1_{i,k} = 0$ and $\mathbf{E}^o_{i,k}$ is a block diagonal matrix:

$$\mathbf{E}^o_{i,k} = \begin{bmatrix}E_1 & & 0\\ & E_{2n} & \\ 0 & & E_{n+m}\end{bmatrix} \tag{22}$$

This relaxation involves only $nn_s(5n^2+m^2+2nm+3n+m)$ variables, leading to a substantial reduction in computational complexity.

4.2. Comparison against Scherer's matrix P-satz

The approach proposed in Scherer and Hol (2006) is perhaps the closest existing technique to the approach proposed in this paper, since both use SoS Techniques to impose positivity of a polynomial matrix over a semi-algebraic set. However, as we show here, the fact that we exploit the specific structure of the polynomial matrices in a duality-based context leads to a substantial complexity reduction over a naive approach that treats the DDLQR problem as a special case of Scherer and Hol (2006). For completeness we begin by restating *Theorem 1* in Scherer and Hol (2006):

Theorem 3. *If there exists some $r>0$ and SoS polynomials $\psi_i(\mathbf{x})$ such that*

$$r^2-\|\mathbf{x}\|^2=\psi_o(\mathbf{x})+\sum_{i=1}^{n_c}\psi_i(\mathbf{x})g_i(\mathbf{x}) \tag{23}$$

then

$\mathbf{H}(\mathbf{x},\mathbf{y})\succ 0$ *for all* $\mathbf{x}\in\mathbb{K}=\{\mathbf{x}\mid g_i(\mathbf{x})\geq 0\}$ *if and only if*

$$\mathbf{H}(\mathbf{x},\mathbf{y})=\mathbf{S}_o(\mathbf{x})+\sum_{i=1}^{n_g}\mathbf{S}_i(\mathbf{x})g_i(\mathbf{x})+\epsilon\mathbf{I}_p \tag{24}$$

for some $p\times p$ SoS matrices $\mathbf{S}_i(\mathbf{x})$ and a scalar $\epsilon>0$

Recall that a $p\times p$ matrix $\mathbf{S}(\mathbf{x})$ with polynomial entries is said to be an SoS matrix if there exist a polynomial matrix $\mathbf{L}(\mathbf{x})$ such that $\mathbf{S}(\mathbf{x})=\mathbf{L}^T(\mathbf{x})\mathbf{L}(\mathbf{x})$. It can be shown that $\mathbf{S}(\mathbf{x})$ containing monomials of order up to $2r$ is SoS if and only if it can be written as

$$\mathbf{S}(\mathbf{x})=(\mathbf{v}_r(\mathbf{x})\otimes\mathbf{I}_p)^T\mathbf{Z}(\mathbf{v}_r(\mathbf{x})\otimes\mathbf{I}_p),\ \ \mathbf{Z}\succeq 0 \tag{25}$$

Consider the discrete-time LQR problem for a system with n states and m control inputs. From (7), we have

$$\begin{gathered}C(\mathbf{A},\mathbf{B},\gamma,\mathbf{Y},\mathbf{M},\mathbf{W})\succ 0 \text{ for all}\\ \|\mathbf{x}(k+1)-\mathbf{A}\mathbf{x}(k)-\mathbf{B}\mathbf{u}(k)\|_\infty\leq 1,\, k=1\dots n_s\end{gathered} \tag{26}$$

so for each time step we get $2n$ constraints on the entries of $\mathbf{a}\doteq\mathbf{vec}(\mathbf{A})$, $\mathbf{b}=\mathbf{vec}(\mathbf{B})$, for a total of $2nn_s$ constraints. Using (24) to recast this as an SoS program in $\mathbf{a}$, $\mathbf{b}$ requires $1+2nn_s$ SoS $n_e\times n_e$ polynomial matrices, in an indeterminate $\boldsymbol{\zeta}=\begin{bmatrix}\mathbf{a}^T & \mathbf{b}^T\end{bmatrix}^T$ of dimension $n_\zeta\doteq n^2+nm$. Hence, from (25) it follows that the first multiplier in (24) leads to a Gram matrix of size $n_o=\binom{n_\zeta+r}{n_\zeta}n_e$ and the remaining $2nn_s$ to Gram matrices of size $n_1=\binom{n_\zeta+r-1}{n_\zeta}n_e$. The total number of variables for Gram matrices is given by $\frac{n_o^2+n_o}{2}+nn_s(n_1^2+n_1)$ for n_o, n_1 defined above. Thus, in this case Asymptotic complexity, given by the size of the largest Gram matrix, is roughly $\mathcal{O}(n^{2r})$, compared to $\mathcal{O}(n^r)$ for our approach. Tables 1–2 illustrate the scaling of the problem for different values of n and r. As shown there, Scherer P-satz quickly leads to problems beyond the capability of existing solvers, even when considering low-order systems, short horizons and low-order relaxations ($n=5$, $m=2$, $n_s=10$, $r=1$).

Remark 3. The computational complexity reduction achieved by our approach stems from the following facts:

(a) In Scherer P-satz, the dimension of ambient space is the number of entries in $\mathbf{A}$, $\mathbf{B}$ which is $\mathcal{O}(n^2)$. In contrast, our method considers the ambient space in $\mathbf{x}$ which is $\mathcal{O}(n)$. This gap is substantial when n is large.

Table 1
Scherer P-satz versus DDLQR, $n=2$, $m=2$, $n_s=10$, $r=1$.

	Scherer P-satz	DDLQR
Dimension of ambient space	8	9
Number of Gram matrices	41	80
Dimension of Gram matrices	81, 9	10, 1
Number of variables	5121	2240

Table 2
Scherer P-satz versus DDLQR, $n=5$, $m=2$, $n_s=10$, $r=1$.

	Scherer P-satz	DDLQR
Dimension of ambient space	35	18
Number of Gram matrices	101	200
Dimension of Gram matrices	648, 18	19, 1
Number of variables	227 376	19 100

(b) The size of the Gram matrix in Scherer P-satz is roughly $\mathcal{O}(n^{2r})$ compared to $\mathcal{O}(n^r)$ in our case.

Finally, consistent numerical experience indicates that, in most cases, selecting $r=1$ in our approach suffices to solve the problem. Hence, we can apply the relaxation described in Remark 2. Thus, the number of variables in Tables 1(2) decreases from 2240(19100) to 800(8300). This structural benefit is not seen in Scherer P-satz.

4.3. Multiplier-based techniques

Theorem 1 in Berberich et al. (2020) addresses the robust data-driven control problem using a multiplier based description of the noise (equation (23) in Berberich et al., 2020). This approach requires a Gram matrix of size $n+n_s$, hence only a small number of variables. However, for ℓ_∞ bounded noise, the number of semidefinite constraints is 2^{nT} which quickly becomes intractable. For instance, for $n=2$, $m=2$, $n_s=10$, the number of constraints is 10^6.

4.4. Vertex LMI-based techniques

Theorem 13 in Chesi (2010) states that a robust control law can be found by enforcing an LMI at the vertices of the polytope of the consistency set. Each of these LMIs leads to a Gram matrix of size $(2r+1)^{n_e}$. Similarly, De Oliveira et al. (1999) also require enforcing the LMIs (8) at each vertex. In this case the size of the Gram matrix is only n_e. However, the worst-case number of vertices of the consistency set grows combinatorially with n_ζ and n_s (Avis & Jordan, 2018). For instance, for $n=5$, $m=2$, $n_s=10$, the upper bound of the number of vertices is 1.8×10^{15} given by Eq. (2) in Avis and Jordan (2018). We randomly generated 10 trajectories and compute the number of vertices. On average, we get 6.8×10^{12} vertices. This number is clearly beyond the ability of existing SDP solvers.

5. Simulation results

In this section we investigate the performance of the proposed algorithms, in the discrete-time case as a function of the noise level and the number of samples used to determine the consistency set. To benchmark the proposed approach against existing techniques that handle ℓ_∞ bounded noise, we compare it with Berberich et al. (2020) and De Oliveira et al. (1999), where robust DDC is directly addressed using LMI-based techniques, and the SoS-based technique in Scherer and Hol (2006). These comparisons show that while all approaches are able to solve the

Table 3
Closed-loop $\mathcal{H}_2$ norm for different ϵ and $n_s = 20$.

ϵ	0.2	0.4	0.6	0.8
C_{clp}	6.6815	6.8581	7.2067	7.6362
γ	7.8605	9.6386	11.6791	15.2135

Table 4
Closed-loop $\mathcal{H}_2$ norm for different n_s and $\epsilon = 0.4$.

n_s	10	20	30	40
C_{clp}	7.4863	6.8581	6.7154	6.6672
γ	16.5077	9.6386	8.2640	7.8898

robust DDLQR problem for low-order systems and short horizons, only our approach has the capability to handle moderately sized problems. Finally, we compare our approach with Model-Based Control (MBC) and provide a way to incorporate the partial information. All simulations in the paper were run on a MacBook with a processor 2.2 GHz 6-core Intel Core i7. Codes are implemented using (MATLAB, 2020) and the optimization problems are solved using a combination of YALMIP (Löfberg, 2004) and the (MOSEK, 2019) SDP solver.

5.1. Monte Carlo experiments

In this section we use Monte Carlo experiments to analyze the effects of the noise level ϵ and number of samples n_s. We consider the discrete-time case using data generated by the model

$$\mathbf{A} = \begin{bmatrix} 0.4285 & -0.4298 \\ 0.4018 & 1.3036 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -0.7826 & 0.7731 \\ -0.5110 & 0.0339 \end{bmatrix} \tag{27}$$

This system, generated with the randn command in MATLAB, has eigenvalues (0.7291, 1.003). In all instances for simplicity we consider the lowest order relaxation, with $r = 1$, that is, the functions $\boldsymbol{\Upsilon}_{i,k}(.)$ are second order positive polynomials. As shown below, even this simple relaxation performs well, specially in the case of moderate noise levels.

5.1.1. Discrete-time LQR

We fixed $n_s = 20$ and used Algorithm 2 to design a robust LQR controller. To analyze the performance of the system as a function of noise, ϵ was selected from [0.2, 0.4, 0.6, 0.8]. If the resulting controller stabilized the system, we computed C_{clp} and γ. The median of C_{clp} and γ (over 50 runs) are shown in Table 3. For benchmark purposes, we also solved (7) and obtained the ground truth $\gamma_0 = 6.5863$. One should note that $\gamma_0 \le C_{clp} \le \gamma$ and equality holds only if data is clean.

Next, we fix $\epsilon = 0.4$ and compute the $\mathcal{H}_2$ norm for different n_s. We run the experiment 50 times and compute the median of C_{clp} and γ (Table 4):

As expected, C_{clp} and the worst-case performance bound γ approach the optimal, noiseless performance as ϵ decreases and n_s increases.

5.2. Comparison with existing methods

Following the discussion in Section 4, we provide simulation results to illustrate the advantages of our approach over existing ones. Table 5 shows a simulation result for system (27) with $\epsilon = 0.05$ and $n_s = 6$. Time is averaged over 50 runs. In this case, all methods were able to find a *LQR* controller with good performance. Further, it should come with no surprise that our approach and Scherer and Hol (2006) have the same C_{clp} and γ

Table 5
$\mathcal{H}_2$ norm of the closed-loop system ($\epsilon = 0.05$, $n_s = 6$).

	C_{clp}	γ	t (s)
Algorithm 2	6.6915	8.1877	1.0018
Algorithm 2 with (22)	6.6915	8.1877	0.8574
Algorithm 2 with $r = 2$	6.6915	8.1875	231.1249
Berberich et al. (2020)	6.6696	8.0247	4.8118
Scherer and Hol (2006)	6.6915	8.1877	6.2675
De Oliveira et al. (1999)	6.6655	7.9370	2.4698

since both are SoS-based. However, our duality-based approach is substantially faster. Indeed, using the relaxation outlined in Remark 2 yields almost an order of magnitude reduction in computational time while maintaining the same performance. For completeness, we also present the result for the relaxation $r = 2$. This relaxation leads to virtually the same performance but is 200 times slower. On the other hand, Scherer and Hol (2006) with $r = 2$ introduce 161 919 variables and cannot be solved by existing SDP solvers.

To further illustrate the advantages of our approach in terms of computational complexity and scaling, consider the following discrete-time unstable system:

$$\mathbf{A} = \begin{bmatrix} -0.1660 & -0.8153 & -0.1616 & 0.3409 & 0.6015 \\ 0.4406 & -0.6275 & 0.3704 & -0.1654 & 0.9365 \\ -0.9998 & -0.3089 & -0.5911 & 0.1174 & -0.3732 \\ -0.3953 & -0.2065 & 0.7562 & -0.7192 & 0.3846 \\ -0.7065 & 0.0776 & -0.9452 & -0.6038 & 0.7528 \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} 1.7892 & 0.1701 & 0.0781 & 0.3397 & 1.7563 \\ 0.1967 & 0.8422 & 1.9158 & 1.0663 & 1.3838 \end{bmatrix}^T \tag{28}$$

For $n_s = 10$, $\epsilon = 0.05$, Algorithm 2 with (22) leads, in 9.5653 s, to a robust controller with $\mathcal{H}_2$ performance $C_{clp} = 15.4443$, $\gamma = 18.7818$ and ground truth $\gamma_0 = 15.1617$. Existing approaches cannot handle a problem of this size. Specifically, Berberich et al. (2020) introduce 10^{15} SDP constraints, Scherer and Hol (2006) introduces 227 376 variables and De Oliveira et al. (1999) lead to around 6.8×10^{12} vertices.

5.3. Robust MBC versus robust DDC

Here we provide a short discussion of the advantages of robust DDC compared with robust MBC. Consider the discrete system (27), with $n_s = 10$, $\epsilon = 0.05$. To design a robust MBC, we first obtained a nominal model using least squares. Then the uncertainty set was selected as a circle centered at the nominal model covering the consistency set. This choice is reasonable since we aim to find a robust controller and we do not have information to shrink this set. Designing a robust MBC using Scherer's P-satz yields $C_{clp} = 6.6799$, $\gamma = 8.4589$. On the other hand, Algorithm 2 leads to $C_{clp} = 6.6013$, $\gamma = 7.0531$. This is expected, since the robust DDC yields the tightest uncertainty description, while covering the uncertainty with a ball is typically conservative. It is also worth pointing out that, in many scenarios the number of samples required to find a data-driven controller is less than that for system identification (Van Waarde, Eising, Trentelman, & Camlibel, 2020). For instance, identifying (28) requires at least 7 samples while finding a data-driven controller only requires 6 samples.

6. Conclusions

This paper proposes a data-driven framework for quadratic stabilization and robust $\mathcal{H}_2$ control of unknown continuous/

discrete LTI systems. Our main result shows that using duality these problems can be recast as infinite dimensional LPs. In turn, these LPs can be relaxed to a convergent sequence of finite-dimensional SDPs, through the use of Putinar's Positivstellensatz. When compared with existing SoS-based approaches, the use of duality leads to a substantial reduction in computational complexity and asymptotic scaling. For an nth order system and relaxation order r, the approach in Scherer and Hol (2006) scales as $\mathcal{O}(n^{2r})$, while our approach scales as $\mathcal{O}(n^r)$. Further, the computational complexity of our approach grows linearly with the number of samples. For comparison LMI-based approaches based on either a multiplier description of the noise (Berberich et al., 2020) or enforcing LMIs at the vertices of the consistency set (Chesi, 2010; De Oliveira et al., 1999) scale exponentially with the number of samples and thus become impractical beyond some toy problems.

Appendix A. Proof of Theorem 1

In order to prove this theorem, we need the following preliminary results:

Lemma 2. *Given a fixed* $\mathbf{x}$ *and fixed matrices* $\mathbf{Y} \in \mathbb{R}^{n\times n}$, $\mathbf{M} \in \mathbb{R}^{m\times n}$, $\mathbf{H}_L$, $\mathbf{H}_R$, *consider the following feasibility problem in* $(\mathbf{A},\mathbf{B})$*:*

$$\begin{aligned} f(\mathbf{A},\mathbf{B}) \doteq \mathbf{x}^T \big[&\mathbf{H}_L(\mathbf{AY}+\mathbf{BM})\mathbf{H}_R \\ &+ (\mathbf{H}_L(\mathbf{AY}+\mathbf{BM})\mathbf{H}_R)^T + \mathbf{L}(\mathbf{Y})\big]\mathbf{x} \le 0 \end{aligned} \tag{A.1}$$

$$\mathbf{Tr}(\mathbf{A}\mathbf{Z}^x_{i,k} + \mathbf{B}\mathbf{Z}^u_{i,k}) \le d_{i,k} \tag{A.2}$$

where $\mathbf{L}(.)$, $\mathbf{H}_L$, $\mathbf{H}_R$ *are defined in (9). If the consistency set in Problem 1 is not empty, then (A.1)–(A.2) is infeasible if and only if there exists a* $2n_s \times n$ *positive matrix* $\boldsymbol{\Upsilon}(\mathbf{x},\mathbf{Y},\mathbf{M})$ *such that (16a)–(16b) hold.*

Proof. We will establish this result by exploiting strong duality. Consider the related minimization problem:

$$p^* = \min_{\mathbf{A},\mathbf{B}} f(\mathbf{A},\mathbf{B}) \text{ subject to (A.2)} \tag{A.3}$$

The Lagrangian corresponding to (A.3) is:

$$\begin{aligned} &\mathcal{L}(\mathbf{A},\mathbf{B},\mu) = 2\mathbf{x}^T\mathbf{H}_L(\mathbf{AY}+\mathbf{BM})\mathbf{H}_R\mathbf{x} + \mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} \\ &+ \sum_{k=1}^{n_s}\sum_{i=1}^{2n} \mu_{i,k}(\mathbf{Tr}(\mathbf{A}\mathbf{Z}^x_{i,k} + \mathbf{B}\mathbf{Z}^u_{i,k}) - d_{i,k}) = \\ &\mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} + \mathbf{Tr}\Bigg(\mathbf{A}(2\mathbf{Y}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L + \sum_{k=1}^{n_s}\sum_{i=1}^{2n}\mu_{i,k}\mathbf{Z}^x_{i,k}) \\ &+\mathbf{B}(2\mathbf{M}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L + \sum_{k=1}^{n_s}\sum_{i=1}^{2n}\mu_{i,k}\mathbf{Z}^u_{i,k})\Bigg) - \sum_{k=1}^{n_s}\sum_{i=1}^{2n}\mu_{i,k}d_{i,k} \end{aligned}$$

where $\mu_{i,k} \ge 0$ are the Lagrange multipliers and we omit its dependence on $(\mathbf{x},\mathbf{Y},\mathbf{M})$ for space reason (similarly for $\boldsymbol{\Upsilon}$). The dual function is given:

$$\begin{aligned} &g(\mu) = \inf_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\mu) = \mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} - \sum_{k=1}^{n_s}\sum_{i=1}^{2n}\mu_{i,k}d_{i,k} \\ &\quad \text{if} \quad 2\mathbf{Y}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L + \sum_{k=1}^{n_s}\sum_{i=1}^{2n}\mu_{i,k}\mathbf{Z}^x_{i,k} = 0 \\ &\qquad\quad 2\mathbf{M}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L + \sum_{k=1}^{n_s}\sum_{i=1}^{2n}\mu_{i,k}\mathbf{Z}^u_{i,k} = 0 \\ &g(\mu) = -\infty \quad \text{otherwise} \end{aligned}$$

Collecting the Lagrange multipliers $\mu_{i,k}$ in a matrix $\boldsymbol{\Upsilon}$, reshaping this matrix conformally to $\mathbf{D}$ and $\boldsymbol{\Xi}$ and using the explicit expressions for $\mathbf{Z}^x_{i,k}$, $\mathbf{Z}^u_{i,k}$, leads, after some algebra, to an equivalent compact form:

$$g(\boldsymbol{\Upsilon}) = \inf_{\mathbf{A},\mathbf{B}} L(\mathbf{A},\mathbf{B},\boldsymbol{\Upsilon}) = \begin{cases} \mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} - \mathbf{Tr}(\mathbf{D}\boldsymbol{\Upsilon}) \\ \text{if (16a) holds} \\ -\infty \text{ otherwise} \end{cases}$$

Hence the dual problem of (A.3) is given by:

$$\begin{aligned} d^* = &\max_{\boldsymbol{\Upsilon}_{i,j}\ge 0} \mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} - \mathbf{Tr}(\mathbf{D}\boldsymbol{\Upsilon}) \\ &\text{subject to (16a)} \end{aligned} \tag{A.4}$$

If the consistency set is not empty, then (A.2) is feasible (since the actual system satisfies these inequalities). Since all the inequalities involved are affine in $(\mathbf{A},\mathbf{B})$ it follows from the weak Slater's conditions that strong duality holds and $p^* = d^*$. Thus, if (A.1)–(A.2) is feasible, $p^* \le 0$, which implies that:

$$\mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} - \mathbf{Tr}(\mathbf{D}\boldsymbol{\Upsilon}) > 0,\ \ \boldsymbol{\Upsilon}_{i,j} \ge 0 \text{ and (16a)} \tag{A.5}$$

is infeasible. On the other hand, if (A.5) is feasible, then $p^* > 0$ and (A.1)–(A.2) is infeasible, i.e., (A.1)–(A.2) and (A.5) are strong alternatives. The proof is completed by noting that (A.5) holds for all $\mathbf{x}$ iff (16a)–(16b) hold for all $\|\mathbf{x}\|_2 \le 1$, by simply rescaling $\boldsymbol{\Upsilon}$ with $\|\mathbf{x}\|_2$. Hence (A.1)–(A.2) and (16a)–(16b) are strong alternatives. $\square$

Remark 4. Note that the subscripts i,k in $\mu_{i,k}$ do not correspond to row/column indexes of the matrix $\boldsymbol{\Upsilon}$. For instance, with $n = 2$, $n_s = 2$, according to the definitions in Theorem 1, we have

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{3,1} & d_{3,2} \\ d_{2,1} & d_{2,2} & d_{4,1} & d_{4,2} \end{bmatrix} \tag{A.6}$$

while the corresponding $\boldsymbol{\Upsilon}$ is

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \mu_{3,1} & \mu_{3,2} \\ \mu_{2,1} & \mu_{2,2} & \mu_{4,1} & \mu_{4,2} \end{bmatrix}^T \tag{A.7}$$

Lemma 3. *Consider the functional feasibility problem (16a)–(16b). If this problem is feasible for all* $\mathbf{x}$, $\|\mathbf{x}\|_2 \le 1$, *then it admits a continuous solution* $\boldsymbol{\Upsilon}(\mathbf{x})$.

Proof. Given fixed $\mathbf{Y}$, $\mathbf{M}$, collect all elements $\mu_{i,k}$ and $d_{i,k}$ in the matrices $\boldsymbol{\Upsilon}$, $\mathbf{D}$ and consider the following Linear Programming problem in $\hat{\boldsymbol{\Upsilon}}(\mathbf{x})$, parametric in $\mathbf{x}$:

$$\begin{aligned} &J(\mathbf{x}) \doteq \min_{\hat{\boldsymbol{\Upsilon}}(\mathbf{x})} \mathbf{Tr}(\mathbf{D}\hat{\boldsymbol{\Upsilon}}(\mathbf{x})) - \mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} \text{ subject to:} \\ &\hat{\boldsymbol{\Upsilon}}_{i,j}(\mathbf{x}) \ge 0 \text{ and (16a)} \end{aligned} \tag{A.8}$$

(A.5) is feasible if and only if the problem above is feasible, and admits a solution set $\hat{\boldsymbol{\Upsilon}}(\mathbf{x})$ such that $\mathbf{Tr}(\mathbf{D}\hat{\boldsymbol{\Upsilon}}(\mathbf{x})) - \mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} < 0$. Define the set-valued mapping $\boldsymbol{\Upsilon}_{\text{all}}(\mathbf{x}) \doteq \{\boldsymbol{\Upsilon} \in \hat{\boldsymbol{\Upsilon}} : \mathbf{Tr}(\mathbf{D}\boldsymbol{\Upsilon}) - \mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x} \le J(\mathbf{x})\}$. From Theorem 2.4 in Mangasarian and Shiau (1987) establishing continuity of the solutions of linear programs with respect to perturbations in the right hand side, it follows that $\boldsymbol{\Upsilon}_{\text{all}}(\mathbf{x})$ is lower semi-continuous. Consider now the minimum selection

$$\boldsymbol{\Upsilon}(\mathbf{x}) \doteq \mathop{argmin}_{\boldsymbol{\Upsilon}\in\boldsymbol{\Upsilon}_{\text{all}}(\mathbf{x})} \|\boldsymbol{\Upsilon}\|_F$$

Since $\mathbf{Y}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L$ and $\mathbf{M}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L$ are bounded in $\|\mathbf{x}\|_2 \le 1$, it follows that the range of $\boldsymbol{\Upsilon}(.)$ is bounded. Hence, from Proposition 9.3.2 in Aubin and Frankowska (2009), it follows that the function $\boldsymbol{\Upsilon}(\mathbf{x})$ is continuous. The proof is completed by noting that, by construction $\boldsymbol{\Upsilon}(\mathbf{x})$ solves the original problem (16a)–(16b). $\square$

Proof of Theorem 1. We will proceed as follows:

(1) Show that the matrix $\Upsilon(\mathbf{x},\mathbf{Y},\mathbf{M})$ can be taken to be independent of $\mathbf{Y}$ and $\mathbf{M}$.
(2) Show that if (16a)–(16b) are feasible, they admits a polynomial solution $\Upsilon^p(\mathbf{x})$, with $\Upsilon^p_{i,j}(\mathbf{x})>0$.

Begin by noting that since the quadratic form

$$\mathbf{x}^T\left[\Omega+\Omega^T+\mathbf{L}(\mathbf{Y})\right]\mathbf{x}$$

is homogeneous in $\mathbf{x}$, (15) is equivalent to:

$$-\mathbf{x}^T\left[\Omega+\Omega^T+\mathbf{L}(\mathbf{Y})\right]\mathbf{x}<0;\ \forall\mathbf{x}\neq 0,\ \|\mathbf{x}\|_2\leq 1$$

or, equivalently, infeasibility of

$$-\mathbf{x}^T\left[\Omega+\Omega^T+\mathbf{L}(\mathbf{Y})\right]\mathbf{x}\geq 0;\ \forall\mathbf{x}\neq 0,\ \|\mathbf{x}\|_2\leq 1$$

From Lemma 2 we have that Problem 3 is feasible if there exist matrices $\mathbf{Y}\succ 0$, $\mathbf{M}$ and $\Upsilon(\mathbf{x},\mathbf{Y},\mathbf{M})>0$ such that (A.5) holds for all $\|\mathbf{x}\|_2\leq 1$. Since we are interested in finding just one pair $(\mathbf{Y},\mathbf{M})$, it follows that the vector functions Υ can be taken to be independent of $\mathbf{Y}$ and $\mathbf{M}$. To see this, assume that there exist some $\mathbf{Y}^*,\mathbf{M}^*,\tilde{\Upsilon}(\mathbf{x},\mathbf{Y}^*,\mathbf{M}^*)$ such that (A.5) holds for all $\|\mathbf{x}\|_2\leq 1$. Then, setting $\Upsilon(\mathbf{x})\doteq\tilde{\Upsilon}(\mathbf{x},\mathbf{Y}^*,\mathbf{M}^*)$ we have that (A.5) also holds for $\mathbf{Y}=\mathbf{Y}^*,\mathbf{M}=\mathbf{M}^*$ and all $\|\mathbf{x}\|_2\leq 1$.

Next, we will show that if the problem is feasible, it always admits a polynomial solution $\Upsilon^p(\mathbf{x})$. Assume that Ξ has full row rank and let $\mathcal{N}$ be a basis of its (right) null space. Denote by $\Upsilon^*(.)$ a feasible solution to (16a). Then $\Upsilon^*(.)$ can always be written as

$$\Upsilon^*(\mathbf{x})=-2\Xi^T(\Xi\Xi^T)^{-1}\begin{bmatrix}\mathbf{Y}\\ \mathbf{M}\end{bmatrix}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L+\mathcal{N}\mathbf{Z}_n(\mathbf{x})$$

for some continuous $\mathbf{Z}_n(\mathbf{x})$. Since $\mathbf{Z}_n(\mathbf{x})$ is continuous, from Stone–Weierstrass theorem it follows that there exist a polynomial $\mathbf{Z}_p(\mathbf{x})$ such that $\|\mathbf{Z}_n(\mathbf{x})-\mathbf{Z}_p(\mathbf{x})\|_\infty\leq\delta_1$ for all $\|\mathbf{x}\|_2\leq 1$. Consider now the following polynomial matrix

$$\Upsilon^p(\mathbf{x})=-2\Xi^T(\Xi\Xi^T)^{-1}\begin{bmatrix}\mathbf{Y}\\ \mathbf{M}\end{bmatrix}\mathbf{H}_R\mathbf{x}\mathbf{x}^T\mathbf{H}_L+\mathcal{N}\mathbf{Z}_p(\mathbf{x})+\delta_2\mathbf{1} \tag{A.9}$$

We will show that δ_2 can always be chosen so that $\Upsilon^p(\mathbf{x})$ is also a feasible solution of (A.5). Since $\Xi\mathbf{1}=\mathbf{0}$, then, by construction $\Upsilon^p(\mathbf{x})$ satisfies (16a). To show that the elements of Υ^p can be made positive by a suitable choice of δ_2, note that

$$\Upsilon^p(\mathbf{x})=\Upsilon^*(\mathbf{x})-\mathcal{N}(\mathbf{Z}_n(\mathbf{x})-\mathbf{Z}_p(\mathbf{x}))+\delta_2\mathbf{1}$$

Hence

$$\Upsilon^p_{i,j}(\mathbf{x})\geq\Upsilon^*_{i,j}(\mathbf{x})-\delta_1\|\mathcal{N}\|_\infty+\delta_2>0 \text{ if } \delta_2>\delta_1\|\mathcal{N}\|_\infty$$

Now, let $\delta_m\doteq\max_{\|\mathbf{x}\|_2\leq 1}\mathbf{Tr}(\mathbf{D}\Upsilon^*(\mathbf{x}))-\mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x}$. Since Υ^* is a feasible solution of (A.5), $\delta_m<0$. Thus

$$\begin{aligned}\mathbf{Tr}(\mathbf{D}\Upsilon^p(\mathbf{x}))\leq&\mathbf{Tr}(\mathbf{D}\Upsilon^*(\mathbf{x}))+\delta_1\sum_i\sum_j|\mathbf{D}\mathcal{N}|_{i,j}\\&+\delta_2\sum_i\sum_j|\mathbf{D}_{ij}|<\mathbf{x}^T\mathbf{L}(\mathbf{Y})\mathbf{x}\end{aligned}$$

if δ_1 and δ_2 are selected such that

$$\delta_1\sum_i\sum_j|\mathbf{D}^T\mathcal{N}|_{i,j}+\delta_2\sum_i\sum_j|\mathbf{D}_{ij}|<|\delta_m| \tag{A.10}$$

It follows that the polynomial matrix Υ^p is also a feasible solution to (A.5) □

Appendix B. Proof of Theorem 2

(Only a sketch given, due to space constraints). Given a fixed $\mathbf{x}$ and fixed matrices $\mathbf{Y}\in\mathbb{R}^{n\times n}$, $\mathbf{M}\in\mathbb{R}^{m\times n}$, $\mathbf{W}\in\mathbb{R}^{n\times n}$ and a fixed γ, it can be shown (using the same arguments used in the proof of Lemma 2) that

$$\mathbf{x}^TC(\gamma,\mathbf{Y},\mathbf{M},\mathbf{W},\mathbf{A},\mathbf{B})\mathbf{x}\leq 0 \tag{B.1}$$

$$\mathbf{Tr}(\mathbf{A}\mathbf{Z}^x_{i,k}+\mathbf{B}\mathbf{Z}^u_{i,k})-d_{i,k}\leq 0 \tag{B.2}$$

and (20a)–(20b) are strong alternatives. The proof is completed by noting that (20b) holds for all $\mathbf{x}$ iff it holds for all $\mathbf{x}$ with $\|\mathbf{x}\|_2\leq 1$. Finally, the proof that the entries of $\Upsilon(\mathbf{x})$ are positive polynomials follows along the same arguments used in the proof of Theorem 1. □

References

Aubin, J.-P., & Frankowska, H. (2009). *Set-valued analysis*. Springer Science & Business Media.

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

Berberich, Julian, Scherer, Carsten W., & Allgöwer, Frank (2020). Combining prior knowledge and data for robust controller design. arXiv preprint arXiv:2009.05253.

Bisoffi, Andrea, De Persis, Claudio, & Tesi, Pietro (2022). Data-driven control via Petersen's lemma. *Automatica*, *145*, Article 110537.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Chen, J., & Gu, G. (2000). *Control oriented system identification: an $\mathcal{H}_\infty$ approach*. Wiley & Sons, Inc..

Chesi, Graziano (2010). LMI techniques for optimization over polynomials in control: a survey. *IEEE Transactions on Automatic Control*, *55*(11), 2500–2510.

Dai, Tianyu, & Sznaier, Mario (2018a). Data driven robust superstable control of switched systems. *IFAC-PapersOnLine*, *51*(25), 402–408.

Dai, Tianyu, & Sznaier, Mario (2018b). A moments based approach to designing MIMO data driven controllers for switched systems. In *2018 IEEE conference on decision and control* (pp. 5652–5657). IEEE.

Dai, Tianyu, & Sznaier, Mario (2020). Data-driven quadratic stabilization of continuous LTI systems. *IFAC-PapersOnLine*, *53*(2), 3965–3970.

De Oliveira, Maurício C, Bernussou, Jacques, & Geromel, José C (1999). A new discrete-time robust stability condition. *Systems & Control Letters*, *37*(4), 261–265.

De Persis, Claudio, & Tesi, Pietro (2019). On persistency of excitation and formulas for data-driven control. arXiv preprint arXiv:1903.06842.

Feron, Eric, Balakrishnan, Venkataramanan, Boyd, Stephen, & Ghaoui, Laurent El (1992). Numerical methods for H2 related problems. In *American control conference* (pp. 2921–2922). IEEE.

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

Löfberg, Johan (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD conference, Vol. 3*. Taipei, Taiwan.

Mangasarian, O. L., & Shiau, T.-H. (1987). Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems. *SIAM Journal on Control and Optimization*, *25*(3), 583–595.

MATLAB (2020). *9.9.0.1524771 (R2020b)*. Natick, Massachusetts: The MathWorks Inc..

MOSEK (2019). The MOSEK optimization toolbox for MATLAB manual. Version 9.0.. URL http://docs.mosek.com/9.0/toolbox/index.html.

Putinar, Mihai (1993). Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, *42*(3), 969–984.

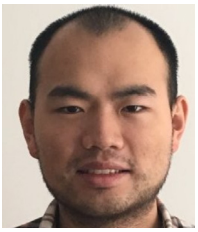
Sánchez Peña, R., & Sznaier, M. (1998). *Robust systems theory and applications*. Wiley & Sons, Inc..

Scherer, Carsten W., & Hol, Camile W. J. (2006). Matrix sum-of-squares relaxations for robust semi-definite programs. *Mathematical Programming*, *107*(1), 189–211.

Van Waarde, Henk J., Eising, Jaap, Trentelman, Harry L., & Camlibel, M. Kanat (2020). Data informativity: a new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, *65*(11), 4753–4768.

Vanwaarde, Henk J., Camlibel, M. Kanat, & Mesbahi, Mehran (2020). From noisy data to feedback controllers: non-conservative design via a matrix S-lemma. *IEEE Transactions on Automatic Control*.

Zhou, Kemin, & Doyle, John Comstock (1998). *Essentials of robust control, Vol. 104*. Upper Saddle River, NJ: Prentice Hall.



Tianyu Dai received his B.S. degree in Optical information of Science and Technology from Harbin Institute of Technology (Weihai), China in 2014. He received his Ph.D. degree in Electrical Engineering at Northeastern University, United States in 2022. His research interests lie in data-driven control and convex optimization.

Mario Sznaier is currently the Dennis Picard Chaired Professor at the Electrical and Computer Engineering Department, Northeastern University, Boston. Prior to joining Northeastern University, Dr. Sznaier was a Professor of Electrical Engineering at the Pennsylvania State University and also held visiting positions at the California Institute of Technology. His research interest include robust identification and control of hybrid systems, robust optimization, and dynamical vision. Dr. Sznaier is currently serving as Editor in Chief of the section on AI and Control of the journal Frontiers in Control Engineering, and chair of IFAC's Technical Committee TC 2.5, Additional recent service includes Associate Editor for the Journal Automatica (2005–2021), chair of the CSS Technical Committee on Computational Aspects of Control System Design (2012–2016), General co-Chair of the 2016 MSC and Program Chair of the 2017 CDC. He is a distinguished member of the IEEE Control Systems Society and an IEEE Fellow.