Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-28 (2024) 282–287

Data-Driven Estimation of Region of Attraction Using Koopman Operator and Reverse Trajectory

Rober Velasco * **Almuatazbellah Boker** * **Lamine Mili** *
Amir Abolmasoumi **

* *Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: robertv3@vt.edu, boker@vt.edu, lmili@vt.edu)*

** *Department of Electrical Engineering, Arak University, Arak, Iran (e-mail: a.abolmasoumi@gmail.com)*

Abstract: We propose to estimate the region of attraction (ROA) for the stability of nonlinear systems from only system measurement data and without knowledge of the system model. The key to our result is the use of Koopman operator theory to approximate the nonlinear dynamics in linear coordinates. This approximation is typically more accurate than the traditional Jacobian-based linearization method. We then employ the Extended Dynamic Mode Decomposition (EDMD) method to estimate the linear approximation of the system through data. This is then used to construct a Lyapunov function that helps estimate the ROA. However, this estimate is typically very conservative. The trajectory reversing method is then used on the set of points that form this conservative estimate, to enlarge the ROA approximation. The output of EDMD is also utilized in the trajectory reversing method, keeping the entire analysis data-driven. Finally, an example is used to show the accuracy of this data-driven method, despite not knowing the system.

Keywords: Region of Attraction, Koopman Operator Theory, Stability of Nonlinear Systems, Machine Learning in Modeling, Estimation, and Control

1. INTRODUCTION

Stability analysis of nonlinear systems is of great importance to analyzing and controlling real-world dynamic systems. It is extremely important to know a system's stable equilibrium points and - just as important - their region of attraction (ROA). A system's ROA is the region of initial conditions surrounding an equilibrium point that will stay within the same region for all time. Furthermore, any initial condition that lies outside of this region will be unstable and diverge to infinity. Although finding equilibrium points for a system can be trivial, determining the ROA of a system's equilibrium point is much harder and can rarely be exactly found through analytical methods. One consistent approach used to estimate the ROA of a system is by constructing Lyapunov functions Mauroy et al. (2020). Previous studies have involved creating an optimal Lyapunov function using learning-based algorithms Chen et al. (2021), recurrent sets Shen et al. (2022), Wang et al. (2023) Gaussian process classification, optimization of the sum of squares Anghel et al. (2013), and more recently the Koopman operator Mauroy and Mezić (2016); Garcia-Tenorio et al. (2023).

Koopman Operator Theory is a recently developed idea that aims to represent a nonlinear system as a linear evolution of functions. It proves to have many advantages in controlling and analyzing dynamic systems Mauroy et al. (2020). The Koopman Operator is infinitely large, but there have been various ways presented to create a finite-sized approximation of it using data-driven methods, specifically Dynamic Mode Decomposition Schmid (2010); Kutz et al. (2016), Outlier-Robust Dynamic Mode Decomposition Abolmasoumi et al. (2022), Noise-Robust Dynamic Mode Decomposition Hemati et al. (2017), and Extended Dynamic Mode Decomposition (EDMD) Williams et al. (2015), which will be covered and applied in this paper. EDMD allows the ability to create the linear Koopman operator for a system without needing the knowledge of the original nonlinear system.

As previously stated, there have been multiple attempts at using the Koopman operator to construct Lyapunov functions in the interest of approximating the ROA. Although some of these methods have very accurate results, their drawback is they consider having full knowledge of the dynamic equations of the system. This can be an optimistic and in some cases unrealistic assumption when working with real-world nonlinear systems, such as power systems. Instead, a more applicable approach is to determine a data-driven method that can approximate the ROA of a nonlinear system. This is where EDMD comes in. Additional research has been conducted in the area of approximating ROA from the results of EDMD. One paper used the dominant eigenfunction to create a "naive" estimate for the ROA Choi and Bose (2018), however, this method would fail for Koopman approximations with multiple dominant eigenfunctions. Another article

10.1016/j.ifacol.2025.01.008

uses the results of EDMD to create a new eigenfunction whose trajectories determine the stability regions Garcia-Tenorio et al. (2023). Although this paper creates a ROA approximation from data-driven Koopman techniques, it specifically analyzes hyperbolic and polynomial systems. Meng et al. (2023) proposes another learning approach utilizing 'Zubov-Koopman' operators and a near-maximal Lyapunov function.

In this paper, we address the problem of estimating the ROA of an equilibrium point of a general nonlinear system. A key to our approach is the use of Koopman operator theory to provide a linear approximation of the nonlinear system. Thanks to the EDMD, we are also able to find some Koopman eigenfunctions which can be used to construct a Lyapunov function candidate. All this can be done on a data-driven basis. The linear approximation is then used to provide a conservative estimate of the ROA using the Laypunov method Khalil (2002). This serves as a starting point for applying the enlargement method Kant et al. (2017) to provide a more accurate estimate. The enlargement method provides sufficient conditions for a convex hull representation of the enlarged estimated ROA using a set of discrete points obtained through backward integration. Relative to the literature, the main advantage of our approach is that it is based on the intuitive Lyapunov approach and relaxes its limitations. Namely, in our approach, there is no difficulty in selecting the Lyapunov function candidate, as we rely on the standard Lyapunov function used for linear systems. Additionally, our approach does not result in a conservative estimate.

The rest of the paper is organized as follows. Background materials will be included in Section 2. Our main results are given in Section 3. In this section, we will present our methods for using the EDMD output to create a conservative estimate for the ROA and then to enlarge the ROA approximation using backward trajectory techniques. The simulation results will be presented in Section 4, and the conclusions will be presented in Section 5.

2. PRELIMINARIES

2.1 Koopman Operator Theory

Consider a nonlinear dynamical system

$$\dot{x} = f(x) \tag{1}$$

where $x \in \mathcal{M}$ and the state space $\mathcal{M} \subseteq \mathbb{R}^n$. Additionally, $f(\cdot)$ is a nonlinear vector-valued function and is used by the map $S_t : \mathcal{M} \to \mathcal{M}$. Let the observable function, ψ, be a function of x such that $\psi : \mathcal{M} \to \mathbb{C}$ and ψ belongs in the function state space $\mathcal{F}$, which we will refer to as the "Koopman Space". Then the Koopman operator, $\mathcal{K}_t$ is a linear, infinite-dimensional operator that acts on ψ (in continuous time) such that Mauroy et al. (2020)

$$\mathcal{K}_t\psi = \psi \circ S_t, \tag{2}$$

where $\mathcal{K}_t$ maps $\mathcal{F} \to \mathcal{F}$. As (2) shows, the Koopman operator acting on an observable function is equivalent to the composition of that function and S_t, the system map of (1). Therefore, the evolution of ψ from the Koopman operator in the Koopman space $\mathcal{F}$ is equivalent to the evolution of x from S_t in the state space $\mathcal{M}$. Therefore, it can be written as follows.

$$\dot{\psi} = \mathcal{K}_t\psi, \tag{3}$$

which describes the dynamics of (1) in the Koopman Space. This allows for the definition of Koopman eigenfunctions, $\phi_i(x)$, and Koopman eigenvalues, λ_i, that satisfy

$$\mathcal{K}_t\phi_i = \lambda_i\phi_i, \tag{4}$$

where $\lambda_i \in \mathbb{C}$, $\phi_i \in \mathcal{F}$, and $i = 1, ..., \infty$. Putting this into the same form as (3) leads to

$$\dot{\phi}_i = \mathcal{K}_t\phi_i = \lambda_i\phi_i. \tag{5}$$

Equations (4) and (5) are significant and show the value of Koopman eigenfunctions and Koopman eigenvalues. These equations represent the nonlinear dynamics of (1) as linear scalar equations in the Koopman Space.

There is additionally the Koopman Operator in discrete time, which we will denote as $\hat{\mathcal{K}}$, which follows many of the same properties as the continuous-time operator $\mathcal{K}_t$. In the discrete case, for a nonlinear dynamical system

$$x_{k+1} = F(x_k), \tag{6}$$

$\hat{\mathcal{K}}$ is defined such that

$$\hat{\mathcal{K}}\psi = \psi \circ F, \tag{7}$$

where $F : \mathcal{M} \to \mathcal{M}$ and $\psi \in \mathcal{F}$. The Koopman eigenfunctions and eigenvalues are written as

$$\hat{\mathcal{K}}\phi_i = \mu_i\phi_i, \tag{8}$$

with $\phi_i \in \mathcal{F}$ and $\mu_i \in \mathbb{C}$. If (1) is discretized as (6) at some sample time Δt, then $\mu_i = e^{\lambda_i \Delta t}$ Williams et al. (2015).

2.2 Extended Dynamic Mode Decomposition (EDMD)

EDMD is a data-driven approach to find the matrix $\hat{K} \in \mathbb{C}^{N_k \times N_k}$, which is a finite-dimensional approximation of the discrete-time Koopman operator $\hat{\mathcal{K}}$, for a system from sampled data of the state variables of the system Williams et al. (2015). From $\hat{K}$, one can also obtain the Koopman eigenfunctions and Koopman eigenvalues.

In the data collection stage, the system described by (1), where f is unknown, is discretized as (6) and sampled in time Δt to create a data set of M 'snapshot pairs' (x_i, y_i), where $x_i = x(i\Delta t)$, $y_i = x((i+1)\Delta t) = F(x_i)$ and $i = 0, \ldots, M-1$. Let the data matrices be $X = [x_0, x_1, \ldots, x_{M-1}]$ and $Y = [y_0, y_1, \ldots, y_{M-1}]$, where $X, Y \in \mathbb{R}^{n \times M}$. In addition to collecting the data, we create a "dictionary" of observable functions, $\Psi(x) = [\psi_1(x), \psi_2(x), \ldots, \psi_{N_k}(x)]^T$, which will be used in EDMD. A further discussion of the choice of these observable functions is presented in Section 2.3.

Now consider an observable function $\psi(x) \in \mathcal{F}$ that can also be written as $\psi(x) = a^T\Psi(x)$, where a is a weight vector and $\Psi(x)$ is our dictionary of observable functions. From (7), we have

$$\hat{\mathcal{K}}\psi = \psi \circ F = a^T(\Psi \circ F)(x) = a^T(\hat{K}\Psi(x)) + r(x), \tag{9}$$

where $r(x)$ is a residual term. It can be shown that, see Appendix A, the best estimate of $\hat{K}$ takes the form

$$\hat{K} = G_1 G_0^+, \tag{10}$$

where

$$G_0 = \sum_{m=0}^{M-1} \Psi(x_m)\Psi(x_m)^T,\ G_1 = \sum_{m=0}^{M-1} \Psi(y_m)\Psi(x_m)^T$$

and G_0^+ is the pseudoinverse of G_0.

For each discrete-time eigenvalue μ_i in U, where U is the diagonal matrix of eigenvalues of $\hat{K}$, we define a diagonal matrix Λ composed of $\lambda_i = \frac{\ln(\mu_i)}{\Delta t}$ (defined in Section 2.1). This allows for

$$\dot{\Phi}(x) = \Lambda\Phi(x), \tag{11}$$

which is the matrix version of (5). Essentially, (11) serves as a linear approximate model of the nonlinear system (1).

2.3 Selection of Observable Functions

The EDMD method is about finding a finite-dimensional Koopman operator approximation that can accurately reflect the dynamics of the original nonlinear system. As one may infer from Section 2.2, the accuracy of performance of EDMD can vary heavily depending on the selection of observable functions $\Psi(x)$, given that those are used to determine $\hat{K}$. Essentially, selecting the best observable functions to optimize EDMD accuracy is an art form in itself. There have been many studies that have looked into selecting the proper observable functions. For example, the work in Mauroy and Mezić (2016) looked at using the Taylor series expansion of the system. Another work Netto et al. (2021) showed that the optimal observable functions are the independent nonlinear terms in the system equation and additionally the Lie derivatives of those functions. All of these previous methods have found ideal observable functions using knowledge of the system. However, through this work, we aim to solve the EDMD approximation through completely data-driven methods (without knowledge of the original system). We leave the problem of finding ideal observable functions through data-driven methods to a future study.

In this work, we choose the observable functions $\Psi(x)$ to be polynomial combinations of the state x. $\Psi(x)$ contains all possible polynomial terms for a set number of dimensions n and up to an order p. For example, if $n = 1$ and $p = 3$, $\Psi(x) = [x, x^2, x^3]^T$. If $n = 2$ and $p = 2$, $\Psi(x) = [x_1, x_2, x_1^2, x_1x_2, x_2^2]^T$. We have found through testing EDMD on numerous systems that simply having polynomials to a certain order accurately reflects the dynamics of many systems, even when the original system does not contain polynomial terms. Furthermore, as will be shown in Section 3, having each state as a separate observable function helps for the purposes of this paper.

2.4 Lyapunov Stability and Region of Attraction

There have been many proposed ways of determining ROAs, but maybe the most common one is using Lyapunov functions Khalil (2002).

Definition: For system (1), a Lyapunov Function, $V(x)$ is a scalar function of the states of the system. In a region $D \in \mathbb{R}^n$, if one can construct a Lyapunov function such that $V(0) = 0$, $V(x) > 0$ for all $x \neq 0, x \in D$, and $\dot{V}(x) < 0$ for all $x \neq 0, x \in D$, then there exists a region $\bar{\Omega}_{ROA}$ defined by

$$\bar{\Omega}_{ROA} = \{V(x) < c\}, \tag{12}$$

where c is a constant such that $\bar{\Omega}_{ROA} \subset D$. All the points that lie inside $\bar{\Omega}_{ROA}$ are guaranteed to decrease and converge to the origin. The region $\bar{\Omega}_{ROA}$ is a conservative estimate for the ROA Khalil (2002).

Consider system (1), which has a Jacobian matrix $A = \frac{\partial f}{\partial x}$. If A is Hurwitz, then there exists a positive definite symmetric matrix P that is the solution to the equation

$$PA + A^TP = -Q, \tag{13}$$

where Q is a positive definite matrix. We can then use this function to construct a Lyapunov Function candidate, V,

$$V(x) = x^TPx, \tag{14}$$

which will be positive definite for all x. Additionally $V(0) = 0$. Since the first two conditions are satisfied, the only thing to do is find the region D in which $\dot{V}(x) < 0$. One simple analytical approach to finding D is by setting a boundary on x to be $||x|| < r$. If $\dot{V}(x) < 0$ for $||x|| < r$, then the region D can be defined as $D = \{||x|| < r\}$. A conservative estimate for c is then $c = \lambda_{min}(P)r^2$, where P comes from (13). Now we have a conservative estimate for the region of attraction (12).

3. MAIN RESULTS

3.1 Estimating the Original System Using EDMD

Although the primary purpose of EDMD is to find the Koopman operator, eigenfunctions, and eigenvalues, one can also estimate the function f from (1) as $\bar{f}$ from the output of EDMD. This will be crucial to our proposed method.

First, consider a linear system in a discrete form $x_{k+1} = \hat{A}x_k$, with sample time Δt. Then we can approximate $\dot{x}$ as

$$\dot{x} = \frac{dx}{dt} \approx \frac{\Delta x}{\Delta t} = \frac{x_{k+1} - x_k}{\Delta t} = \frac{\hat{A}x_k - x_k}{\Delta t}$$

and this leads to $\dot{x} = A_t x$, where

$$A_t \approx \frac{1}{\Delta t}(\hat{A} - I). \tag{15}$$

Now, apply this transformation to $\hat{K}$. Through EDMD, we obtain $\hat{K}$, which represents the evolution of the observable functions, $\Psi(x)$, in discrete time in the form

$$\Psi_{k+1}(x) = \hat{K}\Psi_k(x). \tag{16}$$

Applying (15) to (16), we can take the continuous-time approximation for the Koopman operator to be $K_t = \frac{1}{\Delta t}(\hat{K} - I)$, where Δt is the same Δt that was used to sample the system to gather the data for EDMD, and K_t, I are of size $N_k \times N_k$. This leads to having the continuous time version of (16) as

$$\dot{\Psi}(x) = K_t\Psi(x). \tag{17}$$

Recall from the previous section that we define the dictionary of observable functions to be all possible polynomial combinations of n states up to some order p. This means that the first n observable functions will be just the states themselves. Therefore, the first n rows of (17) will be dynamic equations of the states. This will give an approximation for $f(x)$.

For example, if we define our dictionary of observables for a system as $n = 2$ and $p = 2$, then (17) looks like

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ (\dot{x_1^2}) \\ (\dot{x_1x_2}) \\ (\dot{x_2^2}) \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix}$$

The first two rows of this matrix equation are dynamic equations for the states of the system, x_1 and x_2. Therefore we can use these as the approximation $\bar{f}(x)$ of the vector-valued function $f(x)$, that is

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{bmatrix} = \bar{f}(x).$$

3.2 Data-Driven Enlargement of the estimate of the ROA

Recall from Section 2.4 that the Lyapunov function V is found by solving for P in (13), where the Jacobian A is the linear approximation of (1). In this section, instead of using the Jacobian A, we will use the eigenvalue matrix, Λ, from the eigenfunction evolution equation (11) determined from EDMD. This promises a more accurate representation of the system (1). In a similar way to the Jacobian method, the system can be shown to admit a Lyapunov function of the same form Mauroy et al. (2020). Accordingly, we solve the Lyapunov equation

$$P_\Lambda \Lambda + \Lambda^T P_\Lambda = -Q. \tag{18}$$

Here, we will set $Q = I$, where Q, P_Λ are size $N_k \times N_k$. Then we define the Lyapunov function candidate in a similar way to (14) such that

$$V(x) = \Phi(x)^T P_\Lambda \Phi(x) \tag{19}$$

Differentiating $V(x)$ leads to

$$\dot{V}(x) = \frac{dV}{dx}\frac{dx}{dt} = \frac{dV}{dx}\dot{x} \tag{20}$$

Since we know V, we can easily differentiate with respect to x to find $\frac{dV}{dx}$. We do not have exact values for $\dot{x}$, but we do have the approximation of Section 3.1, $\dot{x} = \bar{f}(x)$, making (20)

$$\dot{V}(x) = \frac{dV}{dx}\bar{f}(x). \tag{21}$$

We next propose a data-driven algorithm to maximize c over the analytical method described in 2.4: To find a larger estimate for the ROA, start with a very small c, and start to increase it to expand the region $\bar{\Omega}_{ROA} = \{V(x) < c\}$. Do this until the boundary crosses into the region where $\dot{V}(x) > 0$.

By finding a Lyapunov function based on the results from EDMD, along with using the method that maximizes c, we usually find a larger, more accurate approximation for the conservative ROA compared to the method presented in Section 2.4. Not only is this significant given that it is found through only data, but larger ROA results in less computational effort for the backward trajectory method presented in the next section.

3.3 Reversing Trajectories to Obtain a Larger ROA

This method follows closely the one presented in Genesio and Vicino (1984) and adapted in Kant et al. (2017). All points inside of the complete ROA, Ω_{Real}, will have trajectories moving inwards towards the origin and all points outside of the ROA will have trajectories moving outwards. These are the natural trajectories from the forward-in-time system (1). Now consider the system

$$\dot{x} = -f(x). \tag{22}$$

This system has the same magnitudes as (1), as the function $f(x)$ did not change, but because of the minus sign now every trajectory is reversed. Points that are near the origin and within the ROA will move outward towards the boundary of the ROA while points outside of the ROA will move inwards. Therefore, the way to enlarge the conservative ROA approximation is to gather all of the points on the boundary of $\bar{\Omega}_{ROA}$, as found in the previous section, and perform backward integration with a discretized version of (22), with sample time Δt_1. To keep this as a data-driven algorithm, (22) is approximated using the methods of Section 3.1, to give us

$$\dot{x} = -\bar{f}(x) \tag{23}$$

Now this is discretized as

$$x_{k+1} = x_k - \bar{f}(x_k)\Delta t_1 \tag{24}$$

for some sample time Δt_1. To perform backward integration, we must start with a set of points, S_0, that are on the boundary of $\bar{\Omega}_{ROA}$. Every point on S_0 is plugged into (24), to go one step backward to make the set S_1.

After each movement, the points on S_i, where $i = 0, 1, \ldots$ must be checked for two conditions:
1) That all points are within a certain distance metric, ϵ, from each other. This is to maintain the idea that the set of points is one closed, continuous boundary.
2) That none of the points in the set S_{i+j}, where j is a predetermined integer, are inside of the set S_i. This is done because assuming that the boundary of $\bar{\Omega}_{ROA}$ is inside the real ROA, Ω_{Real}, the backward trajectories of all points along the boundary of $\bar{\Omega}_{ROA}$ will move outwards, resulting in a larger ROA estimation. However, if the approximation has moved outside of Ω_{Real}, then the next iteration will be smaller than the previous ROA approximation. This is because the reverse trajectories of the points outside of Ω_{Real} will move inwards. Therefore if a set of future points S_{i+j} is contained in S_i, it is likely because the ROA approximation went too far outwards. This process is iterated until one of the above conditions is not met. Next, we discuss using the concept of the convex hull to extend the applicability of the method to systems with dimensions $n > 3$.

3.4 Convex Hulls

An extension of the reverse trajectory method to systems of dimension $n > 3$ can be accomplished through the use of convex hulls Kant et al. (2017).
Definition: Kant et al. (2017) Given a set of points, S, that forms some region, a convex hull is the set of all vertices in S that make up the boundary of this region. Convex hulls can be found computationally using the Matlab function *convhulln*.

In the trajectory reversing method, it is necessary to work with regions as we are aiming to find the ROA. Because regions are typically defined in physical space, the trajectory reversing method - or any ROA approximation method - becomes computationally difficult with $n > 3$. However, with convex hulls, one can define a region using the points along the boundary of the region, and although

we cannot visualize regions in multidimensional space, we can know what points lie in or out of those regions by using the Matlab function *inhull* John D'Errico (2022), which returns whether or not an $n-$dimensional point lies within the $n-$dimensional convex hull.

The proposed procedure is shown in Algorithm 1.

Algorithm 1: Data-driven algorithm for estimating ROA

Initialization: Gather system data in the form of X and Y. Define a dictionary of observable functions, $\Psi(x)$, that are polynomial terms for n states up to order p.
STEP 1: Perform EDMD using X, Y, and $\Psi(x)$.
STEP 2: Using the matrix $\hat{K}$, the output from EDMD, and $\Psi(x)$, find $\Phi(x)$, Λ, and $\bar{f}(x)$, from (A.4) and (11), and Section 3.1, respectively.
STEP 3: Use Λ to solve for P_Λ in (18). From this, find the Lyapunov function $V(x)$ in (19). Use $V(x)$ and $\bar{f}(x)$ to find $\dot{V}(x)$ in (21).
STEP 4: Maximize c. Expand the boundary of the ROA $\Omega_c = \{V(x) = c)\}$ until a point along the boundary is in the region where $\dot{V} > 0$.
numPositive counts how many values in a set are positive.
Let $q = 0$, $c =$ some small number < 1
while $q = 0$ **do**
  increment c
  $\Omega_c = \{V(x) = c\}$
  $q = numPositive(\dot{V}(\Omega_c))$
end
Ω_c is the boundary of the conservative ROA
STEP 5: Select a time step, Δt_1, and find the equation of the backward trajectory (24) using Δt_1 and $\bar{f}(x)$
STEP 6: *maxdist(S)* finds the maximum distance between two consecutive points in S. k_1 and k_2 are how many iterations the enlargement algorithm will take.
Let $l = 0$, $k_1 < k_2$, $S_0 = \Omega_c$.
for $i = 1$ *to* k_1 **do**
  if $l = i - 1$ **then**
    for $j = 1$ *to* k_2 **do**
      if S_{i+j} *not in* S_i *and* $maxdist(S_i) < \epsilon$ **then**
        $l = i$
        Break (back to initial for loop)
      end
    end
  end
end
STEP 7: If $l = 0$, repeat STEPS 6 to 7 with smaller Δt_1. If $l = k_1$, repeat STEPS 6 to 7 with a larger value for k_1 and k_2.
RESULT $\bar{\Omega}_{ROA} = convhull(S_l)$ is the enlarged ROA approximation for (1), found without using the system equations.

4. NUMERICAL SIMULATION

To show how effective this method is, we present the Van der Pol Oscillator, also used in Kant et al. (2017), whose dynamics are

$$\dot{x}_1 = -x_2, \quad \dot{x}_2 = x_1 + (x_1^2 - 1)x_2.$$

The system is asymptotically stable at the origin. To gather data, the system is sampled at time $\Delta t = 0.0001$ and converted into the X and Y matrices of the data. EDMD is then performed, with the dictionary of observables being the set of polynomial combinations with $n = 2$ and $p = 4$.

Following the results of EDMD, an equation for $V(x)$ and $\bar{f}$ are obtained, leading to a conservative estimate for $\bar{\Omega}_{ROA} = \{V(x) < c\}$. This data-driven conservative ROA can be seen in Fig. 1, which compared to the analytical approach using the Jacobian Khalil (2002), is a larger and more accurate approximation.

From the conservative estimate, we begin the reverse trajectory method. We select $\Delta t_1 = 0.005$ and $\epsilon = 0.1$. After the algorithm completes, we find the best approximation was found with $k_1 = 400$, $k_2 = 405$, and $l = 396$. The results are shown in Fig. 2, where it is clear that $\text{Conv}(S_l)$ is a very good approximation for the ROA of this system.

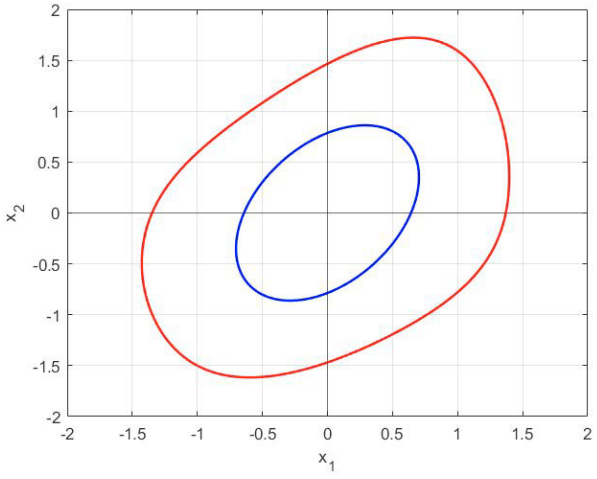

Fig. 1. The red line shows the conservative ROA approximation using the data-driven technique. The blue line shows the conservative ROA approximation using the Jacobian method.

5. CONCLUSION

We developed a data-driven approach to constructing a conservative estimate for the ROA of nonlinear systems using the Koopman operator and Lyapunov functions. We then enlarged the conservative estimate to a more accurate ROA approximation through trajectory reversing. All of this was completed without knowledge of the original system, and all approximations were made by using EDMD. By using convex hulls, we can define regions in multidimensional space without the need to physically visualize them. To improve upon this work, more research should be done on how to select the dictionary of observables, through data-driven techniques, that can optimize the performance of EDMD and ultimately improve the accuracy of each ROA approximation.

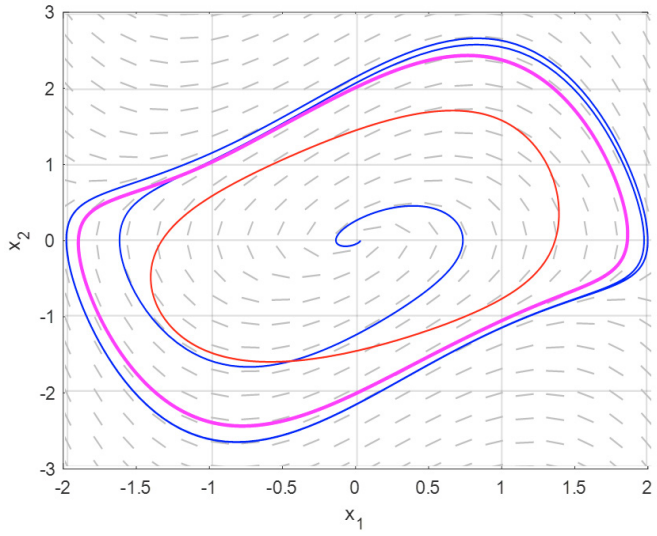

Fig. 2. The outermost blue line is the real ROA of the system, found by using the Matlab function *pplane9* Gerardo Garcia (2022). The red line is the initial conservative ROA approximation, and the magenta line is the ROA approximation after using the backward trajectory method.

REFERENCES

Abolmasoumi, A.H., Netto, M., and Mili, L. (2022). Robust dynamic mode decomposition. *IEEE Access*, 10, 65473–65484.

Anghel, M., Milano, F., and Papachristodoulou, A. (2013). Algorithmic construction of lyapunov functions for power system stability analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(9), 2533–2546.

Chen, S., Fazlyab, M., Morari, M., Pappas, G.J., and Preciado, V.M. (2021). Learning region of attraction for nonlinear systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 6477–6484. IEEE.

Choi, H. and Bose, S. (2018). Transient stability analysis of power systems using koopman operators. *[Preprint]*.

Garcia-Tenorio, C., Tellez-Castro, D., Mojica-Nava, E., and Vande Wouwer, A. (2023). Evaluation of the regions of attraction of higher-dimensional hyperbolic systems using extended dynamic mode decomposition. *Automation*, 4(1), 57–77.

Genesio, R. and Vicino, A. (1984). New techniques for constructing asymptotic stability regions for nonlinear systems. *IEEE transactions on circuits and systems*, 31(6), 574–581.

Gerardo Garcia (2022). www.mathworks.com/matlabcentral.

Hemati, M.S., Rowley, C.W., Deem, E.A., and Cattafesta, L.N. (2017). De-biasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets. *Theoretical and Computational Fluid Dynamics*, 31(4), 349–368.

John D'Errico (2022). www.mathworks.com/matlabcentral.

Kant, N., Chowdhury, D., Mukherjee, R., and Khalil, H.K. (2017). An algorithm for enlarging the region of attraction using trajectory reversing. In *2017 American Control Conference (ACC)*, 4171–4176. IEEE.

Khalil, H.K. (2002). Nonlinear systems third edition. *Patience Hall*, 115.

Kutz, J.N., Brunton, S.L., Brunton, B.W., and Proctor, J.L. (2016). *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM.

Mauroy, A. and Mezić, I. (2016). Global stability analysis using the eigenfunctions of the koopman operator. *IEEE Transactions on Automatic Control*, 61(11), 3356–3369.

Mauroy, A., Susuki, Y., and Mezić, I. (2020). *Koopman operator in systems and control*. Springer.

Meng, Y., Zhou, R., and Liu, J. (2023). Learning regions of attraction in unknown dynamical systems via zubov-koopman lifting: Regularities and convergence. *arXiv preprint arXiv:2311.15119*.

Netto, M., Susuki, Y., Krishnan, V., and Zhang, Y. (2021). On analytical construction of observable functions in extended dynamic mode decomposition for nonlinear estimation and prediction. In *2021 American Control Conference (ACC)*, 4190–4195. IEEE.

Schmid, P.J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656, 5–28.

Shen, Y., Bichuch, M., and Mallada, E. (2022). Model-free learning of regions of attraction via recurrent sets. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 4714–4719. IEEE.

Wang, K., Menon, P.P., Veenman, J., and Bennani, S. (2023). Estimation of region of attraction with gaussian process classification. *European Journal of Control*, 100856.

Williams, M.O., Kevrekidis, I.G., and Rowley, C.W. (2015). A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6), 1307–1346.

Appendix A. EDMD ANALYSIS

We seek the best approximation for $\hat{K}$. For this purpose, we seek to minimize the least squares function

$$J = \sum_{m=0}^{M-1} |a^T(\Psi(y_m) - \hat{K}\Psi(x_m))|^2 \tag{A.1}$$

The value of $\hat{K}$ that minimizes J is $\hat{K} = G_1 G_0^+$, where

$$G_0 = \sum_{m=0}^{M-1} \Psi(x_m)\Psi(x_m)^T,\ G_1 = \sum_{m=0}^{M-1} \Psi(y_m)\Psi(x_m)^T$$

and G_0^+ is the pseudoinverse of G_0. Note that $\hat{K}$, G_0, $G_1 \in \mathbb{C}^{N_k \times N_k}$. Now diagonalize $\hat{K}$

$$\hat{K} = VUV^{-1} \tag{A.2}$$

where V is the matrix of right eigenvectors and U is the diagonal matrix of eigenvalues. Then from (7) and (A.2), it can be shown that

$$UV^{-1}\Psi(x) = V^{-1}\Psi(x) \circ F. \tag{A.3}$$

Let

$$\Phi(x) = V^{-1}\Psi(x). \tag{A.4}$$

Then (A.3) becomes $U\Phi(x) = \Phi(x) \circ F = \hat{K}\Phi(x)$, which satisfies the definition of Koopman eigenfunctions and discrete-time Koopman eigenvalues from (8), with $\Phi(x)$ being the vector of Koopman eigenfunctions and U being the diagonal matrix of Koopman Eigenvalues.