Les Cahiers du GERAD ISSN: 0711–2440

Learning stable Koopman embeddings for identification and control

F. Fan, B. Yi, D. Rye, G. Shi, I. R. Manchester

G–2024–63

Septembre 2024

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Learning stable Koopman embeddings for identification and control

Fletcher Fan [a]

Bowen Yi [b, c]

David Rye [a]

Guodong Shi [a]

Ian R. Manchester [a]

[a] *Australian Centre for Robotics and School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, NSW 2006, Australia*

[b] *Department of Electrical Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3T 1J4*

[c] *GERAD, Montréal (Qc), Canada, H3T 1J4*

bowen.yi@polymtl.ca

Septembre 2024
Les Cahiers du GERAD
G–2024–63

Abstract : This paper introduces new model parameterizations for learning dynamical systems from data via the Koopman operator, and studies their properties. Whereas most existing works on Koopman learning do not take into account the stability or stabilizability of the model – two fundamental pieces of prior knowledge about a given system to be identified – in this paper, we propose new classes of Koopman models that have built-in guarantees of these properties. These models are guaranteed to be stable or stabilizable via a novel *direct parameterization approach* that leads to *unconstrained* optimization problems with respect to their parameter sets. To explore the representational flexibility of these model sets, we establish novel theoretical connections between the stability of discrete-time Koopman embedding and contraction-based forms of nonlinear stability and stabilizability. The proposed approach is illustrated in applications to stable nonlinear system identification and imitation learning via stabilizable models. Simulation results empirically show that the learning approaches based on the proposed models outperform prior methods lacking stability guarantees.

Keywords : Nonlinear systems, Koopman operator, contraction analysis, system identification, data-driven control

Acknowledgements: This work was supported in part by the Australian Research Council (ARC), the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Programme PIED of Polytechnique Montréal.

1 Introduction

Many fundamental phenomena in engineering and science can be described by dynamical systems, making the modeling of dynamical systems a ubiquitous problem across various domains. These models can not only be used to predict future behavior but have also proven effective in planning, estimation, and designing a controller to interact with the real physical world. In general, deriving a model of a dynamical system from first principles may be challenging or even intractable for cases involving complex tasks, such as imitating human behavior. This is where system identification approaches that learn a model from data become useful.

A central consideration for learning algorithms is the model structure. For identifying memoryless input-output mappings, deep neural networks have achieved state-of-the-art results in many domains, such as image classification [26] and playing strategy games [42]. In contrast, learning *dynamical* models introduces additional challenges due to the presence of internal memory and feedback. In particular, ensuring the behavioural properties of dynamical models during learning, including stability and stabilizability, is an important aspect that is non-trivial even for linear systems. For example, even if a physical system is known to be stable, a model learned from data might exhibit instability due to the unavoidable effects of measurement noise, under-modeling, and the challenges of optimization.

To address this, some recent works aim to impose constraints in terms of prior physical knowledge, specifically using stability constraints as a control-theoretic regularizer for model learning. As summarized in [13], there are two main categories for learning dynamical models with stability guarantees: 1) constrained optimization, and 2) learning potential functions via diffeomorphism. Learning algorithms for stable systems have been comprehensively studied for linear systems (see, for example [15, 18, 27, 50]); in constrast, the nonlinear couterpart is more techinically challenging, necessitating universal fitting tools to address nonlinearity. Most approaches for nonlinear stable systems employ the Lyapunov method [22] and contraction analysis [45, 46, 49], which yield constrained optimization that limits scalability to large models. More recently, research has focused on *directly parameterizing* stable nonlinear systems to achieve unconstrained optimization problems [6, 33, 35, 39]. These are more general than the second category outlined in [13].

In recent years, there has been a growing interest in the Koopman operator for the analysis, control, and learning of nonlinear systems [36, 44]. It is a composition operator that characterizes the evolution of scalar observables from a spectral decomposition perspective [24]. Despite its infinite-dimensional nature, the Koopman operator itself exhibits linearity and proves powerful in addressing various data-driven analysis and prediction problems [17, 25, 44]. Through Koopman theory, nonlinear systems can be studied via a spectral decomposition of the Koopman operator, akin to linear systems analysis. This has huge potential in applying tools from linear systems theory to nonlinear systems, including global stability analysis [36, 52] and a number of linear control methodologies.

In this paper, we focus on Koopman models – a recently emerging class of models that are both flexible and interpretable – and Koopman learning frameworks. When learning a Koopman model from data, one attempts to find a finite-dimensional (usually approximate) representation of the Koopman operator, which amounts to a linear matrix along with a mapping that transforms the original state space of the system to a so-called Koopman-invariant subspace. As mentioned above, it can be important to consider model stability and stabilizability during learning, which has not been fully addressed in earlier work on Koopman learning. The paper aims to address the aforementioned challenges with the main contributions below:

1. We provide a novel parameterization to the stable Koopman model set, which is *unconstrained* in its parameters, allowing for efficient and "plug-and-play" optimization by leveraging software tools for automatic differentiation.
2. For nonlinear discrete-time systems, we prove the equivalence between the Koopman and contraction criteria for stability analysis, extending our earlier work [52] to the discrete-time context.

Such an equivalence is practically useful in proposing a novel Koopman learning framework that is capable of learning most stable autonomous systems under some mild technical assumptions.

3. The proposed Koopman model set is extended to the generalized feedback linearizable systems, for which we also develop unconstrained optimization but simultaneously impose the stabilizability constraint to the model set. These results are applied to a problem of imitation learning (i.e. learning a control policy from demonstrations) incorporating regularization to guaranteejg closed-loop stability.

Compared to the preliminary conference version [12], this paper provides the full proof of Theorem 1. In addition, we extend the main idea to nonlinear control, which forms the basis for introducing the stabilizable Koopman model and outlining the imitation learning framework in Section 5.

Notation. All mappings and functions are assumed sufficiently smooth. $\lambda_{\texttt{min}}(\cdot)$ and $\lambda_{\texttt{max}}(\cdot)$ respectively represent the smallest and largest eigenvalues of a symmetric matrix. Given a matrix $A \in \mathbb{R}^{n\times m}$ $(n > m)$, we use $A^\perp \in \mathbb{R}^{(n-m)\times n}$ to represent a full-rank left annihilator such that $A^\perp A = 0$. We use $|\cdot|$ to denote the standard Euclidean norm, i.e. $|x| = \sqrt{x^\top x}$. When clear from context, we may simply write $x(t)$ as x_t, omitting the arguments of mappings and functions, and use $\tilde{x}_t$ to represent the measured data corresponding to the true state x_t at time t.

2 Preliminaries

This section presents some preliminaries on the Koopman operator and contraction analysis. Consider the discrete-time autonomous system in the form

$$x_{t+1} = f(x_t) \tag{1}$$

with the state $x \in \mathbb{R}^n$, and a smooth vector field $f : \mathbb{R}^n \to \mathbb{R}^n$.

The Koopman operator, originally proposed in [24], provides a simple and effective way to analyze nonlinear systems. Its discrete-time version is defined as follows.

Definition 1. Let $\mathcal{F}$ be the observable space of scalar functions $\mathbb{R}^n \to \mathbb{C}$. For the system (1), the Koopman operator $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ is defined by

$$\mathcal{K}[\varphi] := \varphi \circ f \tag{2}$$

for $\varphi \in \mathcal{F}$, where $\circ$ represents function composition.

Since the Koopman operator is defined on the functional space, it is infinite-dimensional. It is easy to verify the linearity of the Koopman operator, which leads to many useful methods for the analysis, control, and learning of nonlinear systems. Despite its infinite dimensionality, one can sometimes obtain a computationally tractable expression of the Koopman operator by finding a finite set of observables that span an invariant subspace. This allows us to represent the action of the operator with a matrix using the chosen bases on the invariant subspace.

Definition 2. A Koopman-invariant subspace is defined as $\mathcal{G} \subset \mathcal{F}$ such that $\mathcal{K}[\varphi] \in \mathcal{G},\ \forall \varphi \in \mathcal{G}$.

If a Koopman-invariant subspace $\mathcal{G}$ is spanned by a finite set of observables $\{\varphi_k\}_{k=1}^N$ with $N \in \mathbb{N}_+$, any function $f \in \mathcal{G}$ can be represented as $f(x) = \sum_{j=1}^N k_j \varphi_1$ with some scalars k_j. In some works, this is referred to as the mapping $\phi = \mathrm{col}(\varphi_1, \ldots, \varphi_N)$ as a Koopman embedding of the system (1).[1] Furthermore, if ϕ is a surjection, then the original nonlinear system (1) is topologically semiconjugate to a linear system via the coordinate transformation $x \mapsto z = \phi(x)$. More specifically, if ϕ is a homeomorphism, it becomes topologically conjugate [9, Ch. 10]. In the topological sense, an injective

[1] This is a restrictive assumption for general nonlinear systems, and thus in the literature the Koopman embedding is often approximated on a non-invariant set and can be learned using deep models [44].

continuous map ϕ is an embedding if ϕ yields a homeomorphism between $\mathbb{R}^n$ and $\phi(\mathbb{R}^n) \subset \mathbb{R}^N$. In this paper, we employ the term of "Koopman embedding" in the topological sense rather than in reference to a linear combination of observables. Hence, we do not need to study the invariance of the space spanned by observables; instead, we will use the left inverse of the Koopman embedding ϕ to pull the lift dynamics back to the original coordinate.

Contraction, also known as incremental exponential stability (IES), is a strong form of stability: if a given system is contracting, any two trajectories will ultimately converge to each other [30]. Contraction analysis provides another "exact and global linearization" way to study nonlinear stability by analyzing the stability of the linear time-varying (LTV) differential system

$$\delta x_{t+1} = \frac{\partial f}{\partial x}(x_t)\delta x_t \tag{3}$$

along all feasible trajectories. The new variable δx, living on the tangent space of the original system (1), represents the infinitesimal displacement among trajectories. It has shown success in a series of constructive problems for nonlinear systems, including controller synthesis [32], observer design [53], and learning algorithms [39, 43].

We briefly recall the discrete-time definition of contraction as follows.

Definition 3. Given the nonlinear system (1), if there exists a uniformly bounded metric $M(x)$, i.e. $a_1 I_n \preceq M(x) \preceq a_2 I_n$ for some $a_2 \geq a_1 > 0$, guaranteeing

$$\frac{\partial f}{\partial x}(x_t)^\top M(x_{t+1})\frac{\partial f}{\partial x}(x_t) - M(x_t) \preceq -\beta M(x_t), \tag{4}$$

with $0 < \beta < 1$, then the given system is contracting.

Intuitively, the inequality (4) can guarantee the quadratic Lyapunov-like function $\delta x^\top M(x)\delta x$ on tangent bundles decreases over time, uniformly across all feasible trajectories of x_t. A central result of contraction analysis is that, for contracting systems, all trajectories converge exponentially to a single trajectory. That is, for any trajectories x^a and x^b and for some $a_0 > 0$

$$|x_t^a - x_t^b| \leq a_0 \beta^t |x_0^a - x_0^b|. \tag{5}$$

Considering the similarity between the contraction and Koopman approaches, our recent paper [52] shows the equivalence between them for nonlinear stability analysis but focuses on *continuous-time* systems.

3 Motivations and problem set

In this paper, we are concerned with the discrete-time nonlinear autonomous system (1) and the control system

$$x_{t+1} = f(x_t) + g(x_t)u_t, \tag{6}$$

but the dynamics is assumed *unknown*, where the state $x \in \mathbb{R}^n$, the input $u \in \mathbb{R}^m$, and the vector fields $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$. When there is no external input, i.e. $u \equiv 0$, the control model (6) degrades into the autonomous system as introduced in (1).

Suppose $N_{\texttt{traj}}$ data samples $\mathcal{E}_{\texttt{D}} := \{\tilde{x}_t, \tilde{u}_t\}_{t=1}^{N_{\texttt{traj}}}$ are used for model identification and learning a stabilizing controller, in which $\tilde{x}, \tilde{u}$ represents the measured noisy data of x, u generated by the system (6) over time. The fundamental question in system identification is to use the dataset $\mathcal{E}_{\texttt{D}}$ to approximate the vector fields f, g, denoted as $\hat{f}, \hat{g}$, in some optimal sense. Sometimes, it is necessary to impose additional *constraints* based on prior physical knowledge, such as stability, stabilizability,

and controllability [34, 45]. We may compactly write as $(\hat{f}, \hat{g}) \in \mathcal{E}_{\mathtt{M}}$ with the set $\mathcal{E}_{\mathtt{M}}$ characterizing these constraints. The system identification problem is generally based on minimizing a cost function

$$\min_{(\hat{f},\hat{g})\in\mathcal{E}_{\mathtt{M}}} \; J(\mathcal{E}_{\mathtt{D}}, \hat{f}, \hat{g}). \tag{7}$$

Given a data set, the main considerations of nonlinear system identification are the parameterization of nonlinear functions $\hat{f}, \hat{g}$, the selection of the cost function J, and specific optimization algorithms.

In this paper, we propose two novel model parameterizations: the stable Koopman model and the stabilizable Koopman model. The main theoretical problem we are interested in is how to parameterize these model sets that are unconstrained in parameters. This endeavor is motivated by and finds practical applications in the following.

Motivating Applications: Given the dataset $\mathcal{E}_{\mathtt{D}}$ and a cost function $J(\mathcal{E}_{\mathtt{D}}, \hat{f}, \hat{g})$, solve the following two problems.

P1: (Learning stable autonomous systems) Consider a contracting system for the case $u \equiv 0$, learn an approximate model $\hat{f}$ from the dataset $\mathcal{E}_{\mathtt{D}}$ generated by the system, and guarantee that the identified dynamics $x_{t+1} = \hat{f}(x_t)$ is contracting.

P2: (Imitation learning) Considering the dataset $\mathcal{E}_{\mathtt{D}}$ generated from an asymptotically stabilizable system (6) via smooth static feedback, learn a static feedback $u = \rho(x)$ approximating the demonstration data and concurrently guaranteeing that the closed loop $x_{t+1} = f(x_t)+g(x_t)\rho(x_t)$ is contracting.

We will address the above motivating applications in Sections 4 and 5, respectively. Note that determining the functions f and g requires solving an infinite-dimensional optimization problem. To make this tractable, we parameterize the functions, i.e. $\hat{f}(x, \theta)$, $\hat{g}(x, \theta)$ using some basis functions that may be selected as polynomials, neural networks or many others. The theoretical question therein is how to introduce parameterizations to guarantee stability and stabilizability properties for the proposed model sets.

4 Learning stable Koopman embeddings

In this section, we focus on the autonomous case, introducing a novel stable model class that covers all contracting systems under some technical assumptions and studying its equivalent parameterization. Based on them, we propose an algorithm to learn stable Koopman embeddings.

4.1 Stable Koopman model class: Covering all contracting systems

Let us consider a Koopman model class for discrete-time autonomous systems in the form of (1). We define a Koopman model for this system as follows. [2]

M1. Stable Koopman Model $(A, \phi, \phi^{\mathtt{L}})$:

$$\begin{aligned} z_0 &= \phi(x_0) \\ z_{t+1} &= Az_t \\ x_t &= \phi^{\mathtt{L}}(z_t), \end{aligned} \tag{8}$$

in which 1) $z \in \mathbb{R}^N$ $(N \geq n)$ is a lifted internal variable; 2) A is Schur stable; and 3) ϕ has a left inverse satisfying $\phi^{\mathtt{L}}(\phi(x)) = x$, $\forall x$.

[2] The left invertibility of ϕ implies $x_t = \phi^{\mathtt{L}}(z_t)$.

4.1.1 Stability criterion for Koopman models

The following theorem is a discrete-time version of the main results in [52], showing the equivalence between the Koopman and contraction approaches. As a consequence, it illustrates the Koopman model class **M1** covers all the contracting discrete-time autonomous systems under some technical assumptions of f below.

Assumption 1. The vector field f is invertible and its inverse f^{-1} is continuous satisfying $|x| \leq c_1 + c_2|f(x)|$ for some $c_1, c_2 \in \mathbb{R}_+$ and $\|\frac{\partial f}{\partial x}(x)\| < \min\{1, c_2^{-1}\}$.

Thus, the model class is capable of providing sufficient degrees of freedom for learning discrete-time nonlinear systems. We have the following.

Theorem 1. For the system (1), suppose that there exists a C^1-continuous mapping $\phi : \mathbb{R}^n \to \mathbb{R}^N$ with $N \geq n$ such that

D1: There exists a Schur stable matrix $A \in \mathbb{R}^{N\times N}$ satisfying the algebraic equation

$$\phi \circ f - A\phi = 0, \quad \forall x \in \mathbb{R}^n. \tag{9}$$

D2: $\frac{\partial \phi}{\partial x}$ has full column rank, and $(\frac{\partial \phi}{\partial x})^\top \frac{\partial \phi}{\partial x}$ is uniformly bounded.

Then, the system is contracting with the contraction metric $M(x) := \frac{\partial \phi}{\partial x}(x)^\top P \frac{\partial \phi}{\partial x}(x)$, where P is any positive-definite matrix satisfying $P - A^\top P A \succ 0$. Conversely, if the system (1) is contracting and satisfies Assumption 1, then in any forward invariant[3] compact set $\mathcal{X} \subset \mathbb{R}^n$ for the dynamics (1), there exists a C^1-continuous mapping $\phi : \mathbb{R}^n \to \mathbb{R}^N$ verifying D1-D2.

Proof. ($\Longrightarrow$) In this part of the proof, we need to verify the contraction condition (3) from the Koopman conditions D1 and D2.

From D1 (Schur stability of A), there exists a matrix $P = P^\top \succ 0$ satisfying the Lyapunov condition

$$P - A^\top P A \succ I \succeq \rho I, \tag{10}$$

for some scalar $\rho \in (0, 1]$. Considering the C^1-continuity of ϕ and f, we calculate the partial derivative of (9), obtaining

$$\frac{\partial \phi}{\partial x}(f(x))\frac{\partial f}{\partial x}(x) = A\frac{\partial \phi}{\partial x}(x) \tag{11}$$

Invoking $x_{t+1} = f(x_k)$, the above can be rewritten as

$$\frac{\partial \phi}{\partial x}(x_{t+1})\frac{\partial f}{\partial x}(x_t) = A\frac{\partial \phi}{\partial x}(x_t). \tag{12}$$

Due to the full rank of $\frac{\partial \phi}{\partial x}$ and (10), it follows that

$$\frac{\partial \phi}{\partial x}(x)^\top (P - A^\top P A)\frac{\partial \phi}{\partial x}(x) \succ \frac{\partial \phi}{\partial x}(x)^\top \rho I \frac{\partial \phi}{\partial x}(x). \tag{13}$$

Then, by substituting (12), we have

$$\begin{aligned} &\frac{\partial \phi}{\partial x}(x_t)^\top P \frac{\partial \phi}{\partial x}(x_t) - \frac{\partial f}{\partial x}(x_t)^\top \frac{\partial \phi}{\partial x}(x_{t+1})^\top P \frac{\partial \phi}{\partial x}(x_{t+1})\frac{\partial f}{\partial x}(x_t) \\ \succ\ &\rho \frac{\partial \phi}{\partial x}(x_t)^\top \frac{\partial \phi}{\partial x}(x_t) \\ \succeq\ &\beta \frac{\partial \phi}{\partial x}(x_t)^\top P \frac{\partial \phi}{\partial x}(x_t), \end{aligned} \tag{14}$$

[3] For a complete system (1) with the solution $X(x, k)$, a set $\mathcal{X}$ is said to be a forward invariant set if whenever $x_0 \in \mathcal{X}$ and $j \in \mathbb{Z}_+$, we have $X(x_0, j) \in \mathcal{X}$ [40, pp. 198].

with $\beta := \frac{\rho}{\lambda_{\max}(P)}$. We choose $M(x) := \frac{\partial \phi}{\partial x}(x)^\top P \frac{\partial \phi}{\partial x}(x)$, which is uniformly bounded due to D2 and $P \succ 0$. Substituting into (14) leads to

$$M(x_t) - \frac{\partial f}{\partial x}(x_t)^\top M(x_{t+1}) \frac{\partial f}{\partial x}(x_t) \succ \beta M(x_t). \tag{15}$$

By selecting $\rho \in (0, 1]$ sufficiently small, we can guarantee that $\beta \in (0, 1)$. This is exactly the contraction condition for the system (1) with the contraction metric M.

($\Longleftarrow$) The second part of the proof is to show that a contracting system satisfies the Koopman conditions D1 and D2 in any invariant compact set $\mathcal{X} \subset \mathbb{R}^n$. The key step for D1 is to construct a feasible solution to the algebraic equation (9), and this part is motivited by the technical results in [5].

For the given discrete-time system, we directly apply the Banach fixed-point theorem, concluding the existence of a unique fixed-point $x_\star \in \mathcal{X}$, i.e. $f(x_\star) = x_\star$.[4] To construct a Koopman embedding ϕ, we parameterize it as $\phi(x) := x + T(x)$, for the particular case $N = n$, with a new mapping $T : \mathbb{R}^n \to \mathbb{R}^n$ to be searched for. Then, the equation (9) becomes

$$T(f(x)) + f(x) = Ax + AT(x). \tag{16}$$

Let us fix $A = \frac{\partial f}{\partial x}(x_\star)$. From the contraction assumption, we have $M_\star - A^\top M_\star A \succeq \beta M(x^\star)$ with $M_\star := M(x_\star) \succ 0$, and thus A is Schur stable. By defining $H(x) := Ax - f(x)$, the algebraic equation (16) becomes

$$T(f(x)) = AT(x) + H(x). \tag{17}$$

We make the key observation that (17) exactly coincides with the algebraic equation in the formulation of the Kazantzis-Kravaris-Luenberger (KKL) observer for nonlinear discrete-time systems in [5, Eq. (7)]. In our case, the function H is continuous and, following [5, Thm. 2], we have a feasible solution to (17):

$$T(x) = \sum_{j=0}^{+\infty} A^j H(X(x, -j+1)), \tag{18}$$

which is well-defined due to $\|\frac{\partial f}{\partial x}(x_\star)\| < \min\{1, c_2^{-1}\}$, with the definition for $j \in \mathbb{N}_+$

$$X(x, j) = \underbrace{f \circ f \circ \cdots \circ f}_{j \text{ times}}(x), \quad X(x, -j) = (f^\dagger)^j(x).$$

Note that the calculation of $X(x, -j)$ requires backward completness and invariance in $\mathcal{X}$. Since contracting systems generally cannot guarantee such invariance, we modify the backwards map f^{-1} as $f^\dagger(x) = \mu(x) f^{-1}(x) + [1 - \mu(x)]x$ with

$$\mu(x) = \begin{cases} 1, & \text{if } \ x \in \mathcal{X} + k_1 \\ \mu_f(x), & \text{if } \ x \in \mathcal{X} + k_2 \backslash \mathcal{X} + k_1 \\ 0, & \text{if } \ x \notin \mathcal{X} + k_2 \end{cases}$$

for some $k_2 > k_1 > 0$, where $\mathcal{X} + k_1$ represents the set of points that lie whthin in the distance $k_1 > 0$, and $\mu_f(x)$ is any locally Lipschitz function such that μ is C^1-continuous [47].

Now, we consider a candidate Koopman embedding $\phi^0(x) := x + T(x)$ with T defined above satisfies D1 in the entire set $\mathcal{X}$. However, the condition D2 does not necessarily hold, and we need to modify ϕ^0. By considering the evolution of the trajectories in the x-coordinate and a lifted coordinate defined as $z := \phi(x)$, respectively, we have

$$z(t_x) = \phi^0(x(t_x)) = \phi^0(X(x, t_x)) = A^{t_x} \phi^0(x),$$

[4] The incremental exponential stability condition (4) implies that for any $x_a, x_b \in \mathcal{X}$ we have $d_M(f(x_a), f(x_b))^2 \leq \beta d_M(x_a, x_b)^2$, with $\beta \in (0, 1)$ and $d_M(\cdot, \cdot)$ the distance associated with the metric $M(x)$. Therefore, the mapping f is a contraction mapping in a Banach space, and we may then apply [21, Appendix B].

with $t_x \in \mathbb{N}_+$, thus satisfying $\phi^0(x) = A^{-t_x}\phi^0(X(x,t_x))$. Then, we modify the candidate embedding ϕ^0 into

$$\phi(x) := A^{-t_x}[X(x,t_x) + T(X(x,t_x))] \tag{19}$$

with a sufficiently large $t_x \in \mathbb{N}_+$.

Finally, let us check conditions D1 and D2. For the algebraic condition, we have

$$\begin{aligned}\phi \circ f(x) &= A^{-t_x}\phi^0 \circ X(f(x), t_x)\\ &= A^{-t_x}\phi^0 \circ f \circ X(x,t_x)\\ &= A^{-t_x} \cdot A\phi^0 \circ X(x,t_x)\\ &= A\phi(x)\end{aligned}$$

where in the second equation we have used the fact

$$X(f(x),t_x) = \underbrace{f \circ f \circ \cdots \circ f}_{(t_x+1)\text{ times}} = f(X(x,t_x)).$$

Therefore, ϕ defined in (19) satisfies D1. Regarding D2, the Jacobian of ϕ is given by

$$\frac{\partial \phi}{\partial x}(x) = A^{-t_x}\left[I + \frac{\partial T}{\partial x}(X(x,t_x))\right]\frac{\partial X}{\partial x}(x,t_x).$$

On the other hand, we have that $\frac{\partial X}{\partial x}$ is full rank and

$$H(x^\star) = 0, \quad \frac{\partial H}{\partial x}(x_\star) = 0,$$

as a result $\frac{\partial T}{\partial x}(x_\star) = 0$. If $t_x \in \mathbb{N}_+$ is sufficiently large, the largest singular value of $\frac{\partial T}{\partial x}(X(x,t_x))$ would be very small, and then the identity part of ϕ will dominate $\frac{\partial \phi}{\partial x}$. Hence, ϕ satisfies the condition D2 that $\frac{\partial \phi}{\partial x}$ has full column rank, and the uniform boundedness can be directly obtained in a compact set. We complete the proof. □

Remark 1. The above shows the equivalence between Koopman and contraction approaches (i.e. the theoretical conditions) for stability analysis of discrete-time systems. This resembles the results for continuous-time systems in [52], which also includes the results for *time-varying* systems. In contrast, we limit ourselves to discrete-time time-invariant systems in this paper.[5] Intuitively, the proposed equivalence implies the generality and flexibility of the proposed model class **M1**, i.e., it covers all contracting nonlinear systems under some technical assumptions. While the construction of the Koopman mapping ϕ is an existence result, this shows the potential to use linear system identification techniques to learn a nonlinear model; see Fig. 1 for its basic idea. We will pursue it in the next subsection.

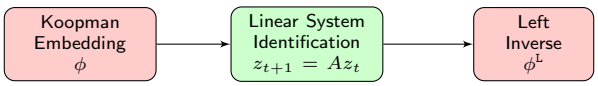


Figure 1: The proposed model class M1: Use linear system identification approaches to learn nonlinear models.

Remark 2. The above theorem shows that in theory, lifting with excessive coordinates ($N > n$) is unnecessary to obtain a linear system for a particular class of nonlinear systems, i.e. contracting systems. Similar results are also obtained in [28] for Schur stable systems. However, overparameterizing with $N \geq n$ may still be useful for black-box learning as we show empirically via simulations. The condition D2 makes ϕ locally injective, thus being an embedding that allows us to lift the given nonlinear system to a linear stable dynamics. Therefore, we use the terminology "Koopman embedding" for the proposed approach in the paper.

[5] Note that for autonomous (i.e. time-invariant unforced) systems, contraction is equivalent to exponential stability within any compact invariant subset of the domain of attraction. The paper [36] proposes a similar equivalence between global asymptotic stability and the Koopman stability criteria in the context of continuous-time *autonomous* systems.

4.2 Parameterization of stable Koopman models

In the proposed Koopman model **M1**, we need to identify three components: the stable matrix A, the mapping ϕ, and its left inverse ϕ^{L}. It is necessary to parameterize them to make the approach computationally tractable.

A key feature of the matrix A is Schur stability, for which there are several equivalent conditions, including the well-known Lyapunov inequality $P - A^\top P A \succ 0$ for some $P \succ 0$, and the recent parameterization in [15]. However, these constraints are non-convex thus yielding heavy computational burden. To address this, we introduce an *unconstrained* parameterization of stable A, which is a special case of the direct parameterization in [39].

Proposition 1. Consider an $N \times N$ matrix A parameterized as $A(L,R)$[6]

$$A(L,R) = 2(M_{11} + M_{22} + R - R^\top)^{-1} M_{21}, \tag{20}$$

where M_{ij} $(i,j = 1,2)$ are blocks in

$$M := \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = LL^\top + \epsilon I, \tag{21}$$

with ϵ a positive scalar, $L \in \mathbb{R}^{2N \times 2N}$, and $R \in \mathbb{R}^{N \times N}$. Then, the matrix $A(L,R)$ is Schur stable. Conversely, for any Schur stable matrix A, we can always find L, R and ϵ to parameterize it in the form of (20).

Proof. (Sufficiency) Let $E = \frac{1}{2}(M_{11} + M_{22} + R - R^\top)$, $F = M_{21}$ and $P = M_{22}$. Then, we have $A(L,R) = E^{-1}F$ and

$$M = \left[\begin{array}{c|c} E + E^\top - P & F^\top \\ \hline F & P \end{array}\right]. \tag{22}$$

It is shown in [46, Lemma 1, pp. 7235] that

$$M \succ \gamma I,\ \gamma > 0 \iff \text{Schur stability of } E^{-1}F. \tag{23}$$

Hence if there exist matrices L and R such that (20) and (21) hold, then $M \succeq \epsilon I$. Thus, $A(L,R)$ is Schur stable.

(Necessity) To prove necessity, invoking the equivalence in (23), it needs to be shown that a positive definite matrix $M \succ \gamma I$ can always be parameterized by $M = LL^\top + \epsilon I$ for some $L \in \mathbb{R}^{2N \times 2N}$ and $\epsilon \in \mathbb{R}_{>0}$. By the continuity of eigenvalues of a matrix with respect to its elements [4, Ch. 7], one has that $M - \epsilon I$ is positive definite by choosing a sufficiently small positive $\epsilon \ll \gamma$. Therefore, the Cholesky factorization guarantees the existence of L such that $M - \epsilon I = LL^\top$, as required. □

The observables are nonlinear functions, rendering them infinite dimensional. In order to be able to provide sufficient degrees, the Koopman embeddings are proposed to be parameterized as

$$\phi(x) = \begin{bmatrix} x \\ \varphi(x, \theta_{\text{NN}}) \end{bmatrix} \tag{24}$$

where the nonlinear part φ can be any differentiable function approximator, parameterized by θ_{NN}. For brevity, the dependence on θ_{NN} is dropped in the notation. In this paper, we choose φ to be a feedforward neural network due to its scalability. The dimensionality of the observables N is a hyperparameter chosen by the user. This specific structure guarantees the *existence* of left inverses, though not uniqueness. We use a separate feedforward neural network $\phi^{\text{L}} = \phi^{\text{L}}(\cdot, \theta_{\text{L}})$ with all unknown parameters collected in the vector θ_{L} to approximate the inverse, which is shown to have better performance empricially.

[6] To simplify the presentation, we use A to represent both a matrix and the parameterization function with a slight abuse of notation.

Remark 3. Eq. (24) is one of the feasible mappings described in Theorem 1. In fact, the construction of (24) can be further generalized as $\phi = \mathrm{col}(\psi, \varphi)$, where $\psi : \mathbb{R}^n \to \mathbb{R}^n$ is an invertible network, and $\varphi : \mathbb{R}^n \to \mathbb{R}^{N-m}$ is an arbitrary networks. Invertible neural networks have been extensively studied, with notable examples including bi-Lipschitz networks [51], invertible residual layers [7], and monotone networks [1]. In this work, we pick ψ to be just the identity map, for simplicity.

4.3 Learning framework for Koopman embeddings

Under the parameterization of the proposed model class, we need to use the dataset $\mathcal{E}_{\mathtt{D}}$ from the real-world system to fit the parameters $\theta := (\theta_{\mathtt{NN}}, \theta_{\mathtt{L}}, L, R)$. To this end, we need to solve optimization problems with a proper cost function.

Here, we consider minimizing the *simulation error* in the lifted coordinate $z = \phi(x)$:

$$J_{\mathtt{SE}} := \frac{1}{T} \sum_{t=1}^{T} |\tilde{z}_t - z_t|^2, \tag{25}$$

with $T = N_{\mathtt{traj}}$, $\tilde{z}_t = \phi(\tilde{x}_t)$, and $z_t = A(L,R)^{t-1}\phi(\tilde{x}_1)$. Here, $\tilde{x}_t$ denotes the measured data corresponding to the true state x_t at time t. In order to identify the left inverse $\phi^{\mathtt{L}}$ concurrently, we minimize the following composite cost function

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \; J_{\mathtt{SE}} + \alpha J_{\mathtt{RE}} \tag{26}$$

with the parameter space Θ, a weighting coefficient $\alpha > 0$, and the reconstruction loss $J_{\mathtt{RE}} := \frac{1}{T}\sum_{t=1}^{T} |\tilde{x}_t - \phi^{\mathtt{L}}(\phi(\tilde{x}_t, \theta_{\mathtt{NN}}), \theta_{\mathtt{L}})|^2$. The purpose of the term $J_{\mathtt{RE}}$ is to learn an approximate left-inverse $\phi^{\mathtt{L}}$ for ϕ. The loss $J_{\mathtt{RE}}$ can be thought of as a penalty term that relaxes the constraint $x = \phi^{\mathtt{L}}(\phi(x))\ \forall x$, and the scalar α determines the weighting of the penalty. We can make full use of data from multiple trajectories by constructing the cost function as the sum of $(J^j_{\mathtt{SE}} + \alpha J^j_{\mathtt{RE}})$, where, with a slight abuse of notations, the index j denotes the data sets from different feasible trajectories.

Remark 4. The model class is agnostic to the particular cost function that is minimized. Unconstrained parameterization in the proposed framework has the benefit that $J_{\mathtt{SE}} + \alpha J_{\mathtt{RE}}$ may be replaced by other feasible differentiable objective functions. An alternative is the simulation error in the original x-coordinate. However, in practice, this was found to produce poor performance. The simulation error in z can still be large while the simulation error in x is small. As a result, the embedding may fit poorly without including the excess coordinates of z in the minimization.

Remark 5. It is worth emphasizing two important properties of Problem (26). First, it is an *unconstrained* optimization problem, but imposes guranteed stability on the identified lifted linear model. The parameter set Θ is the space of real numbers of the appropriate dimensionality. Second, there exists a differentiable mapping from the parameters θ to the objective for any choice of differentiable mapping $\phi(\cdot, \theta_{\mathtt{NN}})$, e.g. using the parameterization (24) with $\varphi(\cdot, \theta_{\mathtt{NN}})$ as a neural network. Regarding the resulting nonlinear model, it can be expressed as $x_{t+1} = \phi^{\mathtt{L}}(A\phi(x_t))$. Due to learning errors, the learnt mapings ϕ and $\phi^{\mathtt{L}}$ may not satisfy $\phi^{\mathtt{L}} \circ \phi = \mathbb{I}_d$. While it is not straightforward to verify stability of the identified nonlinear model in the x-coordinate, we can still ensure the attractivity of the equilibrium.

Remark 6. In fact, the enforcement of stability in learned or identified dynamical models has been extensively studied, both within Koopman-based frameworks and beyond. Most approaches in the literature rely on constrained optimization, typically requiring specialized algorithms [31, 49]. In [31], stability is enforced at every iteration step by projecting the solution onto the feasible set to handle the constraints. The recent work [29] provides a probabilistic Koopman framework based on Gaussian processes, which quantifies model uncertainty in the context of learning dynamical models.

Remark 7. The recent work [38] proposes a Koopman learning framework for *continuous-time* systems, providing stability guarantees and discussing the quantification of uncertainties in the learning process.

That work considers the nonlinear reconstruction of observables from Koopman eigenfunctions, akin to the Koopman mapping $\phi(x)$ in our proposed approach. However, the technical route to impose stability constraints differs: [38] is based on *diagonalizable* matrices for the lifted linear system, whereas our work does not impose such a constraint.

Remark 8. The aforementioned properties enable finding a local optimum to Problem (26) using any off-the-shelf first-order optimizer in conjunction with an automatic differentiation (autodiff) toolbox. This significantly simplifies the implementation of the framework. Using an autodiff software package, one only needs to write code that evaluates the objective function at each iteration of the optimization process, and the gradients w.r.t. θ are automatically computed via the chain rule. While the cost function in (26) is nonconvex, deep learning methods have been shown to be effective at finding approximate local minima for such problems. However, these methods do not guarantee proximity to the global minimum.

5 Imitation learning

In this section, we address the motivating application P2 on imitation learning by extending the framework in Section 4. The main task of imitation learning is to obtain a control policy that reproduces the demonstrated trajectories from a given plant. In this paper, we propose to simultaneously learn a stabilizable model of the dynamics, which acts as a form of regularization encouraging closed-loop stability of the learned policy. To this end, we begin in this section by proposing a class of stabilizable Koopman models and study its properties, before turning to the problem of imitation learning.

5.1 Stabilizable Koopman model class

It is a well-known fact that extending the Koopman operator to control systems is technically challenging and may yield *bilinear* lifted systems [16], which are a special class of nonlinear systems. To obtain a *bona fide* linear lifted model, we focus on a particular class of nonlinear systems, which are referred to as "generalized feedback linearizable systems".

Definition 4. For the system (6), if we can find mappings $\alpha : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ and $\phi : \mathbb{R}^n \to \mathbb{R}^N$ ($N \geq n$), and the matrices $A \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{N\times m}$ satisfying

C1: The algebraic equation

$$B^{\perp}[\phi \circ f_c(x, v) - A\phi(x)] = 0, \quad \forall v \in \mathbb{R}^m \tag{27}$$

with $f_c(x, v) := f(x) + g(x)\alpha(x, v)$;

C2: The mapping ϕ is injective.

Then we call the system (6) generalized feedback linearizable. In addition, if the pair (A, B) is stablizable, we refer to it as Koopman stabilizable.

In the above definition, C1 means that for the closed-loop dynamics, we have $\phi(x_{t+1}) = \phi(f(x_t, v_t)) = A\phi(x_t) + Bv_t$ under the pre-feedback $u = \alpha(x, v)$. By defining $z := \phi(x)$ and viewing v as the new input, the control model is lifted into an LTI system

$$z_{t+1} = Az_t + Bv_t. \tag{28}$$

The injectivity condition C2 guarantees that the coordinate z can be pulled back to the original x-coordinate. The above definition covers all feedback linearizable systems that involve a pre-feedback and a state *diffeomorphism* $z = \phi(x)$, a concept central to constructive nonlinear control over the past three decades [19, 40]. See [19, Thm. 4.2.3] for a necessary and sufficient condition of feedback linearizability and [2] for a discrete-time version. In [37], this class of nonlinear systems is called

"immersed by feedback into a linear system", and the authors provide a *local* version of the necessary and sufficient condition via the differential geometric approach.[7]

We are now in the position to propose the model class for imitation learning.

M2. Stabilizable Koopman Model $(A, B, \alpha, \phi, \phi^{\text{L}})$:

$$\begin{aligned} z_0 &= \phi(x_0) \\ z_{t+1} &= Az_t + Bv_t \\ x_t &= \phi^{\text{L}}(z_t) \\ u_t &= \alpha(x_t, v_t) \end{aligned} \tag{29}$$

in which 1) $z \in \mathbb{R}^N$ $(N \geq n)$ is a lifted internal variable; 2) The pair (A, B) is stabilizable; and 3) ϕ has a left inverse satisfying $\phi^{\text{L}}(\phi(x)) = x,\ \forall x$.

5.1.1 Stabilization criterion for generalized feedback linearizable systems

Proposition 2. Assume the system (6) is Koopman stabilizable under the C^1-continuous pre-feedback α and an immersion $\phi : \mathbb{R}^n \to \mathbb{R}^N$ and $(\frac{\partial \phi}{\partial x})^\top \frac{\partial \phi}{\partial x}$ is uniformly bounded. Then, any matrix K that achieves Schur stability of $(A+BK)$ renders the closed-loop system $x_{t+1} = f_c(x_t, K\phi(x_t))$ contracting.

Proof. The system (6) under the pre-feedback $u = \alpha(x, v)$ becomes $x_{t+1} = f_c(x_t, v_t)$. Combining the above, the stabilizing feedback $v = K\phi(x)$, and the algebraic equation (27), one gets

$$\phi \circ f_c(x, K\phi) = (A + BK)\phi. \tag{30}$$

Taking its partial derivative w.r.t. x, one gets

$$\begin{aligned} \frac{\partial \phi}{\partial x}(x_{t+1}) \left[\frac{\partial f_c}{\partial x}(x_t, \cdot) - \frac{\partial f_c}{\partial v}(x_t, \cdot) K \frac{\partial \phi}{\partial x}(x_t) \right] \\ = (A + BK) \frac{\partial \phi}{\partial x}(x_t). \end{aligned} \tag{31}$$

The stabilizing controller for (28) implies the existence of a matrix $P \succ 0$ such that

$$P - (A + BK)^\top P (A + BK) \succ I \succeq \rho I, \tag{32}$$

with $\rho \in (0, 1]$. It yields

$$\left(\frac{\partial \phi}{\partial x} \right)^\top [P - (A + BK)^\top P (A + BK)] \frac{\partial \phi}{\partial x} \succ \rho \left(\frac{\partial \phi}{\partial x} \right)^\top \frac{\partial \phi}{\partial x}. \tag{33}$$

We now consider a candidate contraction metric $M(x) := \frac{\partial \phi}{\partial x}(x)^\top P \frac{\partial \phi}{\partial x}(x)$, and substitute (31) into (33), obtaining

$$\begin{aligned} M(x_t) - \frac{\partial f_x}{\partial x}(x_t)^\top M(x_{t+1}) \frac{\partial f_x}{\partial x}(x_t) &\succ \rho \frac{\partial \phi}{\partial x}(x_t)^\top \frac{\partial \phi}{\partial x}(x_t) \\ &\succeq \beta M(x_t) \\ \beta &:= \tfrac{\rho}{\lambda_{\max}(P)} \end{aligned}$$

in which, for convenience, we have defined a new function $f_x(x) := f_c(x, K\phi(x))$. By choosing $\rho \in (0, 1]$ sufficiently small, we have $0 < \beta < 1$. Therefore, the closed-loop dynamics $x_{t+1} = f_c(x_t, K\phi(x_t))$ is contracting. □

[7] The necessary and sufficient condition in [37] is proposed for *affine* discrete-time systems. For ease of presentation, we also consider systems in the affine form (6).

The matching equation (27) is closely connected to the condition in control contraction metrics (CCM), originally proposed in [32], and a continuous-time version of the connection between CCM and Koopman stabilizability is revealed in [52, Sec. VI-A]. In general, finding the mappings α and ϕ analytically is non-trivial. In Section 5.2, we will explore how to simultaneously learn these mappings within the imitation learning framework and discuss the approximation.

5.1.2 Parameterization of Koopman stabilizable systems

In the proposed Koopman stabilizable model class, we need to identify four components: the stabilizable matrix pair (A, B), a pre-feedback $\alpha(x, v)$, the mapping ϕ, and its left inverse $\phi^{\mathtt{L}}$. To facilitate the learning framework, in this section, we study how to parameterize them.

Earlier works on learning controllers have used linear matrix inequality (LMI) constraints to impose stabilizability [18, 54]. However, the computation of constrained optimization problems becomes extremely expensive when jointly estimating the system dynamics. In the following, an *unconstrained* parameterization of the triple (A, B, K) is proposed such that the pair (A, B) is guaranteed to be stabilizable.

Proposition 3. Consider a pair $(A, B) \in \mathbb{R}^{N\times N} \times \mathbb{R}^{N\times m}$, in which $\operatorname{rank}\{B\} = m$ and A is parameterized as $A(\theta_{\mathtt{SL}})$

$$\begin{aligned} A(\theta_{\mathtt{SL}}) &= \begin{bmatrix} B^{\perp} \\ B^{\top} \end{bmatrix}^{-1} \begin{bmatrix} 2B^{\perp}(M_{11} + M_{22} + R - R^{\top})^{-1} M_{21} \\ S \end{bmatrix}, \\ \theta_{\mathtt{SL}} &:= (L, R, S, B) \end{aligned} \tag{34}$$

where M_{ij} $(i, j = 1, 2)$ are blocks in

$$M := \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = LL^{\top} + \epsilon I, \tag{35}$$

with ϵ a positive scalar, $L \in \mathbb{R}^{2N\times 2N}$, and $R \in \mathbb{R}^{N\times N}$. Then, the pair (A, B) is stabilizable. Conversely, for any stabilizable pair (A, B), we can always find $\theta_{\mathtt{SL}}$ and ϵ to parameterize it in the form of (34).

Proof. The stabilizability of the pair (A, B) is equivalent to the existence of a matrix $K \in \mathbb{R}^{m\times n}$ such that $A_{\mathtt{CL}} := A + BK$ is Schur stable. Invoking Proposition 1, it is necessary and sufficient to have matrices L, R and ϵ to parameterize the closed-loop system matrix $A_{\mathtt{CL}}$ as

$$A_{\mathtt{CL}} = 2(M_{11} + M_{22} + R - R^{\top})^{-1} M_{21}. \tag{36}$$

Note that the matrix

$$\operatorname{rank} \begin{bmatrix} B^{\perp} \\ B^{\top} \end{bmatrix} = n, \tag{37}$$

and we multiply it to both sides of (36), obtaining

$$\begin{bmatrix} B^{\perp} \\ B^{\top} \end{bmatrix} A = \begin{bmatrix} 2B^{\perp}(M_{11} + M_{22} + R - R^{\top})^{-1} M_{21} \\ S \end{bmatrix}$$

with $S := -B^{\top}BK + 2B^{\top}(M_{11} + M_{22} + R - R^{\top})^{-1} M_{21}$. Considering $\operatorname{rank}\{B^{\top}B\} = m$ and the freedom of K, hence S is a free variable to parameterize A. Since all the above implications are necessary and sufficient, we complete the proof. □

Remark 9. In the parameterization, the sub-block M_{22} qualifies as a Lyapunov matrix P due to $M_{22} - (A + BK)^{\top} M_{22}(A + BK) \succ 0$, in which case $K = \frac{1}{2} B^{\top} P A_{\mathtt{CL}}$. On the other hand, there are infinite numbers of feasible selections for K to guarantee Schur stability.

In the control case, the mapping ϕ shares the same properties as the one in Section 4, and thus we adopt the same parameterizations of ϕ and its left inverse $\phi^{\mathtt{L}}$ as done in Section 4.2. The nonlinear function $\alpha(x, v)$ can be parameterized using another neural networks. Notably, we found that an invertible mapping from v to u is beneficial for the specific training scheme in the next section. We present empirical results from simulations for this point in Section 6.

5.2 Imitation learning framework

In this section, we apply the stabilizable Koopman model to the problem of imitation learning (IL). The objective of IL is to learn a control policy that reproduces trajectories of the system (6) demonstrated by an expert policy $u_t = k_\star(x_t)$, given only state-control trajectories $\mathcal{E}_{\mathtt{D}} := \{\tilde{x}_t, \tilde{u}_t\}_{t=1}^{N_{\mathtt{traj}}}$. One well-studied and widely-used paradigm for IL frames it as a supervised learning problem and directly fits a mapping from state to control input. This is commonly referred to as behavioral cloning [3], which aims to minimize the cost function $J_{\mathtt{BC}} = \min_{\theta\in\Theta} \sum_{i=1}^{N_{\mathtt{traj}}} |\tilde{u}_t^i - k_\theta(\tilde{x}_t^i)|^2 + r(\theta)$, with $r(\cdot)$ a regularization function.

In this paper, the problem of learning stabilizing controllers using the proposed Koopman model is considered, particularly applying to IL (i.e. Problem P2). Recently, some works have studied enforcing certain dynamical constraints (such as stability) on the controller during learning, under the assumption of known dynamics [18, 48, 54].

In this work, stability is used to regularize IL when the dynamics are unknown. Our proposed approach is to jointly learn a stabilizable model and a control policy that stabilizes it. We make use of the stabilizable model set **M2** in Section 5 and use the demonstration dataset $\mathcal{E}_{\mathtt{D}}$ to estimate the parameters $\theta := (L, R, S, B, \theta_{\mathtt{NN}}, \theta_{\mathtt{L}})$ Similar to the autonomous case, we use the following *unconstrained* optimization problem – containing the simulation error and a stability regularization penalty term – to learn a stabilizing feedback:

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \left(c_1 J'_{\mathtt{SE}} + c_2 J_{\mathtt{SL}} + c_3 \alpha J'_{\mathtt{RE}} \right) \tag{38}$$

in which we have defined the following functions:

$$\begin{aligned} J'_{\mathtt{SE}} &= \sum_{t=1}^{T-1} |\tilde{z}_{t+1} - A\tilde{z}_t - B\tilde{v}_t|^2 \\ J_{\mathtt{SL}} &= \sum_{t=1}^{T-1} |\tilde{z}_{t+1} - A_{\mathtt{CL}}\tilde{z}_t|^2 + \left| \tilde{v}_t - \frac{1}{2} B^\top P A_{\mathtt{CL}} \tilde{z}_t \right|^2 \\ J'_{\mathtt{RE}} &= \sum_{t=1}^{T} |\tilde{x}_t - \phi^{\mathtt{L}}(\phi(\tilde{x}_t))|^2 \end{aligned} \tag{39}$$

with $T = N_{\mathtt{traj}}$, $A_{\mathtt{CL}} = A + BK$, weighting coefficients $c_i > 0$ $(i = 1, 2, 3)$, and $\tilde{z}_t = \phi(\tilde{x}_t)$ and $\tilde{v}_t = \alpha^{-1}(\tilde{x}_t, \tilde{u}_t)$, invoking that the selected function $\alpha(\cdot, \cdot)$ is bijective.

The first term $J'_{\mathtt{SE}}$ is the "open-loop" simulation error in the lifted z-coordinate, i.e. treating $\tilde{v}$ as an exogenous input, over stabiliziable pairs (A, B); the second term $J_{\mathtt{SL}}$ can be viewed as the simulation error in the lifted coordinate in closed-loop, over Schur-stable matrices $A_{\mathtt{CL}}$, along with a term similar to behavioural cloning for $\tilde{v}$; finally, similarly to the autonomous case, the last term $J'_{\mathtt{RE}}$ is to ensure the left invertibility of ϕ and learn its left inverse. Just like in the autonomous case, the above optimization can be solved via first-order methods using automatic differentiation software.

6 Simulation results

6.1 Simulations: Learning stable Koopman embedding

The approach in Section 4 is validated on the LASA handwriting dataset [22], which consists of human-drawn trajectories of various letters and shapes.[8] It has been widely used as a benchmark for learning stable systems [41]. Stability is an important constraint for the system characterized by this dataset as unconstrained models can have spurious attractors, leading to poor generalization to unseen initial conditions.

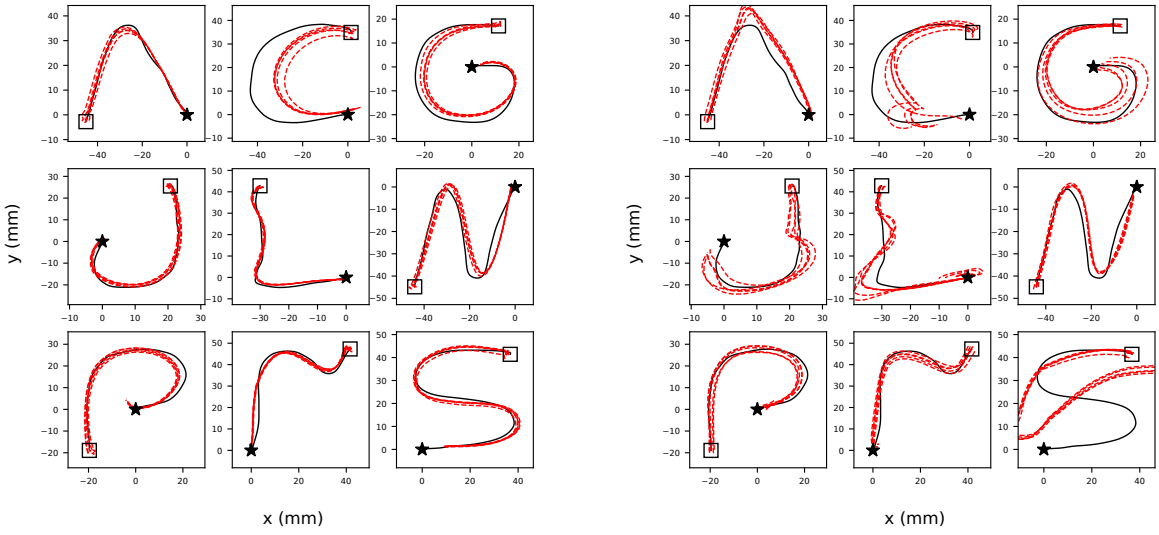
A discrete-time model was trained for each shape in the dataset to follow a desired path from any initial condition. To prepare the data for learning models, splines were fitted to the trajectories and the datapoints were re-sampled at a uniform time interval. The system state was chosen as $\tilde{x}_t = [y_t^\top, \dot{y}_t^\top]^\top \in \mathbb{R}^4$, where $y_t \in \mathbb{R}^2$ and $\dot{y}_t \in \mathbb{R}^2$ are the position and velocity vectors of the end-effector at time t, formulating the minimal realization to this system. All data was scaled to the range $[-1, 1]$ before training. For each shape in the dataset, leave-one-out cross validation was performed. Test trajectories are plotted in Fig. 2a as solid black lines for a subset of the shapes in the dataset. The proposed learning framework was implemented in PyTorch[9] and the ADAM optimizer [23] was used to solve the optimization problem in (26). Hyperparameter values were chosen to be $\alpha = 10^3$ and $\epsilon = 10^{-8}$. All instances of φ were selected as fully-connected feedforward neural networks using rectified linear units (ReLU) as the activation function with its parameter b, 2 hidden layers with 50 nodes each, and an output dimensionality of 20. The neural network parameters $\theta_{\texttt{NN}}$ and $\theta_{\texttt{L}}$ are initialized using the default scheme in PyTorch, while L, R, and b are initialized randomly from a uniform distribution. In the simulation results, our proposed framework is denoted as SKEL.

We compared with a constrained stable parameterization (SOC) in [31] and an unconstrained parameterization (LKIS) in [44] which does not have stability guarantees. We kept most aspects when solving optimization the same when fitting different models, using the normalized simulation error $\texttt{NSE} = (\sum_{t=1}^{T} |\hat{x}_t - \tilde{x}_t|^2)/(\sum_{t=1}^{T} |\tilde{x}_t|^2)$, where $\{\hat{x}\}_{t=1}^{T}$ is the simulated trajectory using the learned model. A boxplot of the normalized simulation error for the three methods is shown in Fig. 3. Our method achieves the lowest median NSE on the test set with 95% confidence. From Fig. 4, it can be seen that LKIS attains the lowest training error, but does not generalize to the test set and SKEL. This can be interpreted as a symptom of overfitting and shows that the stability guarantees of SKEL have a regularizing effect on the model. With regards to SOC, it was observed that the constrained optimization problem would often converge to poor local minima, which is reflected in the relatively high training and test errors. A qualitative evaluation was performed to determine the robustness of the models to small perturbations in the initial condition of the test trajectory. Only SKEL and LKIS were compared as it was clear from Fig. 3 that SOC underperformed in this setting. The results are plotted in Figs. 2a-2b. It can be seen that SKEL produces trajectories that converge to each other due to their contractivity, whereas the LKIS models sometimes behave unpredictably, indicating instability of the learned model.

6.2 Simulations: Imitation learning

Planar Robotic Manipulator. The approach in Section 5 has been validated on the same LASA handwriting dataset, which was used to demonstrate trajectories to be imitated. The data were generated by a simulated 2 degree-of-freedom (DoF) robot, whose dynamics at the end-effector can be simplified as 2-DoF fully-actuated if we are only concerned with the working space rather than the configuration space. It has a standard Euler-Lagrange form and is *feedback linearizable* due to full-actuation, thus satisfying the key assumptions. Since the original LASA dataset only contains state trajectories, an inverse dynamics model was used to generate torques as control inputs for the imitation learning problem. Further details on the model used are given in [11, Sec. 5].

[8] https://cs.stanford.edu/people/khansari/download.html
[9] https://github.com/pytorch/pytorch



(a) SKEL (ours) (b) LKIS [44]

Figure 2: Simulations of SKEL and LKIS models on test data. Trajectories from the models are shown as red dotted lines. True trajectory is shown as a solid black line, with the endpoint denoted by $\star$. Initial conditions are sampled from the square region.

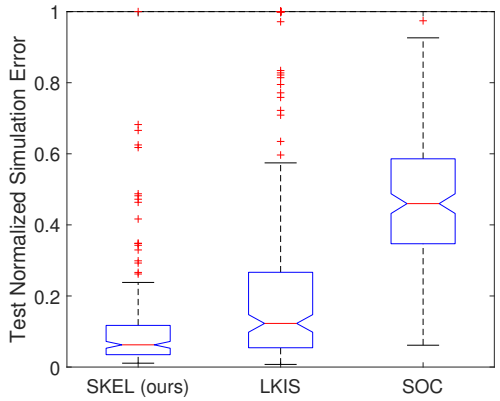


Figure 3: Comparison of SKEL with other Koopman learning methods. Outliers were clipped for better visibility of boxes. Number of outliers with NSE > 1 from left to right: 1 (SKEL), 15 (LKIS), 0 (SOC).

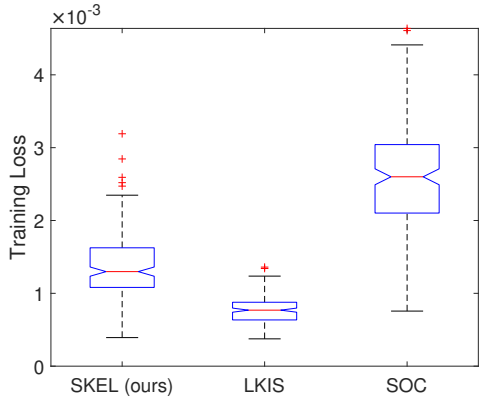


Figure 4: Training loss for each method

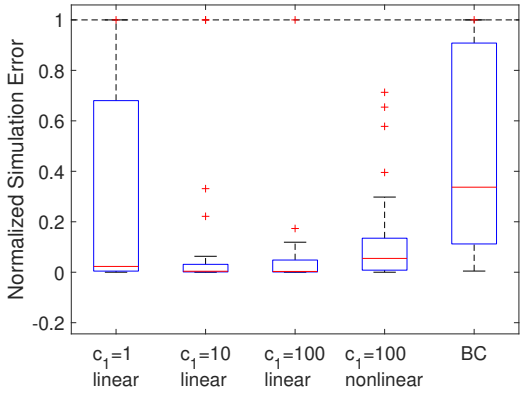


Figure 5: Normalized simulation error of learned controllers on the test set. From left to right: linear parameterization of α — $c_1 = 1$, $c_1 = 10$ and $c_1 = 100$, nonlinear parameterization of α — $c_1 = 100$, behavioural cloning (BC). Number of clipped outliers from left to right: 4, 2, 1, 0, 5.

Comparisons were made of the performance of the learned controller for various values of c_1, and also made against the standard behavioral cloning (BC) method, which is commonly used as a baseline for evaluating imitation learning algorithms [14]. Behavioral cloning was implemented as fitting a neural network mapping states to control inputs by minimizing a mean-squared error loss on the controller output. The neural networks were chosen to have 2 hidden layers with 20 nodes and tanh activations. For a quantitative comparison, normalized simulation error (NSE) was used as a metric, namely $\sum_{t=1}^{T-1} |\tilde{x}_t - \hat{x}_t|^2 / \sum_{t=1}^{T-1} |\tilde{x}_t|^2$, where $\hat{x}_{t+1} = f(\hat{x}_t) + g(\hat{x}_t)k(\hat{x}_t)$ and $\hat{x}_1 = \tilde{x}_1$. Regarding the prefeedback function α, we consider two bijecitive choices: 1) $u = v$ having been used in learning-based control [17, 20, 25]; 2) an affine coupling layer [10]:

$$u = v \odot \exp(s(x)) + h(x),$$

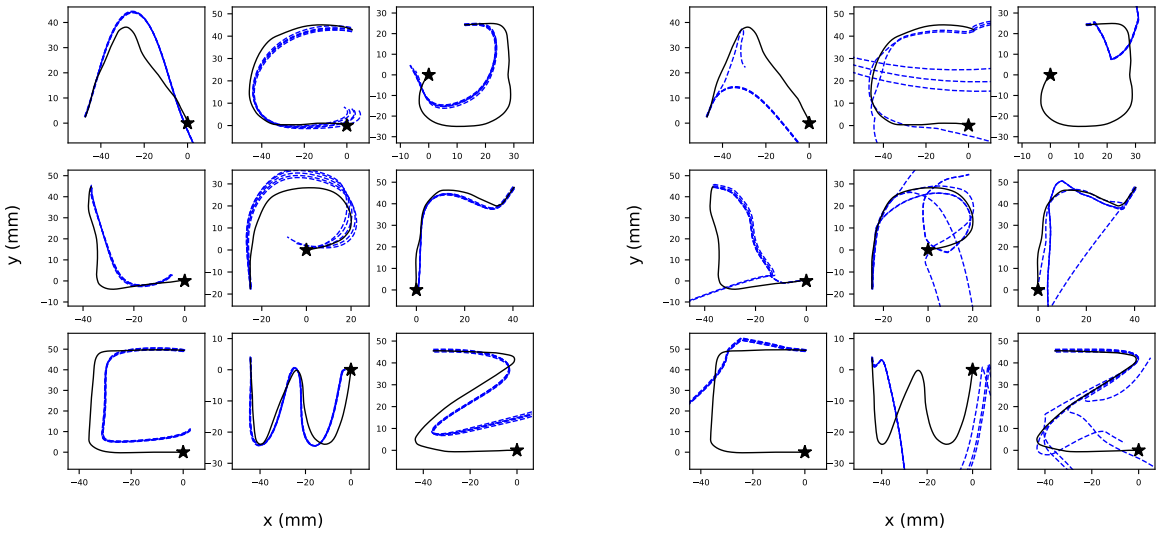
where $\odot$ denotes the Hadamard product, and s, h can be arbitrary function approximators. The second choice includes the nonlinear function h, which adds flexibility by allowing more degrees of freedom to model the nonlinearity in α, while still having an analytical inverse

$$v = (u - h(x)) \odot \exp(-s(x)).$$

The bijectivity is particularly useful for enhancing traning performance.[10]

Fig. 5 shows the normalized simulation error with the learned controllers. We note that increasing c_1 reduces the NSE up to a point, beyond which performance deteriorates. Besides, increasing the model complexity by using the nonlinear parameterization for α does not reduce the NSE, possibly because the small training dataset was insufficient for training larger models. Meanwhile, the BC approach has a substantially larger NSE than the best-performing controller from our proposed method, which shows that the proposed stability-based regularization does indeed improve the performance of the controller over the baseline. A comparison of the trajectories produced by the learned controllers is shown in Fig. 6. It can be seen that the controller produced by our method induces a closed-loop in which nearby trajectories remain close to each other and converge to a single equilibrium, as expected for contracting systems, whereas the trajectories of the BC controller results in divergent trajectories even with small perturbations to the initial condition, which is unacceptable when controlling physical systems. The proposed learning framework provides an obvious improvement over behavioral cloning for the same requirements on the data and without a significant increase in computational cost. The results suggest that the proposed approach does have a regularizing effect on learning stabilizing controllers and outperforms the BC method in terms of imitation error.

[10]There are various methods for learning ivertible maps, e.g. invertible residual networks [7] and BiLipNet [51].



(a) Proposed method with linear α and $c_1 = 100$. **(b) Behavioural cloning**

Figure 6: Simulations of learned controllers on the real system. Trajectories produced by the controllers are shown as blue dotted lines. The true trajectory is shown as a solid black line, with the endpoint denoted by the star.

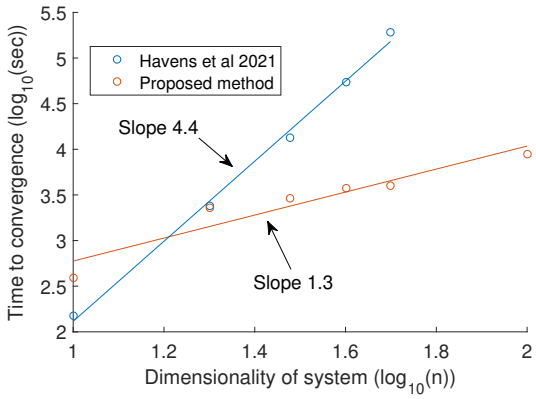


Figure 7: Scatter plot of total time to convergence of the proposed method vs. the PGD algorithm of [18] in log-log scale, plus lines of best fit.

Scability. To evaluate the benefits of direct parameterization vs constrained optimization, we investigated scalability of the learning framework a linear example in [8] which models an unstable graph Laplacian system. We artificially generated trajectory data for learning, with additional details in [11, Sec. 5]. A comparison was made against a prior stability-constrained imitation learning method [18] that requires exact knowledge of (A, B). Their method was applied to this problem set by first estimating (A, B) via least squares.[11] The scalability of both algorithms was evaluated by measuring computation time to convergence of the optimization problems. For the proposed method, this corresponds to the time taken to compute the gradient and update the parameters. In comparison, the projected gradient descent (PGD) algorithm proposed by [18] requires solving a semidefinite program at each iteration. Fig. 7 shows the total convergence time. The slopes of the lines of best fit reveal how the computation times scale with the dimensionality of the system. It can be seen that the proposed method is substantially more scalable, demonstrating the advantage in scalability of optimizing an unconstrained model.

[11] Note that the method of estimating the open-loop dynamics in [18] is not applicable or extensible to nonlinear systems.

7 Conclusion

We introduced new classes of Koopman models with stability and stabilizability guarantees, which are built upon our novel theoretical connections between the contraction and Koopman stability criteria in the paper. The stable Koopman model has been applied to nonlinear system identification, while the stabilizable Koopman model class has shown efficacy in solving imitation learning. In both cases, we proposed parameterization methods to obtain *unconstrained* optimization problems to significantly reduce computation burden. Testing on the renowned LASA handwriting dataset demonstrated that our approaches outperform pervious methods lacking stability guarantees.

References

[1] B. Ahn, C. Kim, Y. Hong, and H. J. Kim. Invertible monotone operators for normalizing flows. Adv. Neural Inf. Process. Syst., 35:16836–16848, 2022.

[2] E. Aranda-Bricaire, Ü. Kotta, and C. H. Moog. Linearization of discrete-time systems. SIAM J. Control Optim., 34(6):1999–2023, 1996.

[3] M. Bain and C. Sammut. A framework for behavioural cloning. In Mach. Intell., pages 103–129, 1995.

[4] R. Bhatia. Matrix Analysis. Springer, 1996.

[5] L. Brivadis, V. Andrieu, and U. Serres. Luenberger observers for discrete-time nonlinear systems. In IEEE Conf. Decis. Control, pages 3435–3440. IEEE, 2019.

[6] J. Cheng, R. Wang, and I. R. Manchester. Learning stable and passive neural differential equations. arXiv preprint arXiv:2404.12554, 2024.

[7] N. De Cao, W. Aziz, and I. Titov. Block neural autoregressive flow. In Uncertain. Artif. Intell., pages 1263–1273. PMLR, 2020.

[8] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu. On the sample complexity of the linear quadratic regulator. Found. Comput. Math., 20(4):633–679, 2020.

[9] R. L. Devaney. A First Course in Chaotic Dynamical Systems: Theory and Experiment. CRC Press, 1992.

[10] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. In Int. Conf. Learn. Represent. OpenReview.net, 2017.

[11] F. Fan. Learning Stable Koopman Models for Identification and Control of Dynamical Systems. PhD thesis, 2023. The University of Sydney.

[12] F. Fan, B. Yi, D. Rye, G. Shi, and I. R. Manchester. Learning stable Koopman embeddings. In Proc. Am. Control Conf., pages 2742–2747, 2022.

[13] B. Fichera and A. Billard. Learning dynamical systems encoding non-linearity within space curvature. arXiv preprint arXiv:2403.11948, 2024.

[14] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. arXiv preprint arXiv:2004.07219, 2020.

[15] N. Gillis, M. Karow, and P. Sharma. A note on approximating the nearest stable discrete-time descriptor systems with fixed rank. Appl. Numer. Math., 148:131–139, 2020.

[16] D. Goswami and D. A. Paley. Bilinearization, reachability, and optimal control of control-affine nonlinear systems: A koopman spectral approach. IEEE Trans. Autom. Control, 67(6):2715–2728, 2021.

[17] Y. Han, W. Hao, and U. Vaidya. Deep learning of Koopman representation for control. In IEEE Conf. Decis. Control, pages 1890–1895. IEEE, 2020.

[18] A. Havens and B. Hu. On imitation learning of linear control policies: Enforcing stability and robustness constraints via LMI conditions. In Am. Control Conf., pages 882–887, 2021.

[19] A. Isidori. Nonlinear Control Systems. Springer, 3 edition, 1995.

[20] E. Kaiser, J. N. Kutz, and S. L. Brunton. Data-driven discovery of Koopman eigenfunctions for control. Mach. Learn. Sci. Technol., 2(3):035023, 2021.

[21] H. K. Khalil. Nonlinear Systems. Prentice Hall, 2002.

[22] S. M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with Gaussian mixture models. IEEE Trans. Robot., 27(5):943–957, 2011.

[23] D. P. Kingma and J. Ba. ADAM: A method for stochastic optimization. In Int. Conf. Mach. Learn., 2015.

[24] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. Proc. Natl. Acad. Sci. U.S.A., 17(5):315–318, May 1931.

[25] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica, 93:149–160, 2018.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Adv. Neural Inf. Process. Syst., pages 1097–1105. Curran Associates, Inc., 2012.

[27] S. L. Lacy and D. S. Bernstein. Subspace identification with guaranteed stability using constrained optimization. IEEE Trans. Autom. Control, 48(7):1259–1263, 2003.

[28] Y. Lan and I. Mezić. Linearization in the large of nonlinear systems and Koopman operator spectrum. Phys. D., 242(1):42–53, 2013.

[29] Y. Lian and C. N. Jones. On Gaussian process based Koopman operators. IFAC-PapersOnLine, 53(2):449–455, 2020.

[30] W. Lohmiller and J.-J. E. Slotine. On contraction analysis for non-linear systems. Automatica, 34(6):683–696, 1998.

[31] G. Mamakoukas, O. Xherija, and T. Murphey. Memory-efficient learning of stable linear dynamical systems for prediction and control. In Adv. Neural Inf. Process. Syst., pages 13527–13538. Curran Associates, Inc., 2020.

[32] I. R. Manchester and J.-J. E. Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. IEEE Trans. Autom. Control, 62(6):3046–3053, 2017.

[33] G. Manek and J. Z. Kolter. Learning stable deep dynamics models. In Adv. Neural Inf. Process. Syst., volume 32. Curran Associates, Inc., 2019.

[34] H. Mania, M. I. Jordan, and B. Recht. Active learning for nonlinear system identification with guarantees. J. Mach. Learn Res., 23(1):1433–1462, 2022.

[35] D. Martinelli, C. L. Galimberti, I. R. Manchester, L. Furieri, and G. Ferrari-Trecate. Unconstrained parametrization of dissipative and contracting neural ordinary differential equations. In IEEE Conf. Decis. Control, pages 3043–3048. IEEE, 2023.

[36] A. Mauroy and I. Mezić. Global stability analysis using the eigenfunctions of the Koopman operator. IEEE Trans. Autom. Control, 61(11):3356–3369, 2016.

[37] S. Monaco and D. Normand-Cyrot. The immersion under feedback of a multidimensional discrete-time non-linear system into a linear system. Int. J. Control, 38(1):245–261, 1983.

[38] S. Pan and K. Duraisamy. Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. SIAM J. Appl. Dyn. Syst., 19(1):480–509, 2020.

[39] M. Revay, R. Wang, and I. R. Manchester. Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness. IEEE Trans. Autom. Control, pages 1–16, 2023.

[40] S. Sastry. Nonlinear Systems: Analysis, Stability, and Control. Springer, 1999.

[41] M. Schonger and et al. Learning barrier-certified polynomial dynamical systems for obstacle avoidance with robots. In IEEE Int. Conf. Robot. Autom., pages 17201–17207. IEEE, 2024.

[42] D. Silver et al. Mastering the game of Go with deep neural networks and tree search. Nature, 529:484, January 2016.

[43] S. Singh, S. M. Richards, V. Sindhwani, J.-J. E. Slotine, and M. Pavone. Learning stabilizable nonlinear dynamics with contraction-based regularization. Int. J. Robot. Res., 40(10-11):1123–1150, 2021.

[44] N. Takeishi, Y. Kawahara, and T. Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In Adv. Neural Inf. Process. Syst., pages 1130–1140, 2017.

[45] M. M. Tobenkin, I. R. Manchester, and A. Megretski. Convex parameterizations and fidelity bounds for nonlinear identification and reduced-order modelling. IEEE Trans. Autom. Control, 62(7):3679–3686, 2017.

[46] M. M. Tobenkin, I. R. Manchester, J. Wang, A. Megretski, and R. Tedrake. Convex optimization in identification of stable non-linear state space models. In IEEE Conf. Decis. Control, pages 7232–7237. IEEE, 2010.

[47] G. Q. B. Tran and P. Bernard. Arbitrarily fast robust KKL observer for nonlinear time-varying discrete systems. IEEE Trans. Autom. Control, (3):1520–1535, 2024.

[48] S. Tu, A. Robey, T. Zhang, and N. Matni. On the sample complexity of stability constrained imitation learning. In Learn. Dyn. Control Conf., pages 180–191, 2022.

[49] J. Umenberger and I. R. Manchester. Specialized interior-point algorithm for stable nonlinear system identification. IEEE Trans. Autom. Control, 64(6):2442–2456, 2019.

[50] J. Umenberger, J. Wågberg, I. R. Manchester, and T. B. Schön. Maximum likelihood identification of stable linear dynamical systems. Automatica, 96:280–292, 2018.

[51] R. Wang, K. Dvijotham, and I. R. Manchester. Monotone, bi-Lipschitz, and Polyak-Lojasiewicz networks. In Int. Conf. Mach. Learn., 2024.

[52] B. Yi and I. R. Manchester. On the equivalence of contraction and Koopman approaches for nonlinear stability and control. IEEE Trans. Autom. Control, 69(7):4336–4351, 2024.

[53] B. Yi, R. Wang, and I. R. Manchester. Reduced-order nonlinear observers via contraction analysis and convex optimization. IEEE Trans. Autom. Control, 67(8):4045–4060, 2021.

[54] H. Yin, P. Seiler, M. Jin, and M. Arcak. Imitation learning with stability and safety guarantees. IEEE Control Syst. Lett., 6:409–414, 2021.