A Semi-Algebraic Optimization Approach to Data-Driven Control of Continuous-Time Nonlinear Systems

T. Dai, *Graduate Student Member, IEEE*, and M. Sznaier, *Fellow, IEEE*

***Abstract*—This letter considers the problem of designing state feedback data-driven controllers for nonlinear continuous-time systems. Specifically, we consider a scenario where the unknown dynamics can be parametrized in terms of known basis functions and the available measurements are corrupted by unknown-but-bounded noise. The goal is to use this noisy experimental data to directly design a rational state-feedback control law guaranteed to stabilize all plants compatible with the available information. The main result of this letter shows that, by using Rantzer's Dual Lyapunov approach, combined with elements from convex analysis, the problem can be recast as an optimization over positive polynomials, which can be relaxed to a semidefinite program through the use of Sum-of-Squares and semi-algebraic optimization arguments. Three academic examples are considered to illustrate the effectiveness of the proposed method.**

***Index Terms*—Robust control, uncertain systems.**

I. Introduction

DATA-DRIVEN control (DDC), that is the design of controllers directly from observed data, has attracted substantial attention in recent years due to its advantages over model-based control (MBC). In general, DDC is less conservative, since by avoiding the model identification step, it bypasses practically difficult questions such as model order/class selection and potential inaccuracy of the identification. Earlier work on nonlinear DDC can be roughly split into three categories: a) Reference model-based method. The idea is to assume a reference model that gives the desired closed-loop performance and the goal of the algorithm is to find a controller that minimizes the error between the true and reference signal. Some pioneering work along these lines includes the virtual reference feedback tuning (VRFT) [1], [2] and iterative feedback tuning (IFT) [3]. These methods work well for the tracking problem, however, issues like guaranteeing stability when only data collected over a finite horizon is used to design the controller, and an automatic way of selecting the reference model and the candidate controller are still open. b) Linearization-based methods. The main idea behind these approaches is to assume that the dynamics of the system are locally represented by a linearized model. Estimation of the Jacobian of the system leads to controllers that locally stabilize the system. This idea is used for instance in model-free adaptive control (MFAC) [4] and lazy learning (LL) [5]. The main drawback of this approach is that global behavior is in general not guaranteed. c) Eigenfunction-based methods. Interest in these methods has recently surged due to the connection with Neural Nets. The basic idea is to pre-define a basis of eigenfunctions that span the trajectories of the nonlinear system and to expand the dynamics in terms of these functions. Typically these eigenfunctions are chosen to optimize the fit of the collected data. Prior work of this type includes Koopman eigenfunction [6], [7] and data-driven inversion-based control (D^2-IBC) [8]. These ideas are very appealing since, in this context, it is possible to handle the different types of nonlinearities once a suitable basis is selected. However, the issue of selecting a dictionary that avoids overfitting is still open. Further, while eigenfunctions based approaches have been very successful in predicting future values of the state, at the present, they can't provide stability certificates when used to design a controller.

The present paper seeks a rapprochement between traditional Lyapunov based methods and eigenfunction based DDC. Our goal is, given a nonlinear system parameterized in terms of a known set of basis functions (dictionary) and experimental measurements corrupted by unknown-but-bounded noise, to synthesize a controller guaranteed to stabilize (in a sense to be precisely defined later) all possible plants compatible with both the a-priori information and the experimental data. The main idea is to use the collected (noisy) data, to bound the set of plants to a polytope in parameter space (the uncertain set). In principle, ideally, the goal is to find a common control Lyapunov function (CCLF) and associated controller for all elements in the set. However, this leads to very challenging non-convex problems.[1] As an alternative, we will consider

[1] Recall that, even for a known plant, the set of stabilizing controllers may not even be connected.

Manuscript received March 17, 2020; revised May 14, 2020; accepted June 7, 2020. Date of publication June 18, 2020; date of current version July 2, 2020. This work was supported in part by NSF under Grant CNS–1646121, Grant CMMI–1638234, Grant IIS–1814631, and Grant ECCS–1808381, in part by AFOSR under Grant FA9550-19-1-0005, and in part by the Alert DHS Center of Excellence under Award 2013-ST-061-ED0001. Recommended by Senior Editor M. Arcak. *(Corresponding author: M. Sznaier.)*

The authors are with the ECE Department, Northeastern University, Boston, MA 02115 USA (e-mail: dai.ti@husky.neu.edu; msznaier@coe.neu.edu).

Digital Object Identifier 10.1109/LCSYS.2020.3003505

the slightly weaker condition introduced in Rantzer's dual Lyapunov theory [9] that only requires all trajectories (except those starting in a set of measure zero) to converge to the origin as $t \to \infty$, and can be expressed in terms of the existence of a positive density $\rho(\mathbf{x})$ satisfying a divergence-type condition. Our main results, motivated by the approach proposed in [10], [11] to solve a DDC problem for LTI systems, shows that combining this characterization of stability with the Farkas lemma [12] allows for recasting the DDC problem into a semi-algebraic feasibility one. In turn, this problem can be relaxed, proceeding as in [13], to a Sum-of-Squares (SoS) optimization leading to a convex Semi-Definite program (SDP) via semi-algebraic optimization arguments.

II. Preliminaries

A. Notation

$\mathbb{R}, (\mathbb{R}^+)$	set of (non-negative) real numbers
$\mathbf{1}$	a vector of 1s
$\mathbf{x}, \mathbf{X}$	a vector in $\mathbb{R}^n$, a matrix in $\mathbb{R}^{m\times n}$
x_i	the i^{th} element of the vector $\mathbf{x}$
$\|\mathbf{x}\|_\infty$	ℓ_∞-norm of the vector $\mathbf{x} \in \mathbb{R}^n$
$\|\mathbf{x}\|_1$	ℓ_1-norm of the vector $\mathbf{x} \in \mathbb{R}^n$
$\|\mathbf{X}\|_1$	$\ell_\infty \to \ell_\infty$ induced norm of the matrix $\mathbf{X}$.
$\sigma_i(\mathbf{X})$	i^{th} largest singular value of a matrix
$\mathbf{X} \geq 0$	$\mathbf{X}$ is element-wise non-negative (e.g., $\mathbf{X}_{i,j} \geq 0$)
$\mathbf{X} \succeq 0$	$\mathbf{X}$ is positive semi-definite
$\mathbf{vec}(\mathbf{X})$	matrix vectorizing operation along columns $\mathbf{vec}(\mathbf{X}) = \left[\mathbf{X}(:,1)^T, \ldots \mathbf{X}(:,n)^T\right]^T$
$\otimes$	matrix Kronecker product
$f \in \mathbf{C}^d$	the d^{th} derivative of f exists and is continuous
∇V	the gradient of a scalar function: $\nabla V = [\frac{\partial V}{\partial x_1} \ \cdots \ \frac{\partial V}{\partial x_n}]$
$\nabla \cdot f$	the divergence of a vector-valued function: $\nabla \cdot f = \frac{\partial f_1}{\partial x_1} + \ldots + \frac{\partial f_n}{\partial x_n}$
$\Sigma[\mathbf{x}]$	cone of SoS polynomials in $\mathbf{x}$.

B. Dual Lyapunov Theorem

The following theorem plays a central role in the formulation of a tractable problem.

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

$$[\nabla \cdot (\rho(f + gu)](\mathbf{x}) > 0 \text{ for almost all } \mathbf{x} \neq 0 \tag{1}$$

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

A detailed proof can be found in [9].

C. Farkas Lemma

We will use the following variant of Farkas' Lemma:

Lemma 1: Given $\mathbf{N} \in \mathbb{R}^{m\times n}$, $\mathbf{d} \in \mathbb{R}^n$, $\mathbf{e} \in \mathbb{R}^m$, assume that $\mathbf{N}\mathbf{x} \leq \mathbf{e}$ is feasible. Then the inequalities:

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \text{ with } \mathbf{A}^T = \begin{bmatrix}\mathbf{N}^T & -\mathbf{d}\end{bmatrix} \text{ and } \mathbf{b}^T = \begin{bmatrix}\mathbf{e}^T & 0\end{bmatrix} \tag{2}$$

and

$$\mathbf{N}^T\mathbf{y} = \mathbf{d},\ \mathbf{e}^T\mathbf{y} < 0, \text{ and } \mathbf{y} \geq 0,\ \mathbf{y} \in \mathbb{R}^m \tag{3}$$

are strong alternatives, that is exactly only one set is feasible.

Proof: Follows from applying Farkas' Lemma (e.g., [12, Sec. 5.8]) to (2) with a multiplier of the form $\mathbf{y}_a^T \doteq \begin{bmatrix}\mathbf{y}_m^T & y_{m+1}\end{bmatrix}$. Since feasibility of $\mathbf{N}\mathbf{x} \leq \mathbf{e}$ implies that $y_{m+1} > 0$, then, without loss of generality, we can take $\mathbf{y}_a^T \doteq \begin{bmatrix}\mathbf{y}_m^T & 1\end{bmatrix}$. ■

D. Property of the Kronecker Product

The following property, whose proof can be found in [14], will be used extensively in this letter.

$$\mathbf{vec}(\mathbf{B}^T\mathbf{X}^T\mathbf{A}^T) = (\mathbf{A} \otimes \mathbf{B}^T)\mathbf{vec}(\mathbf{X}^T). \tag{4}$$

E. Polynomial Optimization Problems

The main result of this letter shows that the DDC problem can be reduced to a (non-convex) polynomial optimization problem and solved using moments-based techniques. For ease of reference, the key ideas of this approach are outlined below. Consider the polynomial optimization problem of the form:

$$\begin{aligned} p^* &= \min_{\mathbf{x}\in\mathcal{K}} p(\mathbf{x}) = \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} \\ \mathcal{K} &= \{\mathbf{x} \in \mathbb{R}^n : g_k(\mathbf{x}) \geq 0,\ k = 1, \ldots, N\} \end{aligned} \tag{5}$$

where $\boldsymbol{\alpha} \doteq [\alpha_1, \ldots, \alpha_n]$, $\mathbf{x}^{\boldsymbol{\alpha}} = \prod_{i=1}^n x_i^{\alpha_i}$, $p_{\boldsymbol{\alpha}}$ are the coefficient of $p(\mathbf{x})$ and $\mathcal{K}$ is a semi-algebraic set defined by the polynomial constraints $g_k(\mathbf{x}) = \sum_{\boldsymbol{\alpha}} g_{k,\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} \geq 0$. As shown in [15], problem (5) is equivalent to the following optimization over the set $\mathcal{P}(\mathcal{K})$ of probability measures μ supported on $\mathcal{K}$:

$$p^* = \min_{\mu\in\mathcal{P}(\mathcal{K})} \int p(\mathbf{x})\mu(d\mathbf{x}) = \min_{\mu} \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} m_{\boldsymbol{\alpha}} \tag{6}$$

$$\text{subject to } m_{\boldsymbol{\alpha}} \doteq \int_{\mathcal{K}} \mathbf{x}^{\boldsymbol{\alpha}}\mu(d\mathbf{x}) \tag{7}$$

where $m_{\boldsymbol{\alpha}}$ denotes the $\boldsymbol{\alpha}^{\text{th}}$ moment with respect to μ. If the quadratic module generated by the constraints $g_k(\mathbf{x})$ is Archimedean (for instance if $\exists a \mid \mathcal{K} \subseteq \{\mathbf{x} \in \mathbb{R}^n\colon \|\mathbf{x}\|_2^2 \leq a^2\}$) [16], (7) is equivalent to a set of constraints of the form:

$$\mathbf{M}(m_{\boldsymbol{\alpha}}) \succeq 0,\ \mathbf{L}(g_k m_{\boldsymbol{\alpha}}) \succeq 0, k = 1, \ldots N \tag{8}$$

where the entries of the (infinite dimensional) moment $\mathbf{M}(m_{\boldsymbol{\alpha}})$ and localization matrices $\mathbf{L}(g_k m_{\boldsymbol{\alpha}})$, are given by

$$\begin{aligned} \mathbf{M}(m_{\boldsymbol{\alpha}})(i,j) &= m_{\boldsymbol{\alpha}^{(i)}+\boldsymbol{\alpha}^{(j)}} \\ \mathbf{L}(g_k m_{\boldsymbol{\alpha}})(i,j) &= \sum_{\boldsymbol{\beta}} g_{k,\boldsymbol{\beta}}\ m_{\boldsymbol{\beta}+\boldsymbol{\alpha}^{(i)}+\boldsymbol{\alpha}^{(j)}},\quad k = 1, \ldots, N \end{aligned} \tag{9}$$

where $g_{k,\boldsymbol{\beta}}$ are the coefficients of the k^{th} polynomial that defines the set $\mathcal{K}$. Thus, Problem (6)-(7) is convex in $m_{\boldsymbol{\alpha}}$, albeit infinite dimensional. Under the Archimedean assumption, a convergent sequence of finite dimensional convex relaxations with cost $p_{\boldsymbol{m}}^d \uparrow p^*$ can be obtained by replacing the matrices in (8) by truncated matrices $\mathbf{M}_d(m_{\boldsymbol{\alpha}})$, $\mathbf{L}_d(g_k m_{\boldsymbol{\alpha}})$ containing moments of order up to 2d. Further, if for some d the solution to the problem above satisfies

$$\text{rank}\,[\mathbf{M}_d(m_{\boldsymbol{\alpha}})] = \text{rank}\big(\mathbf{M}_{d-\max(\deg(g_k(\mathbf{x})))}\big) \tag{10}$$

then the relaxation is exact, that is $p_{\mathbf{m}}^d = p^*$ [15].

Remark 1: Of particular interest to this letter is the Quadratically Constrained Quadratic Programming (QCQP) case where both the objective and constraints are quadratic polynomials. In this case, the lowest order relaxation of (10) corresponds to $d = 1$, with objective and localizing matrices given by Trace($\mathbf{Q}_o\mathbf{M}_1$) and Trace($\mathbf{Q}_k\mathbf{M}_1$) respectively. If the solution to this relaxation satisfies rank ($\mathbf{M}_1$) = 1, it can be easily shown that it is indeed exact. We will exploit this property in Section III to obtain a computationally tractable algorithm to synthesize data-driven controllers.

F. Problem Statement

In this letter we consider continuous-time control-affine non-linear systems of the form[2]:

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

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

Our goal is to find a rational control action that stabilizes the system in the dual Lyapunov sense discussed in the introduction. Formally, the problem can be stated as:

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

III. Main Results

In this section, we present a convex reformulation of the Problem 1. As mentioned in the introduction, the main idea is to consider two sets in parameter space: a) an uncertainty set containing all possible plants compatible with the noisy data, b) the set of plants where there exists a density ρ and associated control action satisfying Theorem 1. These two sets will be related through the Farkas lemma to obtain a condition for the existence of a data-driven controller guaranteed to stabilize all plants in the uncertainty set. Then, this condition will be relaxed to a semi-definite optimization subject to SoS constraints and reduced to a convex semi-definite program (SDP) using moments-based techniques.

A. An SoS Reformulation of the Problem

Given the collected data $\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}$ and a bound ϵ on the noise, define the consistency set $\mathcal{P}_1$ as the set of all pairs of $\mathbf{F}, \mathbf{G}$ compatible with the noisy data. Using (11) $\mathcal{P}_1$ is given by:

$$\mathcal{P}_1 \doteq \{\mathbf{F}, \mathbf{G} : \|\dot{\mathbf{x}} - \mathbf{F}\boldsymbol{\phi} - \mathbf{G}\boldsymbol{\gamma}u\|_\infty \leq \epsilon,\ t = 1\ldots T\} \tag{12}$$

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

Direct application of (4) to the data collected in $[1, T]$ leads to the equivalent expression:

$$\mathcal{P}_1 \doteq \left\{\mathbf{f}, \mathbf{g} : \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ -\mathbf{A} & -\mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix} \leq \begin{bmatrix}\epsilon\mathbf{1} + \boldsymbol{\xi}\\ \epsilon\mathbf{1} - \boldsymbol{\xi}\end{bmatrix}\right\} \tag{13}$$

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

$$\mathbf{A} \doteq \begin{bmatrix}\mathbf{I}\otimes\boldsymbol{\phi}^T(1)\\ \vdots\\ \mathbf{I}\otimes\boldsymbol{\phi}^T(T)\end{bmatrix}, \mathbf{B} \doteq \begin{bmatrix}\mathbf{I}\otimes u(1)\boldsymbol{\gamma}^T(1)\\ \vdots\\ \mathbf{I}\otimes u(T)\boldsymbol{\gamma}^T(T)\end{bmatrix}, \boldsymbol{\xi} \doteq \begin{bmatrix}\dot{\mathbf{x}}(1)\\ \vdots\\ \dot{\mathbf{x}}(T)\end{bmatrix} \tag{14}$$

In the sequel, we will make the following assumption:

Assumption 1: Enough data has been collected so that the matrix $\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ -\mathbf{A} & -\mathbf{B}\end{bmatrix}$ has full column rank (e.g., the polytope $\mathcal{P}_1$ is compact). Note that if this assumption fails the diameter of the consistency set is infinite and thus the worst case identification error of any interpolatory identification algorithm is unbounded. A detailed discussion on this can be found in [17, Ch. 10].

In order to obtain tractable problems, proceeding as in [13], in this letter we will consider rational densities $\rho(\mathbf{x})$ and control actions $u(\mathbf{x})$ of the form

$$\rho(\mathbf{x}) = \frac{a(\mathbf{x})}{b(\mathbf{x})^\lambda},\ u(\mathbf{x})\rho(\mathbf{x}) = \frac{c(\mathbf{x})}{b(\mathbf{x})^\lambda}, b(\mathbf{x}) > 0 \tag{15}$$

where $a(\mathbf{x}), b(\mathbf{x}), c(\mathbf{x})$ are polynomials and λ is chosen to satisfy the integrability condition in Theorem 1. Note that for plants of the form (11) and densities of the form (15), (1) is equivalent to:

$$b\nabla\cdot(\mathbf{F}\boldsymbol{\phi}a + \mathbf{G}\boldsymbol{\gamma}c) - \lambda\nabla b\cdot(\mathbf{F}\boldsymbol{\phi}a + \mathbf{G}\boldsymbol{\gamma}c) > 0 \tag{16}$$

where we explicitly used the fact that $b(\mathbf{x}) > 0$. Given (a, b, c), let $\mathcal{P}_2(a, b, c)$ denote the set of all plants of the form (11) that can be stabilized by a control action of the form (15), that is

$$\mathcal{P}_2(a, b, c) \doteq \{\mathbf{F}, \mathbf{G} : (16) \text{ holds for given polynomials } a, b, c\ \} \tag{17}$$

In this context, Problem 1 reduces to

Problem 2: Find polynomials $a(\mathbf{x}), b(\mathbf{x}), c(\mathbf{x})$ such that $\mathcal{P}_1 \subseteq \mathcal{P}_2(a, b, c)$.

The first step towards enforcing the inclusion above is to obtain a representation of $\mathcal{P}_2$ in terms of $\mathbf{f}, \mathbf{g}$. To this effect, define $a = \mathbf{c}_a^T\mathbf{m}$, $b = \mathbf{c}_b^T\mathbf{m}$, $c = \mathbf{c}_c^T\mathbf{m}$, where $\mathbf{c}_a, \mathbf{c}_b, \mathbf{c}_c$ are coefficient vectors and

$$\mathbf{m} \doteq \left[1\ x_1\ldots(x_1^{\alpha_1}\ldots x_n^{\alpha_n})\ldots x_n^m\right]^T \tag{18}$$

is a vector of monomials ordered in a graded reversed (grevlex) order of the corresponding size. Substituting these definitions in (16) leads to:

$$\begin{aligned}&\mathbf{c}_b^T\mathbf{m}\nabla\cdot(\mathbf{c}_a^T\mathbf{m}\mathbf{F}\boldsymbol{\phi} + \mathbf{c}_c^T\mathbf{m}\mathbf{G}\boldsymbol{\gamma})\\ &\quad - \alpha\nabla(\mathbf{c}_b^T\mathbf{m})\ (\mathbf{c}_a^T\mathbf{m}\mathbf{F}\boldsymbol{\phi} + \mathbf{c}_c^T\mathbf{m}\mathbf{G}\boldsymbol{\gamma}) > 0\end{aligned} \tag{19}$$

Explicitly computing the first term in (19) yields:

$$\begin{aligned}&\mathbf{c}_b^T\mathbf{m}[\mathbf{c}_a^T\frac{\partial\mathbf{m}\boldsymbol{\phi}^T}{\partial x_1}\ldots\mathbf{c}_a^T\frac{\partial\mathbf{m}\boldsymbol{\phi}^T}{\partial x_n}\\ &\quad\mathbf{c}_c^T\frac{\partial\mathbf{m}\boldsymbol{\gamma}^T}{\partial x_1}\ldots\mathbf{c}_c^T\frac{\partial\mathbf{m}\boldsymbol{\gamma}^T}{\partial x_n}]\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix} \doteq \mathbf{d}_1(\mathbf{x})\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix}\end{aligned} \tag{20}$$

Similarly, the second term in (19) is given by:

$$-\lambda[\mathbf{c}_b^T \frac{\partial \mathbf{m}}{\partial x_1}\mathbf{c}_a^T \mathbf{m}\boldsymbol{\phi}^T \dots \mathbf{c}_b^T \frac{\partial \mathbf{m}}{\partial x_n}\mathbf{c}_a^T \mathbf{m}\boldsymbol{\phi}^T$$
$$\mathbf{c}_b^T \frac{\partial \mathbf{m}}{\partial x_1}\mathbf{c}_c^T \mathbf{m}\boldsymbol{\gamma}^T \dots \mathbf{c}_b^T \frac{\partial \mathbf{m}}{\partial x_n}\mathbf{c}_c^T \mathbf{m}\boldsymbol{\gamma}^T]\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix} \doteq \mathbf{d}_2(\mathbf{x})\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix} \quad (21)$$

Rewriting $\mathcal{P}_2$ in terms of these expressions leads to:

$$\mathcal{P}_2 \doteq \{\mathbf{f}, \mathbf{g} : -\big[\mathbf{d}_1(\mathbf{x}) + \mathbf{d}_2(\mathbf{x})\big]\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix} < 0\} \quad (22)$$

Note that the inequalities above are functional inequalities that must hold for all $\mathbf{x}$ and some coefficient vectors $\mathbf{c}_a, \mathbf{c}_b, \mathbf{c}_c$.

Theorem 2: Given data collected from trajectories of (11) in the interval $t \in [1, T]$, form the corresponding matrices $\mathbf{A}, \mathbf{B}, \boldsymbol{\xi}$, defined in (14). Then, there exist polynomials $a, b > 0, c$ such that $\mathcal{P}_1 \subseteq \mathcal{P}_2$(a,b,c) if and only if there exists a vector function $\mathbf{Y}(\mathbf{x}) \in \mathbb{R}^{1\times 2nT} \geq 0$ such that the following (functional) set of affine constraints is feasible:

$$\mathbf{Y}(\mathbf{x})\mathbf{N} = \mathbf{d}(\mathbf{x}) \text{ and } \mathbf{Y}(\mathbf{x})\mathbf{e} < 0 \quad (23)$$

where for notational simplicity we defined

$$\mathbf{N} \doteq \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ -\mathbf{A} & -\mathbf{B}\end{bmatrix},\ \mathbf{e} = \begin{bmatrix}\epsilon\mathbf{1} + \boldsymbol{\xi}\\ \epsilon\mathbf{1} - \boldsymbol{\xi}\end{bmatrix}$$
$$\mathbf{d} \doteq -\big[\mathbf{d}_1(\mathbf{x}) + \mathbf{d}_2(\mathbf{x})\big], \quad (24)$$

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

Proof: Applying Lemma 1 pointwise in $\mathbf{x}$ to (23) shows that existence of $\mathbf{Y}(\mathbf{x})$ is necessary and sufficient for $\mathcal{P}_1 \subseteq \mathcal{P}_2$. Continuity of $\mathbf{Y}(.)$ is shown in the Appendix, using the minimal selection theorem [18]. To prove (ii), assume that there exist some continuous vector function $\mathbf{Y}(\mathbf{x}) \geq 0$ such that (23) holds and that the trajectories of the closed-loop system never leave a compact domain $\mathcal{D} \subset \mathbb{R}^n$. Let $\delta^+ \doteq \max_{\mathbf{x}\in\mathcal{D}} \mathbf{Y}(\mathbf{x})\mathbf{e}$. Since $\mathbf{Y}(.)$ is continuous and $\mathcal{D}$ is compact, $\delta^+ < 0$. From Assumption 1, $\mathbf{Y}(\mathbf{x})$ can be written as:

$$\mathbf{Y}(\mathbf{x}) = \mathbf{Y}_o(\mathbf{x}) + \mathbf{z}(\mathbf{x})\mathbf{U} \text{ where } \mathbf{Y}_o(\mathbf{x}) \doteq \mathbf{d}(\mathbf{x})(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T$$

for some continuous $\mathbf{z}(\mathbf{x})$, where $\mathbf{U}$ is a basis for the (left) null space of $\mathbf{N}$. Since the trajectories of the closed-loop system never leave $\mathcal{D} \subset \mathbb{R}^n$, from Stone-Weierstrass theorem [19], there exist some polynomial vector $\mathbf{z}_p(\mathbf{x})$ such that $\|\mathbf{z}_p(\mathbf{x}) - \mathbf{z}(\mathbf{x})\|_\infty \leq \frac{|\delta^+|}{4\|\mathbf{U}\|_1\|\mathbf{e}\|_1}$, $\forall \mathbf{x} \in \mathcal{D}$. Define the polynomial function:

$$\mathbf{Y}_p(\mathbf{x}) \doteq \mathbf{Y}_o(\mathbf{x}) + \mathbf{z}_p(\mathbf{x})\mathbf{U} - \frac{\delta^+}{4\|\mathbf{e}\|_1}\mathbf{1}^T \quad (25)$$

By construction $\mathbf{Y}_p(\mathbf{x})\mathbf{N} = \mathbf{d}(\mathbf{x})$[3] and satisfies:

$$(\mathbf{Y}_p)_i = (\mathbf{Y} + (\mathbf{z}_p - \mathbf{z})\mathbf{U})_i - \frac{\delta^+}{4\|\mathbf{e}\|_1}$$
$$\geq \mathbf{Y}_i - \frac{|\delta^+|\|\mathbf{U}\|_1}{4\|\mathbf{U}\|_1\mathbf{e}\|_1} - \frac{\delta^+}{4\|\mathbf{e}\|_1} \geq 0$$

where we used the fact that $\|\mathbf{z}\mathbf{U}\|_\infty \leq \|\mathbf{z}\|_\infty\|\mathbf{U}\|_1$. Further

$$\mathbf{Y}_p\mathbf{e} = \mathbf{Y}\mathbf{e} + (\mathbf{z}_p - \mathbf{z})\mathbf{U}\mathbf{e} - \frac{\delta^+}{4\|\mathbf{e}\|_1}\mathbf{1}^T\mathbf{e}$$
$$\leq -|\delta^+| + \frac{|\delta^+|\|\mathbf{U}\|_1\|\mathbf{e}\|_1}{4\|\mathbf{U}\|_1\mathbf{e}\|_1} - \frac{\delta^+}{4\|\mathbf{e}\|_1}\|\mathbf{e}\|_1 < 0$$

Thus, the polynomial vector $\mathbf{Y}_p$ satisfies (23). ■

Remark 2: Note that, if the trajectories of the system are confined to a compact domain $\mathcal{D}$, then from Stone Weierstrass it follows that the dynamics $f(\mathbf{x}), g(\mathbf{x})$, can be uniformly approximated, arbitrarily close, by polynomials. Thus, in this case we can always assume that $\mathbf{Y}_p(\mathbf{x})$ is indeed polynomial, by increasing the noise bound ϵ by an arbitrarily small number.

Algorithm 1 Reweighted $\|.\|_*$ Based DDC Design

Initialize: $iter = 0$, $\mathbf{W}^{(0)} = I$, $d_f, d_g, d_a, d_b, d_c, \lambda, l, h$
repeat
Solve

$$\begin{aligned}
&\min_{m_\alpha} \quad \mathrm{Trace}(\mathbf{W}^{(iter)}\mathbf{M})\\
&\text{subject to:}\\
&\mathbf{M}(m_\alpha) \succeq 0 && (A.1)\\
&\mathbf{M}(1,1) = 1 && (A.2)\\
&\mathbf{k}_l = \mathbf{k}_r && (A.3)\\
&-\mathbf{Y}\mathbf{e} \in \Sigma[\mathbf{x}] && (A.4)\\
&b \in \Sigma[\mathbf{x}] && (A.5)\\
&\mathbf{Y}_i \in \Sigma[\mathbf{x}] && (A.6)\\
&sum(\mathbf{c}_a) \geq h && (A.7)\\
&sum(\mathbf{c}_b) \geq l && (A.8)
\end{aligned}$$

Update

$$\mathbf{W}^{(iter+1)} = (\mathbf{M}^{(iter)} + \sigma_2(\mathbf{M}^{(iter)})I)^{-1}$$
$$iter = iter + 1$$

until rank $(\mathbf{M}) = 1$.

B. A Tractable Convex Relaxation for Polynomial Dynamics

As indicated above, in the case of polynomial dynamics, $\mathbf{Y}(\mathbf{x})$ can be taken to be polynomial. Thus, in this case Theorem 2 provides a necessary & sufficient condition for the existence of a rational density and associated control action that stabilizes the unknown plant, given in terms of existence of positive polynomials satisfying a set of linear (in)equalities. However, certifying positivity of a polynomial in more than 2 variables is NP hard[4] [16]. Thus, in this section, in order to obtain tractable problems we will relax $\mathbf{Y}(\mathbf{x})$ to a vector $\mathbf{Y}_{SoS}(\mathbf{x}) = [y_1(\mathbf{x}), \dots, y_i(\mathbf{x}), \dots, y_{2nT}(\mathbf{x})]$ where $y_i(\mathbf{x}) \doteq \mathbf{c}_{y_i}^T\mathbf{m}$ is SoS. In this scenario (23) reduces to:

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

where $\mathbf{k}_l, \mathbf{k}_r$ are the coefficients of the polynomials $\mathbf{Y}(\mathbf{x})\mathbf{N}$ and $\mathbf{d}(\mathbf{x})$. Since the entries of $\mathbf{k}_l$ are linear functions of $\mathbf{c}_{y_i}^T$ and the entries of $\mathbf{k}_r$ are bilinear in $\mathbf{c}_b, \mathbf{c}_c$ and $\mathbf{c}_b, \mathbf{c}_a$, the problem above is a QCQP. A convex relaxation can be obtained using the ideas outlined in Section II-E as follows. Define the vector $\mathbf{v} = [1, \mathbf{c}_a^T, \mathbf{c}_b^T, \mathbf{c}_c^T]$. The moment matrix corresponding to the first order relaxation of (26) is given by $\mathbf{M} = \mathbf{v}\mathbf{v}^T$, and all terms bilinear in $\mathbf{k}_r$ are replaced by the corresponding linear term in M. Enforcing the constraint rank $(\mathbf{M}) = 1$ introduced in Section II through the use of a re-weighted trace heuristics [20], leads to the following Algorithm 1.

Here d_f, d_g denote the order of the dynamics, while d_a, d_b, d_c indicate the desired order of the controller. λ should be chosen to satisfy the integrability condition. If upper bounds on d_f, d_g are not a-priori known, one can start

[3]Note that $\mathbf{1}^T\mathbf{N} = \mathbf{0}$.

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

from $d_f = d_g = 0$ and increase the values until a feasible solution is found. The constraints (A.1)-(A.2) are the standard requirements of the moments matrix, (A.3)-(A.4) correspond to (24), (A.5)-(A.6) correspond to $b(\mathbf{x}) > 0$ and $\mathbf{Y}_i(\mathbf{x}) \geq 0$. The additional constraints (A.7)–(A.8) are used to regularize the problem, avoiding the trivial solution $\mathbf{k}_r = \mathbf{k}_l = 0$.

Next, we briefly discuss the computational complexity and scaling properties of Algorithm 1. The semi-definite constraint (A.1) involves a matrix $\mathbf{M}$ of size $q \times q$ where $q \doteq \binom{n+d_a}{n} + \binom{n+d_b}{n} + \binom{n+d_c}{n} + 1$. The polynomial vector $-[\mathbf{d}_1 + \mathbf{d}_2] \in \mathbb{R}^{1 \times n(s_f+s_g)}$, where each entry has degree $p = \max(d_f + d_a, d_g + d_c) + d_b - 1$. Therefore the number of constraints in (A.3), given by the size of $\mathbf{k}_r$, is $n(s_f+s_g)\binom{n+p}{n}$. (A.4)-(A.6) involve $2nT+2$ SoS constraints. The Gram matrix associated with each $\mathbf{Y}_i$ has size $n_\mathbf{Y} \times n_\mathbf{Y}$, where $n_\mathbf{Y} \doteq \binom{n+\lceil \frac{p}{2} \rceil}{n}$. Thus (A.4) and (A.6) combined involve a total of $2nT+1$ semi-definite constraints, each on a $n_\mathbf{Y} \times n_\mathbf{Y}$ matrix, while (A.5) involves an additional one of size $\binom{n+\lceil \frac{d_b}{2} \rceil}{n} \times \binom{n+\lceil \frac{d_b}{2} \rceil}{n}$. In conclusion, the major computational burden comes from the condition (A.6) that establishes a trade-off between performance and computational complexity. A larger number of samples T reduces the size of the consistency set, therefore increasing the chances of finding a feasible controller (since the controller is required to stabilize a smaller set of possible plants), but at the price of an increased computational cost. A potential way around this trade-off, beyond the scope of the present paper, is to exploit decomposition based polynomial optimization methods [21].

C. Extension to Rational Dynamics

The results of the previous section easily extend to rational dynamics. When $\boldsymbol{\phi}(\mathbf{x})$ and $\boldsymbol{\gamma}(\mathbf{x})$ are rational, we can search for a density of the form $\rho(\mathbf{x}) = p(\mathbf{x})\rho_r(\mathbf{x})$ where $p(\mathbf{x})$ is the minimum common multiple of the denominator of the entries of $\boldsymbol{\phi}, \boldsymbol{\gamma}$. Using this $\rho(.)$ leads to a polynomial $\mathbf{d}(\mathbf{x})$. From this point on, Algorithm 1 can be directly applied.

IV. Simulation Results

In this section, we illustrate the effectiveness of the proposed approach with several examples. In all cases the simulations were run using MATLAB [22], the data was generated with the ode45 [23] function and the optimization problem was solved with YALMIP [24] and mosek [25]. In all the examples, Algorithm 1 converged in a single iteration.

Example 1 [13]: The nonlinear plant is given by

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

The system was excited for 5 seconds with a random signal uniformly distributed in $[-1, 1]$, starting from the initial condition $[1; -1]$. We collected T $= 500$ samples, corrupted by noise uniformly distributed in $[-0.05, 0.05]$. As in [13], we chose $\lambda = 4$. We used a quadratic function: $b(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x}$ and selected $d_f = 3, d_g = 0, d_a = 0, d_c = 3, l = 3, h = 1\text{e}{-4}$. Algorithm 1 led, in 112s, to a third-order

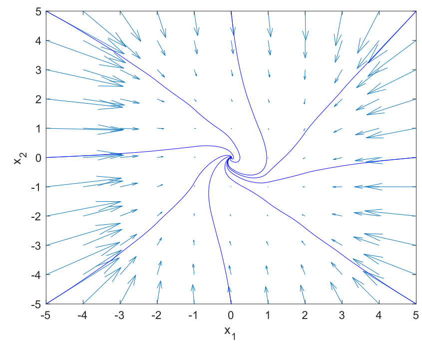

Fig. 1. Phase plot of the closed-loop system in Example 1 ($d_a = 0$).

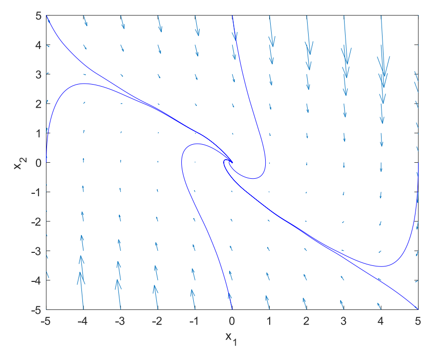

Fig. 2. Phase plot of the closed-loop system in Example 2 ($d_a = 0$).

controller:

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

The phase plot of the corresponding closed-loop system is shown in Fig. 1. For comparison purposes, we also searched for a rational controller with $d_a = 2$. In this case, in 334s, we obtained the following controller:

$$\begin{aligned} u(\mathbf{x}) = &(-0.0751x_1^3 - 0.607x_1^2x_2 - 0.0756x_1^2 \\ &- 0.137x_1x_2^2 + 0.0653x_1x_2 - 1.18x_1 - 0.476x_2^3 \\ &- 0.38x_2^2 - 0.923x_2)/(5.5\text{e}{-4}x_1^2 + 1.03\text{e}{-4}x_1x_2 \\ &+ 0.036x_1 + 5.97\text{e}{-4}x_2^2 + 0.0303x_2 + 0.937). \end{aligned}$$

Example 2: Next we consider the Van der Pol oscillator:

$$\begin{aligned} \dot{x}_1 &= x_2 + \eta_1 \\ \dot{x}_2 &= (1 - x_1^2)x_2 - x_1 + u + \eta_2 \end{aligned} \tag{28}$$

In this case, in 133s, Algorithm 1 led to the controller:

$$\begin{aligned} u(\mathbf{x}) = &-0.102x_1^3 + 0.821x_1^2x_2 - 0.283x_1^2 - 0.251x_1x_2^2 \\ &- 0.437x_1x_2 - 0.315x_1 - 0.24x_2^3 - 0.316x_2^2 - 2.61x_2 \end{aligned}$$

The corresponding phase plot of the closed-loop system is shown in Fig. 2. Next, setting $d_a = 2$, resulted, in 360s, in

$$\begin{aligned} u(\mathbf{x}) = &(-0.207x_1^3 + 0.338x_1^2x_2 - 0.307x_1^2 - 0.744x_1x_2^2 \\ &- 0.532x_1x_2 - 0.162x_1 - 0.463x_2^3 - 0.303x_2^2 \\ &- 2.27x_2)/(0.0067x_1^2 + 0.0208x_1x_2 + 0.044x_1 \\ &+ 0.0192x_2^2 + 0.0425x_2 + 0.869) \end{aligned}$$

Example 3: A grey-box rational case. Consider the system

$$\dot{x}_1 = \frac{1}{1+x_1^2}(x_1+u)+\eta_1$$
$$\dot{x}_2 = x_1+\eta_2 \tag{29}$$

and assume that it is known that the following basis

$$\left[1, x_1, x_2, \dots, x_2^m, \frac{1}{1+x_1^2}, \frac{x_1}{1+x_1^2}, \frac{x_2}{1+x_1^2}, \dots, \frac{x_2^m}{1+x_1^2}\right] \tag{30}$$

spans the dynamics. Searching for a density ρ of the form $\rho = (1+x_1^2)\hat{\rho}_r(\mathbf{x})$, where $\hat{\rho}_r(.)$ is rational, led, in 131s, to the following stabilizing controller:

$$u(\mathbf{x}) = -1.59x_1 - 0.429x_2. \tag{31}$$

V. Conclusion

In this letter, we proposed a framework for designing state-feedback data-driven controllers for nonlinear systems. Specifically, given an unknown nonlinear system parameterized in terms of a known set of basis functions and experimental measurements, we developed an algorithm guaranteed to stabilize (in the sense of [9]) all possible plants compatible with both the a-priori information and the experimental data. Thus, this algorithm can be considered as a robust generalization of the algorithm proposed in [13] to the case of unknown plants. The main theoretical result leading to this generalization established that, by using a combination of Rantzer's Dual Lyapunov approach and elements from convex analysis, the nonlinear data-driven control problem can be recast as an optimization over continuous positive functions (the Farkas multipliers). Further, for the case of polynomial dynamics, we showed that these multipliers can also be taken to be polynomial. Relaxing these polynomials to be sum-of-squares led to a QCQP that, in turn, can be relaxed to a convex SDP by exploiting the commonly used nuclear norm relaxation of rank. The effectiveness of the algorithm was illustrated with three examples, showing that indeed the proposed approach leads to stabilizing controllers directly from noisy data. Possible extensions of the framework presented here include exploiting the underlying sparse structure of the constraints to reduce the computational burden and extending the approach to solve reach-avoid type problems similar to those considered in [26].

Appendix
Proof of Continuity in Theorem 2

For a given $\mathbf{x}$ consider the following optimization problem:

$$J(\mathbf{x}) \doteq \min \hat{\mathbf{Y}}(\mathbf{x})\mathbf{e} \text{ subject to:}$$
$$\hat{\mathbf{Y}}(\mathbf{x})\mathbf{N} = \mathbf{d}(\mathbf{x}), \hat{\mathbf{Y}}(\mathbf{x})\mathbf{e} \geq -1, \text{ and } \hat{\mathbf{Y}}(\mathbf{x}) \geq 0 \tag{32}$$

(23) is feasible if and only if the problem above is feasible, and admits a solution set $\hat{\mathbf{Y}}(\mathbf{x})$ such that $\hat{\mathbf{Y}}(\mathbf{x})\mathbf{e} < 0$. Define the set valued mapping $\mathbf{Y}(\mathbf{x}) \doteq \{\mathbf{y} \in \hat{\mathbf{Y}}(\mathbf{x}) : \mathbf{ye} \leq J(\mathbf{x})\}$. Using [18, Definition 1.4.2] and [27, Th. 2.4] establishing continuity of the solutions of linear programs with respect to perturbations in the right hand side, it follows that $\mathbf{Y}(\mathbf{x})$ is lower semi-continuous. Consider now the minimum selection

$$\mathbf{Y}_m(\mathbf{x}) \doteq \underset{\mathbf{y}\in\mathbf{Y}(\mathbf{x})}{argmin} \|\mathbf{y}\|$$

For any bounded region $\mathcal{D}$, from continuity of $\mathbf{d}$, it follows that the range of $\mathbf{Y}_m(.)$ is bounded. Hence, from [18, Proposition 9.3.2], it follows that the function $\mathbf{Y}_m(\mathbf{x})$ is continuous. The proof is completed by noting that, by construction $\mathbf{Y}_m(\mathbf{x})$ solves the original problem (23).

References

[1] M. Campi and S. Savaresi, "Virtual reference feedback tuning for nonlinear systems," in *Proc. 44th IEEE Conf. Decis. Control (CDC)*, Seville, Spain, 2005, pp. 6608–6613.
[2] M. Campi and S. Savaresi, "Direct nonlinear control design: The virtual reference feedback tuning (VRFT) approach," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 14–27, Jan. 2006.
[3] H. Hjalmarsson, "Control of nonlinear systems using iterative feedback tuning," in *Proc. Amer. Control Conf. (ACC)*, vol. 4. Philadelphia, PA, USA, 1998, pp. 2083–2087.
[4] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2173–2188, Dec. 2011.
[5] G. Bontempi and M. Birattari, "From linearization to lazy learning: A survey of divide-and-conquer techniques for nonlinear control," *Int. J. Comput. Cogn.*, vol. 3, no. 1, pp. 56–73, 2005.
[6] E. Kaiser, J. Kutz, and S. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," 2017. [Online]. Available: arXiv:1707.01146.
[7] B. Lusch, J. Kutz, and S. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nat. Commun.*, vol. 9, no. 1, pp. 1–10, 2018.
[8] C. Novara, S. Formentin, S. Savaresi, and M. Milanese, "Data-driven design of two degree-of-freedom nonlinear controllers: The D^2-IBC approach," *Automatica*, vol. 72, pp. 19–27, Oct. 2016.
[9] A. Rantzer, "A dual to Lyapunov's stability theorem," *Syst. Control Lett.*, vol. 42, no. 3, pp. 161–168, 2001.
[10] T. Dai and M. Sznaier, "Data driven robust superstable control of switched systems," *IFACPapersOnLine*, vol. 51, no. 25, pp. 402–408, 2018.
[11] T. Dai and M. Sznaier, "A moments based approach to designing MIMO data driven controllers for switched systems," in *Proc. IEEE Conf. Decis. Control (CDC)*, Miami Beach, FL, USA, 2018, pp. 5652–5657.
[12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[13] S. Prajna, P. A. Parrilo, and A. Rantzer, "Nonlinear control synthesis by convex optimization," *IEEE Trans. Autom. Control*, vol. 49, no. 2, pp. 310–314, Feb. 2004.
[14] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
[15] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796–817, 2001.
[16] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," in *Emerging Applications of Algebraic Geometry*. New York, NY, USA: Springer, 2009, pp. 157–270.
[17] R. S. Sanchez-Pena and M. Sznaier, *Robust Systems Theory and Applications*. New York, NY, USA: Wiley, 1998.
[18] J. P. Aubin and H. Frankowska, *Set-Valued Analysis*. Boston, MA, USA: Springer, 2009.
[19] W. Cheney and W. Light, *A Course in Approximation Theory*. Providence, RI, USA: Amer. Math. Soc., 2009.
[20] K. Mohan and M. Fazel, "Iterative reweighted algorithms for matrix rank minimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 3441–3473, 2012.
[21] J. Miller, Y. Zheng, M. Sznaier, and A. Papachristodoulou, "Decomposed structured subsets for semidefinite and sum-of-squares optimization," 2019. [Online]. Available: arXiv:1911.12859.
[22] *MATLAB*, MathWorks Inc., Natick, MA, USA, 2018.
[23] L. Shampine and M. Reichelt, "The MATLAB ODE suite," *SIAM J. Sci. Comput.*, vol. 18, no. 1, pp. 1–22, 1997.
[24] J. Lofberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, USA, 2004, pp. 284–289.
[25] *The MOSEK Optimization Toolbox for MATLAB Manual. Version 9.0.*, MOSEK ApS, Copenhagen, Denmark, 2019.
[26] A. Rantzer and S. Prajna, "On analysis and synthesis of safe control laws," in *Proc. 42nd Allerton Conf. Commun. Control Comput.*, 2004, pp. 1468–1476.
[27] O. L. Mangasarian and T.-H. Shiau, "Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems," *SIAM J. Control Optim.*, vol. 25, no. 3, pp. 583–595, May 1987.