

Physics-Informed Machine Learning for Characterizing System Stability

Tomoki Koike* and Elizabeth Qian†
Georgia Institute of Technology, Atlanta, Georgia, 30332

In the design and operation of complex dynamical systems, it is essential to ensure that all state trajectories of the dynamical system converge to a desired equilibrium within a guaranteed stability region. Yet, for many practical systems—especially in aerospace—this region cannot be determined a priori and is often challenging to compute. One of the most common methods for computing the stability region is to identify a Lyapunov function. A Lyapunov function is a positive function whose time derivative along system trajectories is non-positive, which provides a sufficient condition for stability and characterizes an estimated stability region. However, existing methods of characterizing a stability region via a Lyapunov function often rely on explicit knowledge of the system governing equations. In this work, we present a new physics-informed machine learning method of characterizing an estimated stability region by inferring a Lyapunov function from system trajectory data that treats the dynamical system as a black box and does not require explicit knowledge of the system governing equations. In our presented Lyapunov function Inference method (LyapInf), we propose a quadratic form for the unknown Lyapunov function and fit the unknown quadratic operator to system trajectory data by minimizing the average residual of the Zubov equation, a first-order partial differential equation whose solution yields a Lyapunov function. The inferred quadratic Lyapunov function can then characterize an ellipsoidal estimate of the stability region. Numerical results on benchmark examples demonstrate that our physics-informed stability analysis method successfully characterizes a near-maximal ellipsoid of the system stability region associated with the inferred Lyapunov function without requiring knowledge of the system governing equations.

I. Nomenclature

t	= Time variable.
n	= State dimension.
$\mathbb{R}^n, \mathbb{R}^{m \times n}$	= Space of real n vectors and $m \times n$ matrices.
$\mathbf{x}$	= State vector, $\mathbf{x}(t) \in \mathbb{R}^n$.
$\dot{\mathbf{x}}$	= Time derivative, $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} \in \mathbb{R}^n$.
$f(\mathbf{x})$	= Nonlinear function, $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, defining $\dot{\mathbf{x}} = f(\mathbf{x})$.
$\mathcal{D}_0$	= True stability region.
$\mathcal{D}(c)$	= Estimated stability region: $\{\mathbf{x} : V(\mathbf{x}) \leq c, \dot{V} < 0\}$.
$V, \tilde{V}$	= True and approximate Lyapunov functions, $V, \tilde{V} : \mathbb{R}^n \rightarrow \mathbb{R}$.
$h(\mathbf{x})$	= Auxiliary function in Zubov equation, $h : \mathbb{R}^n \rightarrow \mathbb{R}$.
$\mathbf{P}$	= Matrix parametrizing the quadratic Lyapunov function $V(\mathbf{x}) = \mathbf{x}^\top \mathbf{P} \mathbf{x}$.
$\mathbf{Q}$	= Matrix parametrizing the quadratic form $h(\mathbf{x}) = \mathbf{x}^\top \mathbf{Q} \mathbf{x}$.
N	= Number of snapshot data points.
$\mathbf{X} = [\mathbf{x}(t_1), \dots, \mathbf{x}(t_N)]$	= State snapshot matrix in $\mathbb{R}^{n \times N}$.
$\dot{\mathbf{X}} = [\dot{\mathbf{x}}(t_1), \dots, \dot{\mathbf{x}}(t_N)]$	= Derivative snapshot matrix in $\mathbb{R}^{n \times N}$.
$\otimes$	= Kronecker product.
$\ \cdot\ _2, \ \cdot\ _F$	= Euclidean and Frobenius norms.
$\mathcal{X}$	= Specified operating region of a system.

*PhD Candidate, School of Aerospace Engineering, AIAA Student Member, tkoike@gatech.edu

†Assistant Professor, Schools of Aerospace Engineering & Computational Science and Engineering, AIAA Member, eqian@gatech.edu

II. Introduction

In the design of complex nonlinear systems, engineers encounter phenomena such as bifurcations, limit cycles, and chaotic attractors. Stable limit cycles can support sustained periodic operation, whereas instabilities can drive trajectories away from equilibrium into unsafe regimes. To distinguish safe from unsafe initial conditions of such different system behaviors, engineers can study the *stability region* (also called the region of attraction or domain of attraction), defined as the set of initial states whose trajectories converge to the equilibrium. Knowing this region reveals the safe operating regime and how far nonlinear effects may be pushed before stability is lost. Reliable estimation of the stability region can therefore provide insight into the system’s response to varying initial conditions, perturbations, and parameters, guiding decision-making and design for systems in various fields, including robotics [1, 2], power systems [3], and fluid dynamics [4]. In aerospace applications, stability region estimations led to enlarging safe flight envelopes for an F-8 aircraft short-period model [5] and identifying the safe regions of a 4D longitudinal dynamics of NASA’s Generic Transport Model (GTM) aircraft [6].

One of the most common methods for assessing system stability is Lyapunov’s method. This method assesses stability by finding a smooth, *energy-like* function $V(\mathbf{x})$ defined to be zero at the equilibrium and strictly positive for all other states in a specified region. Additionally, V must always be non-increasing along every trajectory in that same region, i.e., its time derivative is non-positive. If those two conditions hold everywhere in the state space, the equilibrium is globally asymptotically stable; if they hold only inside some region around the equilibrium (the stability region), they guarantee only local stability. Moreover, the sublevel sets of V serve as inner estimates of the stability region [7]. Thus, identifying a Lyapunov function is a key step in designing and validating engineering systems that must remain stable despite inherent variations and uncertainties.

Most classical Lyapunov-based methods rely on explicit knowledge of the system’s governing equations to determine a stability region by constructing a Lyapunov function. Analytical methods may assume a parametric form for the gradient of V and solve for coefficients that enforce $\dot{V} < 0$ (variable-gradient method [8]), or solve the Zubov equation—a first order partial differential equation (PDE) which is known to govern the dynamics of the Lyapunov equation—via polynomial or rational expansions [9–11]. Advances in convex optimization have given rise to computational approaches that pose stability certification as linear programs for piecewise-linear V [12, 13] and semidefinite programs (SDPs) for quadratic or higher-order Lyapunov functions [14, 15]. These SDPs include classical Krasovskii’s matrix-inequality formulation [16] and modern sum-of-squares (SOS) programs, which enforce nonnegativity of V by requiring a polynomial to decompose into sums of squares [17, 18]. We emphasize that all of these methods require knowledge of the governing equations *a priori* to set up either the analytical conditions or the constraints that guarantee $\dot{V} < 0$, thereby certifying the equilibrium’s stability region. However, many complex multidisciplinary systems such as those in aerospace design lack fully known governing equations, precluding direct application of these tools and motivating data-driven stability analysis methods.

Building on this motivation, data-driven approaches for estimating safe stability regions have gained increasing attention [19]. For example, [20] proposes a Lyapunov function in Taylor-series form: the quadratic term is obtained via an SDP-based linear matrix inequality, and higher-order terms are modeled with a neural network. Although their approach requires no simulated trajectory data, it still relies on explicit knowledge of the system’s governing equations, since the Taylor-neural Lyapunov function is trained under a reformulated Zubov-equation constraint that directly maximizes the stability region. Another line of data-driven methods leverages the Koopman operator, which is an infinite-dimensional linear operator that evolves observables (functions of the system states) over time. In practice, a finite set of the Koopman operator’s leading eigenfunctions is learned, and then the Lyapunov function is constructed by forming a linear combination of those spectral components [21, 22]. In [23], the authors further demonstrate how such eigenfunctions can be used to approximate the boundary of the stability region directly using the eigenvalues and eigenfunctions of the Koopman operator. Another work [24] partially automates the selection of observables by learning the eigenfunctions using autoencoder networks. However, when the system governing equations are unknown, selecting an appropriate dictionary of observables or tuning the autoencoder to ensure the observables span the key modes of the infinite-dimensional operator remains a challenge.

In this work, we introduce a scientific machine learning approach that embeds domain knowledge, in the form of the Zubov equation, directly into a data-driven Lyapunov function inference method. We restrict $V(\mathbf{x})$ to a quadratic form and learn its quadratic operator by minimizing the average residual of the Zubov equation over available snapshot and time derivative data. This physics-informed training enforces the $\dot{V} < 0$ condition and produces an ellipsoidal estimate of the stability region from trajectory data, without requiring explicit knowledge of the system’s governing equations. Another physics-informed machine learning work using data to approximate solutions to the Zubov equation is [25], which trains a neural network on simulation outputs of an augmented dynamical system to approximate an integral-form

solution of the Zubov equation. In contrast, our approach fits our candidate Lyapunov function directly to trajectory data from the system itself. Our approach is related to that of [26, 27], which fits a neural network form Lyapunov function to system trajectory data by minimizing the residual of the Zubov equation. However, while neural networks offer considerable flexibility in expressing complex nonlinear functions, they require large amounts of data to train and are difficult to interpret: verification of the associated domain of attraction in [26, 27] requires the use of satisfiability modulo theory (SMT) solvers to verify formal conditions relating the domain of attraction of the neural network to that of an associated quadratic Lyapunov function. In contrast, our direct consideration of a quadratic form requires less data to train and allows the resulting Lyapunov function to be more easily interpreted and analyzed. There also exist works which combine Zubov’s theory with Koopman operators: [28] solves a modified Zubov equation in the lifted Koopman space, and [29] extends this framework to jointly learn both the governing dynamics and the Lyapunov function as well as the stability region. Although these Zubov-Koopman methods exploit the Koopman operator’s ability to represent complex nonlinear dynamics as linear evolution in an infinite-dimensional function space and thus capture diverse behaviors via a rich dictionary of observables, they still rely on a neural network embedded in the framework to approximate the Zubov equation solution. This reliance restricts them to an SMT-solver-based verification procedure (as in [26]) and makes their internal structure challenging to interpret compared to our explicit quadratic Lyapunov function.

The contributions of this article are: (i) a new physics-informed method of learning a stability region by inferring a quadratic Lyapunov function from system trajectory data that treats the dynamical system as a black box and does not require explicit knowledge of the system governing equations; and (ii) numerical demonstration on benchmark dynamical systems that shows how our approach successfully identifies a near-maximal ellipsoid estimate of the stability region for a given operating regime and its corresponding Lyapunov function.

The remainder of this work is organized as follows. Section III describes the problem formulation and the key theories by Lyapunov and Zubov. Section IV introduces our method. Section V demonstrates our method with several numerical examples. Finally, Section VI concludes and discusses directions for future work.

III. Background

A. System Definition

Consider an autonomous n -dimensional nonlinear dynamical system

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)), \quad \mathbf{x}_0 = \mathbf{x}(0), \quad t \in [0, t_f], \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the state vector of the system, with t in a given interval $[0, t_f]$ where $0 < t_f < \infty$. The function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is assumed to be locally Lipschitz continuous to ensure the existence and uniqueness of a solution to the initial value problem (1). Although we express the dynamics explicitly in terms of f , our method treats f as a black box and does not require knowledge of its form.

B. Lyapunov Stability Theory

Without loss of generality, let the origin be an equilibrium point of (1). The stability of the system about the origin is defined as follows [30]:

Definition III.1 *Consider the nonlinear dynamical system (1). The origin is said to be Lyapunov stable if, for any $\epsilon > 0$ there exists a $\delta > 0$, such that $\|\mathbf{x}(t)\| < \epsilon$ is true for all $t \geq 0$ whenever $\|\mathbf{x}_0\| < \delta$. Moreover, the origin is asymptotically stable if it is Lyapunov stable and there exists a $\delta > 0$ such that $\lim_{t \rightarrow \infty} \mathbf{x}(t) = 0$ whenever $\|\mathbf{x}_0\| < \delta$.*

Lyapunov stability merely guarantees boundedness of trajectories that start near the origin, whereas the stronger condition of asymptotic stability guarantees that trajectories converge to the origin when initialized close enough to the origin. The full set of initial conditions $\mathbf{x}_0$ leading to system trajectories that converge to the origin is called the *stability region*, denoted $\mathcal{D}_0 \subseteq \mathbb{R}^n$ [30]:

$$\mathcal{D}_0 \triangleq \left\{ \mathbf{x}_0 \in \mathbb{R}^n : \text{if } \mathbf{x}(0) = \mathbf{x}_0, \text{ then } \lim_{t \rightarrow \infty} \mathbf{x}(t) = 0 \right\}. \quad (2)$$

The conditions for stability in Definition III.1 and the definition of the stability region in eq. (2) are given in terms of the long-term behavior of solutions to eq. (1) and are impractical to verify for general nonlinear dynamical systems. Instead, the following theorem of Lyapunov [7] provides sufficient conditions for stability and an estimate of the stability region in terms of a Lyapunov function $V(x)$:

Theorem III.2 (Lyapunov's Theorem [7]) Consider the nonlinear dynamical system (1). If there exists a continuously differentiable function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying $V(0) = 0$ and

$$V(\mathbf{x}) > 0, \quad \dot{V}(\mathbf{x}) = \nabla V(\mathbf{x})^\top f(\mathbf{x}) < 0,$$

for all $\mathbf{x}$ in a neighborhood of the origin, then the origin is locally asymptotically stable. Furthermore, for any $c > 0$, the set

$$\mathcal{D}(c) \triangleq \{\mathbf{x} : V(\mathbf{x}) \leq c, \dot{V} < 0\} \quad (3)$$

is a subset of the stability region, i.e., $\mathcal{D}(c) \subset \mathcal{D}_0$.

The identification of *any* Lyapunov function satisfying the conditions of Theorem III.2 is a sufficient condition for system stability. Moreover, a subset of the true stability region can be characterized by a sublevel set of V , as depicted in Figure 1. Zubov's theorem, introduced in the next section, provides conditions characterizing a *specific* Lyapunov function that characterizes the entire stability region of the nonlinear system.

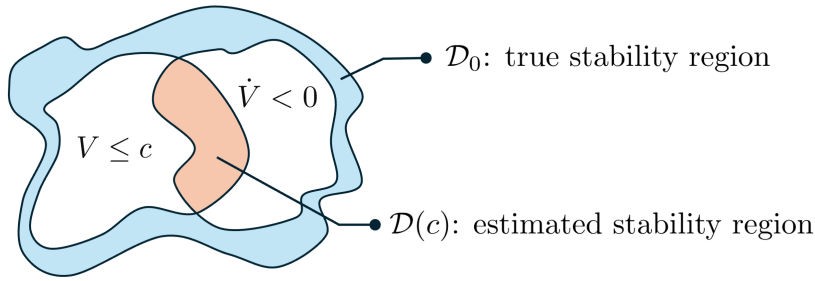


Fig. 1 Illustration of the estimated stability region characterized by some Lyapunov function V .

C. Stability analysis based on Zubov's Theorem

Zubov's theorem defines a specific Lyapunov function as the solution to a first-order PDE:

Theorem III.3 (Zubov's Theorem [9, 30]) Let $h : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuous with $h(0) = 0$ and $h(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \{0\}$. Suppose there exists a continuously differentiable function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying $V(0) = 0$ and

$$\dot{V}(\mathbf{x}) = \nabla V(\mathbf{x})^\top f(\mathbf{x}) = -h(\mathbf{x})[1 - V(\mathbf{x})]. \quad (4)$$

Then, if $\mathcal{D} \subset \mathbb{R}^n$ contains the origin and

- 1) $0 < V(\mathbf{x}) < 1$, for all $\mathbf{x} \in \mathcal{D}$, and
- 2) $V(\mathbf{x}) \rightarrow 1$ as $\mathbf{x} \rightarrow \partial\mathcal{D}$ or $\|\mathbf{x}\| \rightarrow \infty$,

then the system (1) is asymptotically stable about the origin with the stability region of $\mathcal{D}_0 = \mathcal{D}$.

The first-order PDE (4) is known as the *Zubov equation*. One way to characterize the entire stability region of the nonlinear system is therefore to identify a Lyapunov function satisfying the Zubov equation and an associated domain $\mathcal{D}$ satisfying the additional conditions in Theorem III.3. However, closed-form solutions to Zubov's equation have been found only for a limited number of systems.

For nonlinear systems with analytic f , Zubov demonstrated that if h is a quadratic function, then the solution to the Zubov equation (4) can be represented as an infinite series:

$$V(\mathbf{x}) = V_2 + V_3 + \cdots = \sum_{i=2}^{\infty} V_i(\mathbf{x}), \quad (5)$$

where $V_i(\mathbf{x})$ are homogeneous polynomial functions of $\mathbf{x}$ with degree i [31, 32]. Truncating this series at degree p yields an approximation of the Lyapunov function:

$$\tilde{V}(\mathbf{x}) = V_2 + V_3 + \cdots + V_p = \sum_{i=2}^p V_i(\mathbf{x}) \quad (6)$$

which is a Lyapunov function satisfying the conditions of Theorem III.2 and thus characterizes a subset of the stability region [9]. One way to compute a Lyapunov function is to use knowledge of f to recursively compute terms in this series [10]. In contrast, we propose a new method that truncates the series at $p = 2$, yielding a quadratic form for the Lyapunov function, and we will infer the unknown quadratic coefficients from data without requiring knowledge of the system governing equations.

IV. Physics-Informed Machine Learning for Characterizing the System Stability Region

This section presents our physics-informed method for learning the stability region using our proposed Lyapunov function inference method (LyapInf) which infers a quadratic Lyapunov function from system trajectory data and provides us with an ellipsoidal estimate of the stability region. Section IV.A introduces the mathematical formulation of the method. Sections IV.B and IV.C discuss implementation of the method and its dependence on user-defined inputs. Section IV.D describes a Monte Carlo procedure for estimating the largest stability region associated with the inferred quadratic Lyapunov function.

A. Mathematical formulation of the method

We assume that we have a data set of N pairs of state and time derivative data, $\{\mathbf{x}(t_i), \dot{\mathbf{x}}(t_i)\}_{i=1}^N$, where the time derivative data may be extracted directly from a black box solver that evaluates f at $\mathbf{x}(t_i)$, or approximated from the state data using standard finite difference schemes, e.g.,

$$\dot{\mathbf{x}}(t_i) = \frac{\mathbf{x}(t_i) - \mathbf{x}(t_{i-1})}{t_i - t_{i-1}}, \quad \text{for } i = 1, 2, \dots, N. \quad (7)$$

To simplify the exposition, we restrict our notation in this section to the setting where the N data pairs arise from N timesteps of a single state trajectory. However, the formulation can be straightforwardly extended to data sets with multiple trajectories starting from different initial conditions.

We seek to fit a quadratic Lyapunov function that minimizes the residual of the Zubov equation over the data. Or in other words, we inform physics knowledge from the system control domain in the form of the Zubov equation into a data-fitting problem. We assume f in (1) is analytic, and we define $h(\mathbf{x}) = \mathbf{x}^\top \mathbf{Q} \mathbf{x}$, where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is symmetric positive definite. We then adopt a quadratic form for an approximate solution to Zubov's equation, $\tilde{V}(\mathbf{x}) = \mathbf{x}^\top \mathbf{P} \mathbf{x}$, where $\mathbf{P} \in \mathbb{R}^{n \times n}$ is also symmetric positive definite. Substituting the quadratic forms for h and $\tilde{V}$ into (4) and re-arranging terms yields the residual of the Zubov equation for our approximation:

$$(\text{Residual}) \quad \mathcal{R}(\mathbf{P}, \mathbf{x}, \dot{\mathbf{x}}) = \dot{\mathbf{x}}^\top \mathbf{P} \mathbf{x} + \mathbf{x}^\top \mathbf{P} \dot{\mathbf{x}} + \mathbf{x}^\top \mathbf{Q} \mathbf{x} - \mathbf{x}^\top \mathbf{Q} \mathbf{x} \mathbf{x}^\top \mathbf{P} \mathbf{x}. \quad (8)$$

To fit $\mathbf{P}$ to data, we then solve

$$\tilde{\mathbf{P}} = \arg \min_{\mathbf{P} \in \mathcal{M}_n} \frac{1}{N} \sum_{i=1}^N \mathcal{R}(\mathbf{P}, \mathbf{x}(t_i), \dot{\mathbf{x}}(t_i)), \quad (9)$$

where $\mathcal{M}_n$ is the manifold of symmetric positive definite matrices in $\mathbb{R}^{n \times n}$. The inferred Lyapunov function is then given by $\tilde{V}(\mathbf{x}) = \mathbf{x}^\top \tilde{\mathbf{P}} \mathbf{x}$.

B. Algorithmic implementation

We now describe how to implement the mathematical formulation of the minimization problem from Section IV.A as a concrete algorithm. Let $\otimes$ denote the Kronecker product, $\text{vec}(\cdot) : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{mn}$ the vectorization (*flattening*) of a matrix, and $\text{mat}(\cdot) : \mathbb{R}^{mn} \rightarrow \mathbb{R}^{m \times n}$ the inverse operator of vectorization. For computational implementation of the mathematical procedure we have described, we will find it convenient to define $\mathbf{p} = \text{vec}(\mathbf{P})$ and $\mathbf{q} = \text{vec}(\mathbf{Q})$, as well as the k -times repeated Kronecker product of the state vector:

$$\mathbf{x}^{\otimes k} = \underbrace{\mathbf{x} \otimes \dots \otimes \mathbf{x}}_{k\text{-times}} \in \mathbb{R}^{n^k}, \quad (10)$$

e.g., $\mathbf{x} \otimes \mathbf{x} = \mathbf{x}^{\otimes 2}$. Using Kronecker product identities from [33], we can rewrite eq. (8) as:

$$\mathcal{R}(\mathbf{P}, \mathbf{x}, \dot{\mathbf{x}}) = \mathbf{p}^\top \mathbf{x}^{\otimes} + \mathbf{q}^\top \mathbf{x}^{\otimes 2} - (\mathbf{q} \otimes \mathbf{p})^\top \mathbf{x}^{\otimes 4} \quad (11)$$

where $\mathbf{x}^\otimes = \dot{\mathbf{x}} \otimes \mathbf{x} + \mathbf{x} \otimes \dot{\mathbf{x}}$. This lets us reformulate (9) as the following constrained minimization:

$$\tilde{\mathbf{p}} = \arg \min_{\mathbf{p} \in \mathbb{R}^{n^2}} \frac{1}{N} \sum_{j=1}^N \left\| \mathbf{p}^\top \mathbf{x}_j^\otimes + \mathbf{q}^\top \mathbf{x}_j^{\otimes 2} - (\mathbf{q} \otimes \mathbf{p})^\top \mathbf{x}_j^{\otimes 4} \right\|_2^2 \quad \text{subject to } \mathbf{P} = \text{mat}(\mathbf{p}) = \mathbf{P}^\top > 0. \quad (12)$$

This reformulation makes clear that the optimization objective and constraints are both convex, enabling us to compute the solution of the constrained optimization using existing convex optimization packages. In order to use such packages, we find it convenient to reformulate the optimization in terms of the Frobenius norm of the matrix whose columns contain the residuals over all data. The optimization problem can then be written as:

$$\tilde{\mathbf{p}} = \arg \min_{\mathbf{p} \in \mathbb{R}^{n^2}} \frac{1}{N} \left\| \mathbf{p}^\top \mathbf{X}^\otimes + \mathbf{q}^\top \mathbf{X}^{\otimes 2} - (\mathbf{q} \otimes \mathbf{p})^\top \mathbf{X}^{\otimes 4} \right\|_F^2 \quad \text{subject to } \mathbf{P} = \text{mat}(\mathbf{p}) = \mathbf{P}^\top > 0 \quad (13)$$

where we define the data matrices

$$\mathbf{X}^\otimes = \begin{bmatrix} | & & | \\ \mathbf{x}^\otimes(t_1) & \cdots & \mathbf{x}^\otimes(t_N) \\ | & & | \end{bmatrix} \in \mathbb{R}^{n^2 \times N} \quad (14)$$

$$\mathbf{X}^{\otimes k} = \begin{bmatrix} | & & | \\ \mathbf{x}^{\otimes k}(t_1) & \cdots & \mathbf{x}^{\otimes k}(t_N) \\ | & & | \end{bmatrix} \in \mathbb{R}^{n^k \times N}. \quad (15)$$

In our numerical experiments in Section V, we solve (13) in the Julia language using the JuMP [34] and Splitting Conic Solver [35] packages. To enforce positive semidefiniteness, we impose the convex constraint $\mathbf{P} \succeq 0$. We then promote strict positive definiteness by additionally requiring the diagonal elements of $\mathbf{P}$ to satisfy $P_{ii} \geq \varepsilon$ for all i where $\varepsilon > 0$ is a small constant. Although a diagonal lower bound by itself guarantees only a positive trace, when combined with $\mathbf{P} \succeq 0$, it effectively pushes all eigenvalues away from zero. Exploring alternative implementation approaches, including manifold optimization or Cholesky factor parameterizations of $\mathbf{P}$, is an area for future work.

C. Algorithm input considerations

Both the training data and the choice of auxiliary function h are algorithm inputs which influence the performance of the method. This section discusses these considerations and describes the choices used in our numerical experiments.

1. Training Data

The inferred Lyapunov function and associated estimate of the stability region will vary depending on the training data set. For the numerical experiments in this work, we follow an approach consistent with recent data-driven methods for estimating stability regions based on Zubov's theory [26, 28, 36], which collect data by randomly sampling from a specified operating regime $\mathcal{X} \subset \mathbb{R}^n$. To generate training data for LyapInf, we simulate the system for initial conditions $\mathbf{x}_0 \in \mathcal{X}$ and collect the trajectory data at each timestep. We then discard the points of the trajectory outside the region of interest $\mathcal{X}$. Developing further analytical understanding of our method's dependence on training data, and potentially new targeted data acquisition strategies, is an area for future work.

2. Choice of Auxiliary Function h

The choice of the quadratic operator $\mathbf{Q}$ defining the auxiliary function h will also influence the resulting inferred Lyapunov function and associated stability region. In this work, we assume $\mathbf{Q} = \gamma \mathbf{I}_n$, where $\gamma > 0$. This scalar γ modulates the influence of the quartic term $-\mathbf{x}^\top \mathbf{Q} \mathbf{x} \mathbf{x}^\top \mathbf{P} \mathbf{x}$ in eq. (13), ensuring that the quartic scaling does not disproportionately affect the objective function during optimization. We determine γ by testing values within $[1e-3, 10]$ and selecting the γ giving the largest estimated stability region. Further theoretical and numerical exploration of the effect of the choice of $\mathbf{Q}$ on our method is a direction for future work.

D. Monte Carlo Estimation of the Stability Region

As discussed in Sections III.B and III.C, the set $\mathcal{D}(c) = \{\mathbf{x} : \tilde{V}(\mathbf{x}) \leq c, \tilde{V} < 0\}$ associated with our inferred quadratic Lyapunov function $\tilde{V}(\mathbf{x})$ is a subset of the stability region. Because $\tilde{V} = \mathbf{x}^\top \mathbf{P} \mathbf{x}$ with positive definite $\mathbf{P}$, the sublevel set

$\{x : \tilde{V}(x) \leq c\}$ is an ellipsoid which expands as c increases. To provide the largest possible ellipsoid estimate of the stability region, we seek the maximum c such that the sublevel ellipsoid remains inside the $\tilde{V} < 0$ region. This maximal c , denoted c_* is the solution to the following optimization problem [37, 38]:

$$c_* = \min_{\mathbf{x} \in \mathbb{R}^n} \tilde{V}(\mathbf{x}) \quad \text{subject to } \tilde{V}(\mathbf{x}) = 0. \quad (16)$$

Unfortunately, this optimization is non-convex and generally challenging to globally solve [37]. Instead, for the relatively low-dimensional examples considered here, we use the Monte-Carlo sampling method proposed by [39] to estimate c_* . This brute-force approach uniformly samples points within a region of interest and sets c_* to be the largest value such that all sampled points within the sublevel set satisfy $\tilde{V} < 0$. One drawback of this approach is that it can require many samples to provide a good c_* estimate, particularly for higher-dimensional systems.

V. Numerical Experiments

We apply our proposed method to a series of benchmark nonlinear dynamical systems, described in Section V.A. Section V.B presents and discusses the results.

A. Test Problems

This section describes six nonlinear dynamical systems commonly used as benchmarks for nonlinear stability analysis that serve as test problems for our method. After introducing each system's governing equations, we describe the initial conditions used to generate system state trajectories used as training data for the method. For each test problem, Table 2 tabulates the timestep size and final time for the trajectory data, as well as the region of interest $\mathcal{X}$, and the scaling γ that defines the auxiliary quadratic operator $\mathbf{Q}$. Trajectory data are obtained by numerical integration using a fourth-order Runge-Kutta method, and time derivative data are directly extracted from the solver.

1. 2D Quadratic System

This first example from [37, 39, 40] is defined by

$$f(\mathbf{x}) = \begin{bmatrix} -2x_1 + x_1x_2 \\ -x_2 + x_1x_2 \end{bmatrix}. \quad (17)$$

The training data are generated using $M = 16$ different initial conditions, equally spaced along a circle of radius 5 centered at the origin and all lying within the domain $\mathcal{X} = [-5, 5]^2$.

2. Van der Pol Oscillator

$$f(\mathbf{x}) = \begin{bmatrix} -x_2 \\ x_1 - \mu(1 - x_1^2)x_2 \end{bmatrix} \quad (18)$$

where $\mu = 1$ is the damping parameter. The training data are generated using $M = 10$ different initial conditions, equally spaced along a circle of radius 1.5 centered at the origin and all lying within the domain $\mathcal{X} = [-3, 3]^2$.

3. Nonlinear Pendulum

$$f(\mathbf{x}) = \begin{bmatrix} x_2 \\ -\sin(x_1) - 0.5x_2 \end{bmatrix}. \quad (19)$$

The training data are generated using $M = 20$ different initial conditions, equally spaced along the boundary of the domain $\mathcal{X} = [-4, 4]^2$.

Table 2 Test problem parameters.

System	$\mathcal{X}$	t_f	Δt	γ
2D Quadratic	$[-5, 5]^2$	5	0.01	1
Van der Pol	$[-3, 3]^2$	5	0.01	2.0
Nonlinear Pendulum	$[-4, 4]^2$	10	0.001	0.2
Trig-Exp Nonlinear	$[-3, 3]^2$	10	0.01	0.01
3D Cubic	$[-3, 3]^3$	5	0.01	1
Networked Van der Pol	$[-4, 4]^{20}$	10	0.01	0.1

4. Trig-Exp Nonlinear System

The fourth example includes high nonlinearity with cosine and exponential terms. This system is an example from [38, 39, 41] defined by

$$f(\mathbf{x}) = \begin{bmatrix} -x_1 + x_2 + 0.5(\exp(x_1) - 1) \\ -x_1 - x_2 + x_1 x_2 + x_1 \cos(x_1) \end{bmatrix}. \quad (20)$$

The training data are generated using $M = 30$ different initial conditions uniformly sampled from the domain $\mathcal{X} = [-3, 3]^2$.

5. 3D Cubic System

This example is a 3D system with cubic nonlinearity from [37, 39] defined by

$$f(\mathbf{x}) = \begin{bmatrix} -x_1 + x_2 x_3^2 \\ -x_2 - x_1 x_2 \\ -x_3 \end{bmatrix}. \quad (21)$$

The training data are generated using $M = 25$ different initial conditions, equally spaced along the surface of a sphere with radius 3 centered at the origin and all lying within the domain $\mathcal{X} = [-3, 3]^3$.

6. 20D Networked Van der Pol Oscillator

Inspired by [26] and [42], this example consists of 10 Van der Pol oscillator subsystems $\{N_i\}_{i=1}^{10}$, where each is defined by:

$$f_i(\mathbf{x}) = \begin{bmatrix} -x_{i2} \\ x_{i1} - \mu_i(1 - x_{i1}^2)x_{i2} + \sum_{j \neq i} \zeta_{ij} x_{i1} x_{j2} \end{bmatrix} \quad (22)$$

where μ_i is the damping parameter uniformly sampled within the parameter domain $[0.5, 2.5]$ for each subsystem. The interconnection strengths between subsystems i and j are defined by parameters ζ_{ij} that are set to 0 with a 50% probability, indicating no connection, or, with a 50% probability, are uniformly chosen from $[-0.1, 0.1]$. The data are generated using $M = 80$ different initial conditions sampled within the domain $\mathcal{X} = [-4, 4]^{20}$. Specifically, for each subsystem we choose points along a circle of radius one.

As mentioned in Section IV.D, the Monte-Carlo procedure to estimate c_* is a challenge for high-dimensional systems. To estimate c_* for our 20D networked Van der Pol oscillator example, we therefore apply this Monte Carlo algorithm to estimate c_* within the 10 individual two-dimensional state spaces associated with each subsystem while fixing the other subsystem coordinates at zero. We then report the results of the most conservative c_* over all 10 subsystems. Future work will investigate alternative strategies for estimating the stability region associated with our inferred Lyapunov function that scale to higher dimensions. Finally, we note that the method implementation as well as numerical examples are available online*.

*Github repository: <https://github.com/smallpondtom/LyapunovFunctionInference.jl>

B. Results and Discussion

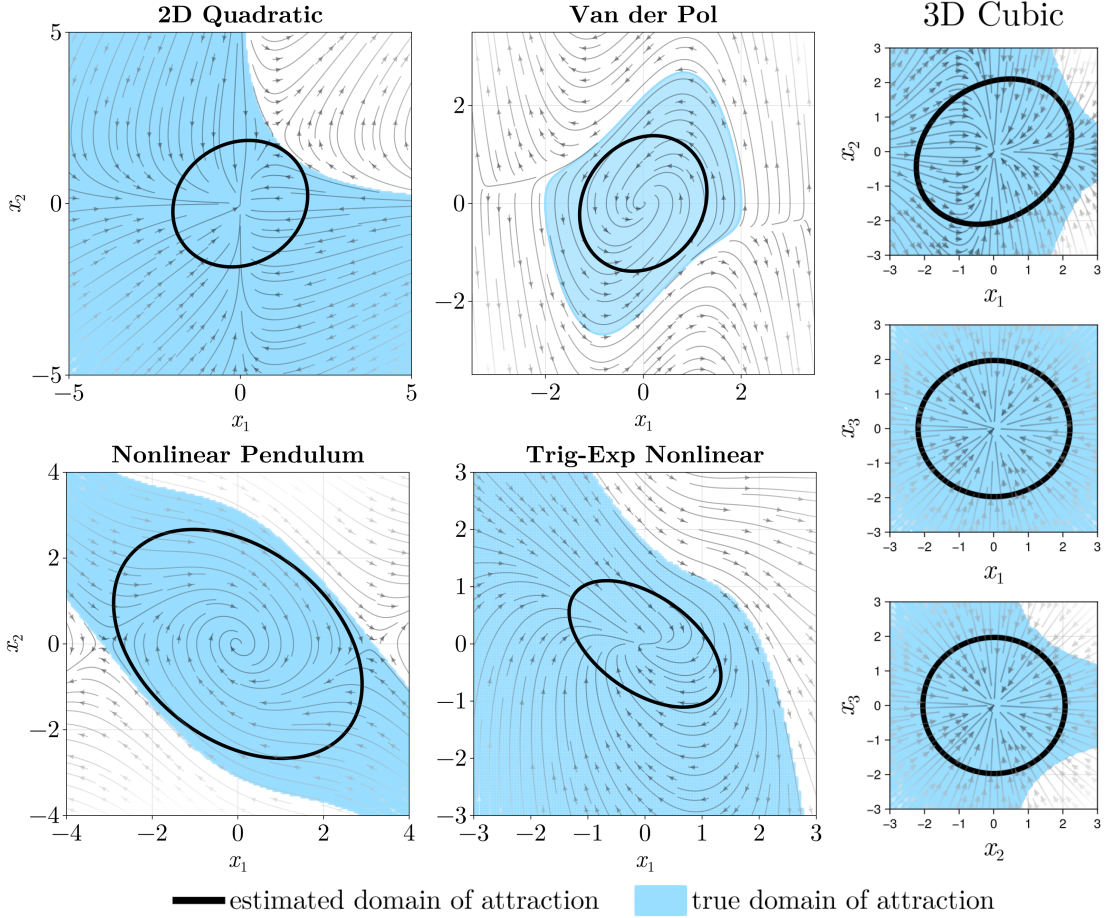


Fig. 2 Overlay of the estimated stability region (interior of black ellipse) and true stability region for 2D numerical examples, and 2D cross-sections of each axis plane for the 3D example.

Figure 2 plots the estimated stability region $\mathcal{D}(c_*)$ obtained using the Monte Carlo procedure described in Section IV.D for the two- and three-dimensional test problems (Examples 1–5): the estimated stability region is the interior of the black ellipse in each plot. For comparison, the shaded blue region in each plot is the true stability region $\mathcal{D}_0$ obtained by direct numerical evaluation for these simple examples. For the 3D cubic system (Figure 2, bottom panel) and the 20D networked Van der Pol system, (Figure 3) we plot the projection of the true and estimated stability regions onto axial planes. For the Van der Pol system, we select the axial planes associated with two-dimensional subsystems and show only results for the first and sixth subsystems, which are representative of the remaining subsystems. Our estimated stability regions are fully contained within the true stability regions. While the estimated stability regions are conservative, they are not overly so: the edges of our estimates are near the boundaries of the true stability region. We note that these ellipsoid estimates are, however, more conservative than estimates obtained in [25, 26], which use neural networks to approximate Lyapunov functions in Zubov-based approaches. This is unsurprising given the greater expressive power of neural networks compared to our quadratic ansatz; the trade-off is the greater data requirements for training neural networks. For example, [26] reports using 300,000 data to train the neural network Lyapunov function for the Van der Pol and networked Van der Pol systems, whereas we can infer a quadratic Lyapunov function using just 5,010 and 80,080 data for these systems, respectively.

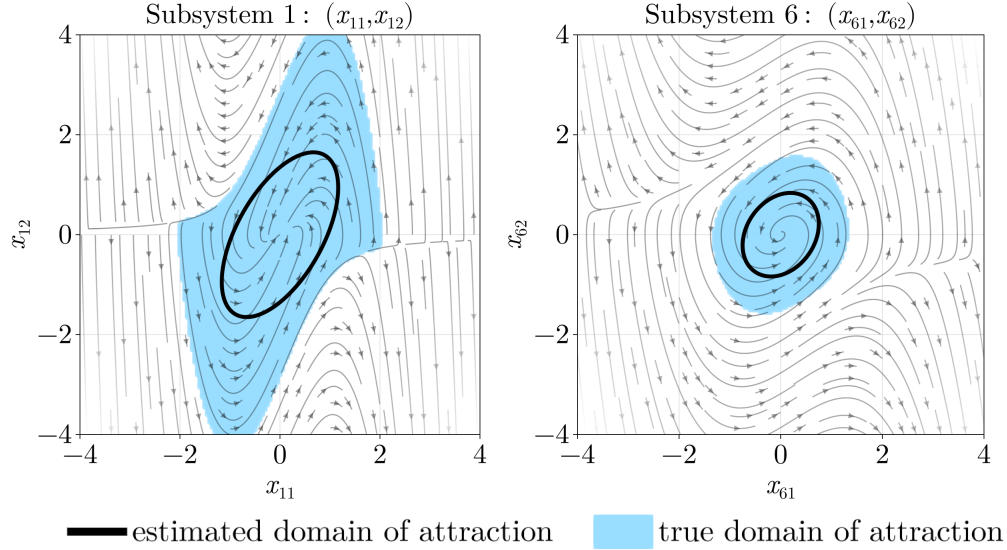


Fig. 3 Estimated stability region (interior of black ellipse) for subsystems 1 and 6 of the 20D networked Van der Pol oscillator.

VI. Conclusion

We have presented a new physics-informed machine learning method for characterizing the stability region of nonlinear dynamical systems to enable the design of systems with known safe operation regimes. The method learns a stability region by inferring a Lyapunov function from state trajectory data, and it is based on embedding Zubov’s theorem, which characterizes a Lyapunov function as the solution to a first-order PDE known as the Zubov equation, into the training process. The proposed method adopts a quadratic approximation for the solution to the Zubov equation and minimizes the average residual of the equation over the data to infer the unknown quadratic operator. We then learn an elliptical stability region characterized by the inferred quadratic Lyapunov function, which is an estimate of the true stability region, indicating a safe region of operation for a given physical system. The method does not require explicit knowledge of the system’s governing equations to estimate the stability region; instead, it leverages physics-informed machine learning to discover operators of Lyapunov functions from data or the underlying physics, making it applicable to black-box systems. Numerical experiments on benchmark examples demonstrate that the method learns near-maximal ellipsoidal estimates of the true stability region for the tested systems with associated inferred quadratic Lyapunov functions.

There are several directions that are the subject of ongoing and future work. We note that like many physics-informed machine learning methods, the performance of our physics-informed stability analysis method is sensitive to both the quality and quantity of data. In our experiments, reducing the number of trajectories in the training data set led to poor estimates of the stability region, likely due to the data covering less of the system phase space. On the other hand, the inclusion in the data set of many unstable trajectories can also potentially lead to poor results, although our experiments do use training data containing some unstable trajectories. Further development of efficient strategies for collecting appropriate training data is a direction of active investigation.

Other research directions include developing computational procedures for efficiently learning higher-order polynomial Lyapunov functions to characterize stability regions with complex geometries, and efficiently estimating these regions for high-dimensional systems arising in multidisciplinary engineering design. For high-dimensional systems such as those arising from PDE discretization, our framework could be applied to projection-based reduced models [43–46] to analyze their stability properties or inform stability guarantees [47–51].

Acknowledgments

This work was supported by the US Department of Energy Office of Science Energy Earthshot Initiative as part of the ‘Learning reduced models under extreme data conditions for design and rapid decision-making in complex systems’ project under award number DE-SC0024721.

References

- [1] Dawson, C., Gao, S., and Fan, C., “Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control,” *IEEE Transactions on Robotics*, Vol. 39, No. 3, 2023, pp. 1749–1767.
- [2] Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A. P., “Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning,” *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 5, No. Volume 5, 2022, 2022, pp. 411–444. <https://doi.org/10.1146/annurev-control-042920-020211>.
- [3] Zhang, Y., Li, Y., Tomsovic, K., Djouadi, S. M., and Yue, M., “Review on set-theoretic methods for safety verification and control of power system,” *IET Energy Systems Integration*, Vol. 2, No. 3, 2020, pp. 226–234.
- [4] Schmid, P. J., and Brandt, L., “Analysis of Fluid Systems: Stability, Receptivity, Sensitivity: Lecture Notes from the FLOW-NORDITA Summer School on Advanced Instability Methods for Complex Flows, Stockholm, Sweden, 2013,” *Applied Mechanics Reviews*, Vol. 66, No. 024803, 2014. <https://doi.org/10.1115/1.4026375>.
- [5] Biswas, B., Ignatyev, D., Zolotas, A., and Tsourdos, A., “Design of Controller with Enlarged Region Of Attraction Using Union Theorem in Sum-of-Squares Optimization for the F-8 Aircraft,” *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, 2024, p. 1018. <https://doi.org/10.2514/6.2024-1018>.
- [6] Lai, B., Cunis, T., and Burlion, L., “Nonlinear Trajectory Based Region of Attraction Estimation for Aircraft Dynamics Analysis,” *AIAA Scitech 2021 Forum*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2021, p. 0253. <https://doi.org/10.2514/6.2021-0253>.
- [7] Lyapunov, A. M., “The general problem of the stability of motion,” *International Journal of Control*, Vol. 55, No. 3, 1992, pp. 531–534. <https://doi.org/10.1080/00207179208934253>.
- [8] Schultz, D. G., and Gibson, J. E., “The Variable Gradient Method for Generating Lyapunov Functions,” *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, Vol. 81, No. 4, 1962, pp. 203–210. <https://doi.org/10.1109/TAI.1962.6371818>.
- [9] Zubov, V. I., *Methods of A.M. Lyapunov and Their Application*, Groningen : P. Noordhoff, 1964. Edited for this English ed. by Leo F. Boron.
- [10] Margolis, S., and Vogt, W., “Control engineering applications of V. I. Zubov’s construction procedure for Lyapunov functions,” *IEEE Transactions on Automatic Control*, Vol. 8, No. 2, 1963, pp. 104–113. <https://doi.org/10.1109/TAC.1963.1105553>.
- [11] Ferguson, D., “Generalisation of Zubov’s construction procedure for Lyapunov functions,” *Electronics Letters*, Vol. 3, No. 6, 1970, pp. 73–74.
- [12] Brayton, R., and Tong, C., “Stability of Dynamical Systems: A Constructive Approach,” *IEEE Transactions on Circuits and Systems*, Vol. 26, No. 4, 1979, pp. 224–234. <https://doi.org/10.1109/TCS.1979.1084637>.
- [13] Pedro Julian, J. G., and Desages, A., “A parametrization of piecewise linear Lyapunov functions via linear programming,” *International Journal of Control*, Vol. 72, No. 7-8, 1999, pp. 702–715. <https://doi.org/10.1080/002071799220876>.
- [14] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear matrix inequalities in system and control theory*, SIAM, 1994.
- [15] Johansson, M., and Rantzer, A., “Computation of Piecewise Quadratic Lyapunov Functions for Hybrid Systems,” *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 555–559. <https://doi.org/10.1109/9.664157>.
- [16] Krasovskii, N., “Problems of the Theory of Stability of Motion,” , 1963.
- [17] Parrilo, P. A., *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, California Institute of Technology, 2000.
- [18] Papachristodoulou, A., and Prajna, S., “On the Construction of Lyapunov Functions Using the Sum of Squares Decomposition,” *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002., Vol. 3, 2002, pp. 3482–3487 vol.3. <https://doi.org/10.1109/CDC.2002.1184414>.
- [19] Drgona, J., Nghiem, T. X., Beckers, T., Fazlyab, M., Mallada, E., Jones, C., Vrabie, D., Brunton, S. L., and Findeisen, R., “Safe Physics-Informed Machine Learning for Dynamics and Control,” , Apr. 2025. <https://doi.org/10.48550/arXiv.2504.12952>.
- [20] Barreau, M., and Bastianello, N., “Learning and Verifying Maximal Taylor-Neural Lyapunov Functions,” , Aug. 2024. <https://doi.org/10.48550/arXiv.2408.17246>.

- [21] Mauroy, A., and Mezić, I., “A spectral operator-theoretic framework for global stability,” *52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 5234–5239.
- [22] Mauroy, A., and Mezić, I., “Global Stability Analysis Using the Eigenfunctions of the Koopman Operator,” *IEEE Transactions on Automatic Control*, Vol. 61, No. 11, 2016, pp. 3356–3369. <https://doi.org/10.1109/TAC.2016.2518918>.
- [23] Umathe, B., and Vaidya, U., “Spectral Koopman Method for Identifying Stability Boundary,” *IEEE Control Systems Letters*, Vol. 8, 2024, pp. 247–252. <https://doi.org/10.1109/LCSYS.2023.3347396>.
- [24] Deka, S. A., Valle, A. M., and Tomlin, C. J., “Koopman-Based Neural Lyapunov Functions for General Attractors,” *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 5123–5128. <https://doi.org/10.1109/CDC51059.2022.9992927>.
- [25] Kang, W., Sun, K., and Xu, L., “Data-Driven Computational Methods for the Domain of Attraction and Zubov’s Equation,” *IEEE Transactions on Automatic Control*, Vol. 69, No. 3, 2024, pp. 1600–1611. <https://doi.org/10.1109/TAC.2023.3326769>.
- [26] Liu, J., Meng, Y., Fitzsimmons, M., and Zhou, R., “Physics-Informed Neural Network Lyapunov Functions: PDE Characterization, Learning, and Verification,” *Automatica*, Vol. 175, 2025, p. 112193. <https://doi.org/10.1016/j.automatica.2025.112193>.
- [27] Zhou, R., Fitzsimmons, M., Meng, Y., and Liu, J., “Physics-Informed Extreme Learning Machine Lyapunov Functions,” *IEEE Control Systems Letters*, Vol. 8, 2024, pp. 1763–1768. <https://doi.org/10.1109/LCSYS.2024.3416407>.
- [28] Meng, Y., Zhou, R., and Liu, J., “Learning Regions of Attraction in Unknown Dynamical Systems via Zubov-Koopman Lifting: Regularities and Convergence,” , Nov. 2023. <https://doi.org/10.48550/arXiv.2311.15119>.
- [29] Zhou, R., Meng, Y., Zeng, Z., and Liu, J., “Learning Koopman-based Stability Certificates for Unknown Nonlinear Systems,” , Apr. 2025. <https://doi.org/10.48550/arXiv.2412.02807>.
- [30] Haddad, W. M., and Chellaboina, V., *Nonlinear dynamical systems and control*, Princeton University Press, Princeton, NJ, 2008.
- [31] Hafstein, S., and Giesl, P., “Review on computational methods for Lyapunov functions,” *Discrete and Continuous Dynamical Systems - Series B*, Vol. 20, No. 8, 2015, p. 2291–2331. <https://doi.org/10.3934/dcdsb.2015.20.2291>.
- [32] Hahn, W., *Stability of motion*, 1967th ed., Grundlehren der mathematischen Wissenschaften, Springer, Berlin, Germany, 1967.
- [33] Brewer, J., “Kronecker Products and Matrix Calculus in System Theory,” *IEEE Transactions on Circuits and Systems*, Vol. 25, No. 9, 1978, pp. 772–781. <https://doi.org/10.1109/TCS.1978.1084534>.
- [34] Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B., and Vielma, J. P., “JuMP 1.0: Recent improvements to a modeling language for mathematical optimization,” *Mathematical Programming Computation*, Vol. 15, 2023, p. 581–589. <https://doi.org/10.1007/s12532-023-00239-3>.
- [35] O’Donoghue, B., Chu, E., Parikh, N., and Boyd, S., “Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding,” *Journal of Optimization Theory and Applications*, Vol. 169, No. 3, 2016, pp. 1042–1068.
- [36] Grüne, L., “Computing Lyapunov Functions Using Deep Neural Networks,” *Journal of Computational Dynamics*, Vol. 8, No. 2, Wed Mar 31 20:00:00 EDT 2021, pp. 131–152. <https://doi.org/10.3934/jcd.2021006>.
- [37] Hachicho, O., “A novel LMI-based optimization algorithm for the guaranteed estimation of the domain of attraction using rational Lyapunov functions,” *Journal of the Franklin Institute*, Vol. 344, No. 5, 2007, p. 535–552. <https://doi.org/10.1016/j.jfranklin.2006.02.032>.
- [38] Matallana, L. G., Blanco, A. M., and Bandoni, J. A., “Estimation of domains of attraction: A global optimization approach,” *Mathematical and Computer Modelling*, Vol. 52, No. 3–4, 2010, p. 574–585. <https://doi.org/10.1016/j.mcm.2010.04.001>.
- [39] Najafi, E., Babuška, R., and Lopes, G. A. D., “A fast sampling method for estimating the domain of attraction,” *Nonlinear Dynamics*, Vol. 86, No. 2, 2016, p. 823–834. <https://doi.org/10.1007/s11071-016-2926-7>.
- [40] Tesi, A., Villoresi, F., and Genesio, R., “On the Stability Domain Estimation via a Quadratic Lyapunov Function: Convexity and Optimality Properties for Polynomial Systems,” *IEEE Transactions on Automatic Control*, Vol. 41, No. 11, Nov./1996, pp. 1650–1657. <https://doi.org/10.1109/9.544002>.
- [41] Chesi, G., “Estimating the domain of attraction for non-polynomial systems via LMI optimizations,” *Automatica*, Vol. 45, No. 6, 2009, pp. 1536–1541.

- [42] Kundu, S., and Anghel, M., “A Sum-of-Squares approach to the Stability and Control of Interconnected Systems Using Vector Lyapunov Functions,” , Jan. 2015.
- [43] Kramer, B., “Stability Domains for Quadratic-Bilinear Reduced-Order Models,” *SIAM Journal on Applied Dynamical Systems*, Vol. 20, No. 2, 2021, p. 981–996. <https://doi.org/10.1137/20m1364849>.
- [44] Peherstorfer, B., and Willcox, K., “Data-Driven Operator Inference for Nonintrusive Projection-Based Model Reduction,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 306, 2016, pp. 196–215. <https://doi.org/10.1016/j.cma.2016.03.025>.
- [45] Berkooz, G., Holmes, P., and Lumley, J. L., “The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows,” *Annual Review of Fluid Mechanics*, Vol. 25, No. 1, 1993, pp. 539–575. <https://doi.org/10.1146/annurev.fl.25.010193.002543>.
- [46] Benner, P., Gugercin, S., and Willcox, K., “A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems,” *SIAM Review*, Vol. 57, No. 4, 2015, pp. 483–531. <https://doi.org/10.1137/130932715>.
- [47] Kaptanoglu, A. A., Callahan, J. L., Aravkin, A., Hansen, C. J., and Brunton, S. L., “Promoting global stability in data-driven models of quadratic nonlinear dynamics,” *Physical Review Fluids*, Vol. 6, No. 9, 2021, p. 094401.
- [48] Sawant, N., Kramer, B., and Peherstorfer, B., “Physics-Informed Regularization and Structure Preservation for Learning Stable Reduced Models from Data with Operator Inference,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 404, 2023, p. 115836. <https://doi.org/10.1016/j.cma.2022.115836>.
- [49] Goyal, P., Duff, I. P., and Benner, P., “Guaranteed stable quadratic models and their applications in SINDy and operator inference,” *Physica D: Nonlinear Phenomena*, 2025, p. 134893.
- [50] Gkimitis, L., Aretz, N., Tezzele, M., Richter, T., Benner, P., and Willcox, K. E., “Non-Intrusive Reduced-Order Modeling for Dynamical Systems with Spatially Localized Features,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 444, 2025, p. 118115. <https://doi.org/10.1016/j.cma.2025.118115>.
- [51] Peng, M., Kaptanoglu, A. A., Hansen, C. J., Stevens-Haas, J., Manohar, K., and Brunton, S. L., “Extending the trapping theorem to provide local stability guarantees for quadratically nonlinear models,” *Physics of Fluids*, Vol. 37, No. 10, 2025.