Learning-based Design of Luenberger Observers for Autonomous Nonlinear Systems

Muhammad Umar B. Niazi*,† John Cao* Xudong Sun* Amritam Das* Karl Henrik Johansson*

***Abstract*—Designing Luenberger observers for nonlinear systems involves the challenging task of transforming the state to an alternate coordinate system, possibly of higher dimensions, where the system is asymptotically stable and linear up to output injection. The observer then estimates the system's state in the original coordinates by inverting the transformation map. However, finding a suitable injective transformation whose inverse can be derived remains a primary challenge for general nonlinear systems. We propose a novel approach that uses supervised physics-informed neural networks to approximate both the transformation and its inverse. Our method exhibits superior generalization capabilities to contemporary methods and demonstrates robustness to both neural network's approximation errors and system uncertainties.**

***Index Terms*—Nonlinear observer design, robust estimation, physics-informed learning, empirical generalization error.**

I. Introduction

Nonlinear Luenberger observers, also known as Kazantzis-Kravaris/Luenberger (KKL) observers, generalize the theory of Luenberger observers [1] to nonlinear systems. The main idea of KKL observers is to find an injective map that satisfies a certain partial differential equation (PDE) and transforms a nonlinear system to another coordinate system, possibly of higher dimensions than the original state space. The dynamics of the transformed system are required to be stable and linear up to output injection. Then, the KKL observer is a copy of the transformed system and estimates the state of the original system by inverting the transformation map.

Initially proposed by [2] and [3], the theory of KKL observers was subsequently rediscovered by Kazantzis & Kravaris [4], who provided local guarantees around an equilibrium point via Lyapunov's Auxiliary Theorem. Although [5] relaxed the restrictive assumptions of [4] to some extent, the analysis remained local until [6] proposed the first global result under the assumption of the so-called finite complexity, which also turned out to be quite restrictive for general nonlinear systems. In this regard, a complete and most general treatment of the problem was presented by Andrieu & Praly [7], who introduced the notion of *backward distinguishability* ensuring the existence of an injective transformation required by the KKL observers. Later, under some additional observability conditions, [8] proved that KKL observers converge exponentially and are also tunable. The theory is also extended to non-autonomous and controlled nonlinear systems in [9]–[11].

The main challenge in the design of KKL observers is to not only find the transformation map but also its left inverse, and both problems turn out to be very difficult in practice; see [12] and [13]. To this end, [14]–[16] have proposed several methods to approximate the transformation map and its inverse via feedforward neural networks. By fixing the dynamics of the KKL observer, they propose to generate synthetic data trajectories by numerically solving both the system's model and the KKL observer, where both are initialized at multiple points in their corresponding state spaces. Then, using a supervised learning approach, a neural network is trained to approximate the transformation map and its left inverse. Moreover, [16] also proposed an unsupervised learning approach by assuming an autoencoder-type architecture and adding the PDE associated with the transformation map as a design constraint. However, both approaches suffer from overfitting on the training samples and do not generalize well in practice.

In this paper, we propose a *supervised physics-informed learning* approach to approximate the transformation map and its left inverse. Such an approach incorporates the physical knowledge described by the PDE constraint, which is directly integrated with the conventional supervised learning [17], [18]. Embedding the physical knowledge of systems by adding the PDE constraint as a physically relevant invariant improves the accuracy, generalization, and training time of the learning method. In this way, we improve upon the idea of [14]–[16] by avoiding overfitting and obtaining better generalization to the whole state space.

The main contribution of this paper includes a complete learning method of the KKL observer design via a supervised physics-informed neural network (PINN). We show that the KKL observer is robust to not only the neural network's approximation error but also to model and sensor uncertainties. The robustness is quantified in terms of input-to-state stability [19] of the state estimation error. We define an empirical metric to quantify the generalization capability of the learned KKL observer and provide a detailed discussion on why our method exhibits better generalization capabilities than the supervised neural network (NN) approach of [14]–[16] and

$*$ Division of Decision and Control Systems and Digital Futures, EECS, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden.

$\dagger$ Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA. Corresponding author's email: niazi@mit.edu

This work is supported by the Swedish Research Council and the Knut and Alice Wallenberg Foundation, Sweden. It also received funding from European Union's Horizon Research and Innovation Programme under Marie Skłodowska-Curie grant agreement No. 101062523.

the unsupervised autoencoder (AE) approach of [16]. Finally, we demonstrate the dominance of our method over these approaches through statistically well-designed experiments.

After summarizing a general idea of KKL observers in Section II, we state the problem addressed in this paper in Section III. The learning method of KKL observers is presented in Section IV. Section V evaluates the performance of the observer under approximation errors and uncertainties, and defines and discusses an empirical metric to assess the generalization capability of the learned observer. Finally, Section VI presents the experimental results and Section VII ends with concluding remarks and the future outlook.

Notations. For a vector $x \in \mathbb{R}^n$, the Euclidean norm $\|x\| = \sqrt{x^{\mathrm{T}}x}$ and the maximum norm $\|x\|_\infty = \max_i |x_i|$. For a measurable essentially bounded function $w \in L^\infty(\mathbb{R};\mathbb{R}^n)$, the essential supremum norm $\|w\|_{L^\infty} = \operatorname{ess\,sup}_{t\in\mathbb{R}} \|w(t)\| \doteq \inf\{c \geq 0 : \|w(t)\|_\infty \leq c$ for almost every $t \in \mathbb{R}\}$. For a matrix $M \in \mathbb{R}^{n\times m}$, $\|M\| = \sup_{\|x\|=1} \|Mx\|$ denotes the induced norm, which is equal to the maximum singular value $\sigma_{\max}(M)$. The spectrum of $M \in \mathbb{R}^{n\times n}$ is denoted by $\operatorname{eig}(M)$, and $\lambda_{\min}(M) = \min_{\lambda\in\operatorname{eig}(M)} |\mathrm{Re}(\lambda)|$ and $\lambda_{\max}(M) = \max_{\lambda\in\operatorname{eig}(M)} |\mathrm{Re}(\lambda)|$. The condition number of M is denoted by $\operatorname{cond}(M)$.

II. Preliminaries on KKL Observers

In this section, we briefly summarize the theory of KKL observers. For more details, see [4], [7], [8].

Consider a nonlinear system

$$\dot{x} = f(x); \quad y = h(x) \tag{1}$$

where $x(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}$ is the state with $x(0) = x_0 \in \mathcal{X}$ the initial condition, $y(t) \in \mathbb{R}^{n_y}$ is the measured output, and the maps $f : \mathcal{X} \to \mathbb{R}^{n_x}$ and $h : \mathcal{X} \to \mathbb{R}^{n_y}$ are smooth.

The design method of a KKL observer is as follows:

1) Find an injective[1] map $\mathcal{T} : \mathcal{X} \to \mathbb{R}^{n_z}$ that transforms (1) to new coordinates $z = \mathcal{T}(x)$, where

$$\dot{z} = Az + Bh(x); \quad z(0) = \mathcal{T}(x_0) \tag{2}$$

with $A \in \mathbb{R}^{n_z\times n_z}$ a Hurwitz matrix and $B \in \mathbb{R}^{n_z\times n_y}$ such that (A, B) is controllable. From (2), it follows that $\mathcal{T}$ must be a solution to the following PDE:

$$\frac{\partial \mathcal{T}}{\partial x}(x)f(x) = A\mathcal{T}(x) + Bh(x); \quad \mathcal{T}(0) = 0. \tag{3}$$

2) Since $\mathcal{T}$ is injective, its left inverse $\mathcal{T}^*$ exists, i.e., $\mathcal{T}^*(\mathcal{T}(x)) = x$. The KKL observer is then given by

$$\begin{aligned} \dot{\hat{z}} &= A\hat{z} + By; \quad \hat{z}(0) = \hat{z}_0 \\ \hat{x} &= \mathcal{T}^*(\hat{z}). \end{aligned} \tag{4}$$

There are certain conditions that system (1) needs to satisfy in order to ensure the existence of a KKL observer (4) in a sense that $\lim_{t\to\infty} \|\hat{x}(t) - x(t)\| = 0$. Let $x(t; x_0)$ denote the state trajectory of (1) with $x(0) = x_0$. Then, (1) is said to be *forward complete within* $\mathcal{X}$ if for every $x_0 \in \mathcal{X}$, $x(t; x_0) \in \mathcal{X}$ is well-defined for every $t \in \mathbb{R}_{\geq 0}$.

Assumption 1. *There exists a compact set $\mathcal{X} \subset \mathbb{R}^{n_x}$ such that the system* (1) *is forward complete within $\mathcal{X}$.*

A map $\mathcal{T} : \mathcal{X} \to \mathbb{R}^{n_z}$ is said to be *uniformly injective* if there exists a class $\mathcal{K}$ function[2] ρ such that, for every $x_1, x_2 \in \mathcal{X}$, $\|x_1 - x_2\| \leq \rho(\|\mathcal{T}(x_1) - \mathcal{T}(x_2)\|)$.

For the existence of a KKL observer (4), it is sufficient that (1) is forward complete and the map $\mathcal{T}$ satisfying (3) is uniformly injective, see [7, Theorem 1]. Since A is a Hurwitz matrix, $\|\hat{z}(t) - z(t)\| = \|\mathcal{T}(\hat{x}(t)) - \mathcal{T}(x(t))\|$ converges to zero exponentially. Thus, the uniform injectivity

$$\|\hat{x}(t) - x(t)\| \leq \rho(\|\mathcal{T}(\hat{x}(t)) - \mathcal{T}(x(t))\|) \tag{5}$$

implies that $\|\hat{x}(t) - x(t)\|$ also converges to zero. However, only asymptotic convergence of the estimation error can be guaranteed because the inverse $\mathcal{T}^*$ is a nonlinear map, which may destroy the exponentiality of the convergence.

Given an open set $\mathcal{O} \supset \mathcal{X}$, the system (1) is said to be *backward $\mathcal{O}$-distinguishable* on $\mathcal{X}$ if for every pair of distinct initial conditions $x_0^1, x_0^2 \in \mathcal{X}$, there exists $\tau < 0$ such that $x(t; x_0^1), x(t; x_0^2) \in \mathcal{O}$ are well-defined for $t \in [\tau, 0]$, and

$$h(x(\tau; x_0^1)) \neq h(x(\tau; x_0^2)).$$

In other words, this means that there exists a finite negative time such that the output maps, corresponding to different trajectories initialized in $\mathcal{X}$, can be distinguished before any of the trajectories leaves $\mathcal{O}$ in backward time.

Assumption 2. *There exists an open bounded set $\mathcal{O} \supset \mathcal{X}$ such that* (1) *is backward $\mathcal{O}$-distinguishable on $\mathcal{X}$.*

It turns out that Assumptions 1 and 2 are sufficient for the existence of an injective map $\mathcal{T}$ satisfying (3). This result is obtained in [7], [13], [20], which can be restated as follows:

Theorem 1. *Let Assumptions 1 and 2 hold. Then, for any controllable $(A, B) \in (\mathbb{R}^{n_z\times n_z}, \mathbb{R}^{n_z\times n_y})\setminus\mathcal{J}$ such that $n_z = n_y(2n_x + 1)$, $A + \delta I_{n_z}$ is Hurwitz for some $\delta > 0$, and $\mathcal{J} \subset (\mathbb{R}^{n_z\times n_z}, \mathbb{R}^{n_z\times n_y})$ is a set of zero Lebesgue measure, there exists a uniformly injective map $\mathcal{T} : \mathcal{X} \to \mathbb{R}^{n_z}$ satisfying* (3).

By relying on Theorem 1, we propose a learning method for $\mathcal{T}$ and $\mathcal{T}^*$ under the constraint that $\mathcal{T}$ satisfies (3). In what follows, we choose and fix $A \in \mathbb{R}^{n_z\times n_z}$ and $B \in \mathbb{R}^{n_z\times n_y}$ such that A is Huwitz and (A, B) is controllable, where $n_z = n_y(2n_x + 1)$.

III. Problem Statement

We aim to design a KKL observer for (1) that estimates the state $x(t)$ by using the knowledge of the system's output y and its model $f(\cdot)$ and $h(\cdot)$. That is, the observer (4) ensures $\lim_{t\to\infty} \|\hat{x}(t) - x(t)\| = 0$ when $\mathcal{T}$ and $\mathcal{T}^*$ are known. In

[1] A map $\mathcal{T} : \mathcal{X} \to \mathbb{R}^{n_z}$ is said to be *injective* if for every $x_1, x_2 \in \mathcal{X}$, $\mathcal{T}(x_1) = \mathcal{T}(x_2)$ implies $x_1 = x_2$.

[2] A function $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}$ if it is continuous, zero at zero, and strictly increasing.

case, $\mathcal{T}$ and $\mathcal{T}^*$ are respectively approximated by $\hat{\mathcal{T}}$ and $\hat{\mathcal{T}}^*$, then the asymptotic estimation error satisfies

$$\limsup_{t\to\infty} \|\hat{x}(t) - x(t)\| \leq \epsilon$$

where $\epsilon > 0$ depends on the approximation error.

The problem can be divided into two parts:

1) Learn the map $\mathcal{T}$ satisfying the PDE (3) and its left inverse $\mathcal{T}^*$.
2) Evaluate the performance of the KKL observer in terms of its robustness to the approximation error, model uncertainties, and measurement noise, and its generalization capability when $\mathcal{T}$ and $\mathcal{T}^*$ are learned on a discrete subset of $\mathcal{X}$.

IV. Learning the Transformation Map and its Left Inverse

A critical step of KKL observer design is to find the injective map $\mathcal{T} : \mathcal{X} \to \mathbb{R}^{n_z}$ satisfying the PDE (3), so that (1) admits a linear representation (2), and its left inverse $\mathcal{T}^*$, so that a state estimate can be obtained in the original state space coordinates. This amounts to solving the PDE (3) for $\mathcal{T}$, whose solution is obtained in [7] as

$$\mathcal{T}(x) = \int_{-\infty}^{0} \exp(A\tau) B h(\breve{x}(\tau; x)) \mathrm{d}\tau \tag{6}$$

where $\breve{x}(\tau; x) \in \mathcal{X}$ is the backward solution initialized at $x \in \mathcal{X}$, for $\tau \leq 0$, to the modified dynamics $\dot{\breve{x}}(\tau) = g(\breve{x}(\tau))$ with $g(\breve{x}(\tau)) = f(\breve{x}(\tau))$ if $\breve{x}(\tau) \in \mathcal{X}$ and $g(\breve{x}(\tau)) = 0$ otherwise. However, there are two issues with this solution:

- It is practically impossible to obtain a backward output map $h(\breve{x}(\tau; x))$ for $\tau < 0$ and then compute the integral (6) for every initial point $x \in \mathcal{X}$; [13].
- Even if $\mathcal{T}$ is known in any other form[3] than (6), finding the left inverse $\mathcal{T}^*$ is very difficult both analytically and numerically; [12].

To circumvent these challenges, it is reasonable to approximate these maps using neural networks.

Let $\hat{\mathcal{T}}_\theta$ and $\hat{\mathcal{T}}_\eta^*$ be the parametrized neural networks that approximate $\mathcal{T}$ and $\mathcal{T}^*$, respectively. Here, θ, η are vectors containing all the weights and biases of each neural network, respectively, and can be considered as learning parameters for the nonlinear regression problem. In the following subsections, we describe our method, illustrated in Figure 1, for learning $\mathcal{T}$ and $\mathcal{T}^*$ through neural networks $\hat{\mathcal{T}}_\theta$ and $\hat{\mathcal{T}}_\eta^*$.

A. Generating Data for Training

Since the system trajectories for arbitrary initial conditions can be obtained numerically by solving the nonlinear system (1) for x and the linear system (2) for z, one can pose the problem of learning θ and η as a nonlinear regression over the simulated data trajectories on a finite time horizon $T > 0$. The steps to generate these trajectories are described below:

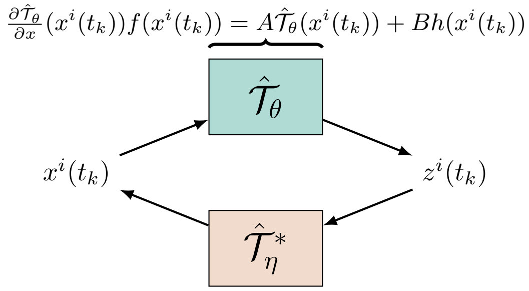

$\underbrace{\frac{\partial \hat{\mathcal{T}}_\theta}{\partial x}(x^i(t_k)) f(x^i(t_k)) = A\hat{\mathcal{T}}_\theta(x^i(t_k))} + Bh(x^i(t_k))$
$\hat{\mathcal{T}}_\theta$
$x^i(t_k)$
$z^i(t_k)$
$\hat{\mathcal{T}}_\eta^*$

Fig. 1: Architecture for learning the transformation $\mathcal{T}$ and its inverse $\mathcal{T}^*$ using neural networks with parameters θ and η.

1) Define a set $\mathcal{X}^{\text{train}} \times \mathcal{Z}^{\text{train}} \subset \mathcal{X} \times \mathcal{Z}$ from which the initial conditions are chosen for training, where $\mathcal{Z} \subset \mathbb{R}^{n_z}$. For some $p \in \mathbb{N}$, choose a set of initial conditions
$$(x_0^1, z_0^1), \ldots, (x_0^p, z_0^p) \in \mathcal{X}^{\text{train}} \times \mathcal{Z}^{\text{train}}.$$
2) Simulate (1) and (2) with these initial conditions and generate sampled trajectories from $t_0 = 0$ to $t_{\tau-1} = T$
$$x^i(t_k) \doteq x(t_k; x_0^i) \text{ and } z^i(t_k) \doteq z(t_k; z_0^i)$$
for $k = 0, 1, 2, \ldots, \tau - 1$ and $i = 1, \ldots, p$.
3) Partition the data samples into regression points $\mathsf{P}_r \subset \{0, \ldots, \tau - 1\}$ and physics points $\mathsf{P}_p \subset \{0, \ldots, \tau - 1\}$ such that $\mathsf{P}_r \cap \mathsf{P}_p = \emptyset$.

Remark 1. We provide the following guidelines for generating synthetic data trajectories:

(i) The initial conditions $x_0^1, \ldots, x_0^p$ can be chosen using the Latin hypercube sampling method; see [14].

(ii) Choosing $z_0^1, \ldots, z_0^p$ arbitrarily results in large regression errors for the initial time samples until the effect of the initial condition vanishes in $z(t; z_0^i)$ due to A being Hurwitz. To avoid this, we follow a technique suggested by [16]: (a) Arbitrarily choose p non-zero points $z_\tau^1, \ldots, z_\tau^p$ in $\mathcal{Z} \subset \mathbb{R}^{n_z}$, where $\tau < 0$ is such that $\|\exp(A(t - \tau)) z_\tau^i\| \leq \epsilon$ for some small $\epsilon > 0$ and $t = 0$. Solving this inequality for τ gives
$$\tau \leq \frac{1}{\lambda_{\min}(A)} \ln\left(\frac{\epsilon}{\operatorname{cond}(V)\bar{z}_\tau}\right)$$
where $\bar{z}_\tau = \max_i \|z_\tau^i\|$ and V is obtained from the eigendecomposition $A = V\Lambda V^{-1}$. (b) Simulate (1) from $x_0^1, \ldots, x_0^p$ in backward time and obtain output trajectories $h(x(t; x_0^i))$ for $t \in [\tau, 0]$. (c) Simulate (2) from $z_\tau^1, \ldots, z_\tau^p$ in forward time and obtain $z(t; z_\tau^i)$ for $t \in [\tau, 0]$. (d) Choose $z_0^i = z(0; z_\tau^i)$, for $i = 1, \ldots, p$, which is approximately equal to $\mathcal{T}(x_0^i)$.

(iii) A simple way to partition the data samples into regression points P_r and physics points P_p is to, for instance, choose even samples for P_r and odd samples for P_p. ⋄

[3] See [11] for some of the examples.

B. Defining the Empirical Loss Function

The regression problem minimizes a loss function that accounts for the deviation of the neural network's output with respect to the training data generated previously. To this end, we can exploit both $x^i(t_k)$ and $z^i(t_k)$ for learning $\mathcal{T}$ and $\mathcal{T}^*$ because both trajectories can be generated easily. The empirical loss function is defined as a *mean squared error*

$$\begin{aligned}\mathcal{L}_{\theta,\eta}(X,Z) \doteq \frac{1}{p}\sum_{i=1}^{p}\frac{1}{|\mathsf{P}_r|}\sum_{k\in\mathsf{P}_r}&\big\|z^i(t_k)-\hat{\mathcal{T}}_\theta(x^i(t_k))\big\|^2\\ &+\chi\big\|x^i(t_k)-\hat{\mathcal{T}}^*_\eta(\hat{\mathcal{T}}_\theta(x^i(t_k)))\big\|^2\end{aligned} \tag{7}$$

where $\chi > 0$ is a hyperparameter that not only weights the loss function properly but also discounts for different units of measurement of $x^i(t_k)$ and $z^i(t_k)$. Also, $X \in \mathbb{R}^{pn_x \times \tau}$ and $Z \in \mathbb{R}^{pn_z \times \tau}$ are defined as

$$X \doteq \begin{bmatrix} x^1(t_0) & x^1(t_1) & \dots & x^1(t_{\tau-1}) \\ x^2(t_0) & x^2(t_1) & \dots & x^2(t_{\tau-1}) \\ \vdots & \vdots & \ddots & \vdots \\ x^p(t_0) & x^p(t_1) & \dots & x^p(t_{\tau-1}) \end{bmatrix}$$

$$Z \doteq \begin{bmatrix} z^1(t_0) & z^1(t_1) & \dots & z^1(t_{\tau-1}) \\ z^2(t_0) & z^2(t_1) & \dots & z^2(t_{\tau-1}) \\ \vdots & \vdots & \ddots & \vdots \\ z^p(t_0) & z^p(t_1) & \dots & z^p(t_{\tau-1}) \end{bmatrix}.$$

C. Enforcing the PDE Constraint

An additional requirement of the learning problem is that $\hat{\mathcal{T}}_\theta$ must satisfy the PDE (3) for every sample in $\mathcal{X}^{\text{train}}$. Evaluating (3) for all the physics points P_p, we define the *mean squared residual* of the PDE (3) over $\mathcal{X}^{\text{train}}$ as

$$\begin{aligned}\mathcal{N}_\theta(X) \doteq \frac{1}{p}\sum_{i=1}^{p}\frac{1}{|\mathsf{P}_p|}\sum_{k\in\mathsf{P}_p}&\Big\|\frac{\partial\hat{\mathcal{T}}_\theta}{\partial x}(x^i(t_k))f(x^i(t_k))\\ &-A\hat{\mathcal{T}}_\theta(x^i(t_k))-Bh(x^i(t_k))\Big\|^2\end{aligned} \tag{8}$$

Enforcing the PDE constraint essentially avoids overfitting on the training samples and improves generalization by regularizing the neural network $\hat{\mathcal{T}}_\theta$.

D. Supervised Physics-Informed Learning Problem

By dedicating one part of the data for minimizing the mean squared error (7) and the other part for making the mean squared residual (8) equal to zero, the *supervised physics-informed learning problem* is formulated as:

$$\min_{\theta,\eta}\mathcal{L}_{\theta,\eta}(X,Z) \text{ subject to } \mathcal{N}_\theta(X)=0. \tag{9}$$

Note that (9) can be posed as

$$\min_{\theta,\eta}\mathcal{L}_{\theta,\eta}(X,Z)+\lambda\mathcal{N}_\theta(X) \tag{10}$$

for a sufficiently large Lagrange multiplier $\lambda > 0$ that discounts for the constraint $\mathcal{N}_\theta(X)=0$.

E. Testing the Learned Model on a Different Dataset

Once the neural networks $\hat{\mathcal{T}}_\theta$ and $\hat{\mathcal{T}}^*_\eta$ are trained, we evaluate the model's performance on the testing dataset $\mathcal{X}^{\text{test}} \times \mathcal{Z}^{\text{test}} \subset \mathcal{X} \times \mathcal{Z}$. It must be that the testing dataset is distinct from the training dataset for a fair evaluation of the performance. Moreover, we select multiple instances of testing dataset to tune the hyperparameters χ and λ of the trained neural networks. Among the two, λ is a critical hyperparameter in (10) that largely impacts the satisfaction of the PDE constraint and, hence, the quality of the training.

V. Evaluating the Performance of the Learned KKL Observer

The neural networks $\hat{\mathcal{T}}_\theta$ and $\hat{\mathcal{T}}^*_\eta$ are mere approximations of $\mathcal{T}$ and $\mathcal{T}^*$, respectively. Thus, the performance of the observer will be influenced by the approximation error. Moreover, the model (1) of the state dynamics and sensors is never perfect in real-world applications, and there are several underlying uncertainties that could influence the state estimation. In this section, we provide robustness guarantees for the estimation error under both the approximation error and the system uncertainties. We also provide a metric to assess the generalization capability of the observer beyond the training data and discuss the specific features of the proposed learning method that avoid overfitting and enable better generalization as compared to other techniques.

A. Robustness to the Approximation Error

Given that the activation functions of the neural network are Lipschitz continuous, it can be shown that $\hat{\mathcal{T}}^*_\eta$ is also Lipschitz, i.e., there exists ℓ^* such that, for every $\hat{z}, z \in \mathbb{R}^{n_z}$,

$$\|\hat{\mathcal{T}}^*_\eta(\hat{z}(t))-\hat{\mathcal{T}}^*_\eta(z(t))\| \le \ell^*\|\hat{z}(t)-z(t)\|. \tag{11}$$

Specifically, we remark that ReLU networks are Lipschitz continuous, which is particularly important because we consider such a network in Section VI. It is important to further remark that theoretical computation of the Lipschitz constant turns out to be quite conservative in practice. Although an NP-hard problem, empirically estimating a minimal Lipschitz constant of neural networks has been investigated extensively in the machine learning community [21]–[24].

For any $z \in \mathbb{R}^{n_z}$, $\mathcal{T}^*(z)$ can be written as

$$\mathcal{T}^*(z) = \hat{\mathcal{T}}^*_\eta(z) + \mathcal{E}^*(z) \tag{12}$$

where $\mathcal{E}^*(z)$ is the approximation error of $\hat{\mathcal{T}}^*$ at z. Because the state space $\mathcal{X} \subset \mathbb{R}^{n_x}$ is bounded, $h(\cdot)$ is a smooth map, and A is Hurwitz, there exists a compact set $\mathcal{Z} \subset \mathbb{R}^{n_z}$ containing the trajectory $z(t;\mathcal{T}(x_0))$ of (2) for every $t \geq 0$ and every $x_0 \in \mathcal{X}$. Thus, as a consequence of (5) and (11), there exists a finite approximation bound $\epsilon^* > 0$ satisfying

$$\epsilon^* = \sup_{z\in\mathcal{Z}}\|\mathcal{E}^*(z)\|. \tag{13}$$

There have been several attempts [25]–[28] to estimate ϵ^* and to show that it can be reduced by improving the design and learning technique of the neural network, and also by increasing the size of the dataset (X, Z) (see [29]).

Using (12), we can write the KKL observer (4) as

$$\begin{aligned} \dot{\hat{z}} &= A\hat{z} + By; \quad \hat{z}(0) = \hat{z}_0 \\ \hat{x} &= \hat{\mathcal{T}}_\eta^*(\hat{z}) + \mathcal{E}^*(\hat{z}) \end{aligned} \tag{14}$$

where the approximation error $\mathcal{E}^*(\hat{z})$ is an unknown signal.

Proposition 2. *Subject to Assumptions 1 and 2, there exist positive constants* $b, c > 0$ *such that the estimation error* $\tilde{x}(t) = \hat{x}(t) - x(t)$ *of* (14) *satisfies*

$$\|\tilde{x}(t)\| \leq be^{-ct} + \epsilon^*, \quad \forall t \in \mathbb{R}_{\geq 0} \tag{15}$$

where ϵ^* *is given in* (13).

Proof. We have

$$\begin{aligned} \|\tilde{x}(t)\| &= \|\hat{\mathcal{T}}_\eta^*(\hat{z}(t)) - \mathcal{T}^*(z(t))\| \\ &= \|\hat{\mathcal{T}}_\eta^*(\hat{z}(t)) - \hat{\mathcal{T}}_\eta^*(z(t)) - \mathcal{E}^*(z(t))\| \\ &\leq \|\hat{\mathcal{T}}_\eta^*(\hat{z}(t)) - \hat{\mathcal{T}}_\eta^*(z(t))\| + \|\mathcal{E}^*(z(t))\| \\ &\leq \ell^*\|\hat{z}(t) - z(t)\| + \epsilon^* \end{aligned} \tag{16}$$

where the first step is due to (12), the second step is due to the triangle inequality, and the last step is due to (11) and (13). Since A is Hurwitz, there exist $a, c > 0$ such that $\|\hat{z}(t) - z(t)\| \leq ae^{-ct}$, which completes the proof. □

B. Robustness to Model Uncertainties and Sensor Noise

Consider a nonlinear system

$$\dot{x} = f(x) + w; \quad y = h(x) + v \tag{17}$$

where $w(t) \in \mathbb{R}^{n_x}$ and $v(t) \in \mathbb{R}^{n_y}$ are unknown but essentially bounded signals. In (17), the functions $f(\cdot)$ and $h(\cdot)$ represent the model of the system, and $w(t)$ represent model uncertainties and $v(t)$ the sensor noise.

We remark that the design method of KKL observers as presented in Sections II and IV remains the same for (17). However, to better attenuate the effects of uncertainties and noise, one can seek an $\mathcal{H}_\infty$-based design [30] of matrices A and B in the linear part of the KKL observer under the constraints that A is Hurwitz and (A, B) is controllable.

Proposition 3. *Let Assumptions 1 and 2 hold. Then, if* $\|w\|_{L^\infty} \leq \bar{w}$ *and* $\|v\|_{L^\infty} \leq \bar{v}$*, for every* $t \in \mathbb{R}_{\geq 0}$*, there exist positive constants* $b, c, \alpha_1, \alpha_2 > 0$ *such that the estimation error* $\tilde{x}(t) = \hat{x}(t) - x(t)$ *of* (14) *satisfies*

$$\|\tilde{x}(t)\| \leq be^{-ct} + \alpha_1\bar{w} + \alpha_2\bar{v} + \epsilon^*, \quad \forall t \in \mathbb{R}_{\geq 0} \tag{18}$$

where ϵ^* *is given in* (13).

Proof idea. The proof follows from (16) and the linear analysis of the error $\hat{z}(t) - z(t)$. □

Given that the model uncertainties and sensor noise are bounded, the above result shows that the KKL observer is robust in terms of input-to-state stability of the estimation error; see [19]. Moreover, it can as well be shown that the constants b, c, α_1, α_2 in (18) are computable because of the linear dynamics of the KKL observer.

C. Assessing the Observer's Generalization Capability

Another key contribution in this paper is to evaluate the performance of the learned KKL observer even when the true initial condition of the system in real-time is far from the training region $\mathcal{X}^{\text{train}}$. To this end, we define a metric quantifying the generalization capability of the trained model in Figure 1 for the KKL observer. This metric compares the estimation errors resulting from the training and the testing phases, and describes how the error varies as a function of the distance between the two sets $\mathcal{X}^{\text{train}}$ and $\mathcal{X}^{\text{test}}$.

Let the testing region $\mathcal{X}^{\text{test}} \subset \mathcal{X} \setminus \mathcal{X}^{\text{train}}$, and consider a set of points $\{\xi_0^j : j = 1, \ldots, q\} \in \mathcal{X}^{\text{test}}$ that, for every $j \in \{1, \ldots, q\}$, satisfy $d(\xi_0^j, \mathcal{X}^{\text{train}}) = \delta$, for some $\delta > 0$, where

$$d(\xi_0^j, \mathcal{X}^{\text{train}}) \doteq \inf_{x_0 \in \mathcal{X}^{\text{train}}} \|x_0 - \xi_0^j\|.$$

The *empirical generalization error* $\mathrm{G}_{\text{emp}}(\delta)$ is defined as

$$\mathrm{G}_{\text{emp}}(\delta) \doteq |\mathrm{E}_{\text{test}}(\delta) - \mathrm{E}_{\text{train}}| \tag{19}$$

where

$$\begin{aligned} \mathrm{E}_{\text{test}}(\delta) &\doteq \frac{1}{q}\sum_{j=1}^{q}\frac{1}{\tau}\sum_{k=0}^{\tau-1}\frac{\|\hat{x}(t_k; \hat{\xi}_0^j) - x(t_k; \xi_0^j(\delta))\|^2}{\|x(t_k; \xi_0^j(\delta))\|^2} \\ \mathrm{E}_{\text{train}} &\doteq \frac{1}{p}\sum_{i=1}^{p}\frac{1}{\tau}\sum_{k=0}^{\tau-1}\frac{\|\hat{x}(t_k; \hat{x}_0^i) - x(t_k; x_0^i)\|^2}{\|x(t_k; x_0^i)\|^2} \end{aligned}$$

with $\hat{\xi}_0^j$ and $\hat{x}_0^i$ chosen sufficiently close to $\xi_0^j(\delta)$ and x_0^i, respectively, to avoid the errors accumulated in the observer's transient. Notice that E_{test} denotes the normalized mean estimation error variance of multiple test trajectories initialized at δ-distance from $\mathcal{X}^{\text{train}}$, whereas $\mathrm{E}_{\text{train}}$ denotes the normalized mean estimation error variance of all the training trajectories.

In short, during the testing phase, we select q initial points ξ_0^j that are δ-distant from the training region, where $\delta \in \{\delta_1, \ldots, \delta_m\}$ with $0 < \delta_1 < \cdots < \delta_m$. Then, for each δ_i, the change in the normalized testing error variance $\mathrm{E}_{\text{test}}(\delta_i)$ provides an empirical quality measure (19) on the generalization capability of the learned KKL observer.

D. Discussion on the Observer's Generalization Capability

Since Assumptions 1 and 2 ensure uniform injectivity of $\mathcal{T}$, and $\mathcal{T}$ satisfies the PDE (3), the inverse $\mathcal{T}^*$ exists and is unique. Thus, the data samples used in the training are of the form $(x, \mathcal{T}(x))$ and $(z, \mathcal{T}^*(z))$, which entails that the problem (10) is a *realizable* learning task that is *probably approximately correct* (PAC) learnable [31]. Then, one of the sources for non-zero generalization error is the fact that the training data (X, Z) induced loss $\mathcal{L}_{\theta,\eta}(X, Z)$ in (7) is an approximation of the actual loss

$$\begin{aligned} \bar{\mathcal{L}}_{\theta,\eta}(x, z) \doteq \int_{\mathcal{X}}\int_0^T &\|z(t; \mathcal{T}(\xi)) - \mathcal{T}(x(t; \xi))\| \\ &+ \chi\|x(t; \xi) - \mathcal{T}^*(\mathcal{T}(x(t; \xi)))\| \mathrm{d}t \, \mathrm{d}\mu(\xi) \end{aligned}$$

where μ is a measure on $\mathcal{X}$.

In our formulation, an unlimited amount of synthetic data can be generated using the method described in Section IV-A, which enables one to enhance the generalization capability of the learned KKL observer and improve its performance. However, it is not practical to utilize arbitrarily large amount of data for training. Therefore, under the same training data size, a key feature that makes the supervised PINN to have better generalization capability than the neural network architectures of [14]–[16] is the regularization with the PDE (3), which reduces the search space of the hypothesis and avoids overfitting on the training data.

In the unsupervised AE architecture of [16], the neural network is also regularized by the PDE (3). However, unlike (7), the loss function of [16] doesn't include additional regression term that accounts for the deviation between $z^i(t_k)$ and $\hat{\mathcal{T}}_\theta(x^i(t_k))$. This is very important because without the explicit supervision to connect the system's state space $\mathcal{X}$ to the observer's state space $\mathcal{Z}$, the AE will minimize the reconstruction loss on a limited number of training samples $x^i(t_k)$, which may belong to a larger hypothesis space. Thus, the unsupervised AE of [16] makes the neural network overfit upon the partial training data, i.e., only in the x-domain, and hinders the generalization on the unseen data. In the extreme case, without the PDE regularization, if the decoder is complex enough, one could essentially recover the x sample even from noise, and the learned left inverse $\hat{\mathcal{T}}^*_\eta$ can as well be arbitrary [32].

VI. Experimentation and Testing

Performance of the proposed supervised PINN-based KKL observer is numerically tested under different scenarios. First, we test its performance under approximation errors when the state trajectory is initialized outside the training region $\mathcal{X}^{\text{train}}$. Second, we test its performance under model uncertainties and sensor noise and demonstrate the robustness of the proposed observer. Third, we examine the estimation error trajectories for multiple experiments where the system's state is always initialized randomly outside $\mathcal{X}^{\text{train}}$. We show that the proposed supervised PINN-based KKL observer demonstrates better performance than 1) supervised NN [14]–[16] and 2) unsupervised AE [16]. Finally, we compare the empirical generalization error resulting from all these techniques and demonstrate that our method exhibits better generalization capabilities.

For the experimentation and testing, we consider the following nonlinear oscillators:

- Reverse Duffing oscillator

$$\dot{x}_1 = x_2^3, \quad \dot{x}_2 = -x_1; \quad y = x_1. \tag{20}$$

- Rössler attractor

$$\begin{aligned} \dot{x}_1 &= -x_2 - x_3, & \dot{x}_2 &= x_1 + a x_2 \\ \dot{x}_3 &= b + x_3(x_1 - c); & y &= x_2 \end{aligned} \tag{21}$$

where the parameters $a = 0.2$, $b = 0.2$, and $c = 5.7$.

A. Experimental Setup for Training and Testing

For both (20) and (21), we follow the data generation and sampling procedure described in Section IV-A. For reverse Duffing oscillator, $\mathcal{X}^{\text{train}} = [-1,1]^2$. For Rössler attractor, $\mathcal{X}^{\text{train}} = [-1,1]^3$. We generate $\{x_0^1, ..., x_0^p\}$ using Latin hypercube sampling method. The initial conditions $\{z_0^1, \dots, z_0^p\}$ are generated using Remark 1(ii). Runge-Kutta-4 is used as the numerical ODE-solver for (20)-(21) over a time horizon $[0, 50]$.

The architecture of both neural networks $\hat{\mathcal{T}}_\theta$ and $\hat{\mathcal{T}}^*_\eta$ in Figure 1 is chosen to be a multi-layer perceptron with five hidden layers, where each layer has 50 neurons with ReLU activation function. We use normalization and denormalization layer for data standardization in order to facilitate the training. Training is further facilitated by a learning rate scheduler. All models in this section are trained using the `Adam` optimization algorithm with a batch size of 32. In the testing stage, initial conditions are generated outside the training domain, from which (20) and (21) are then simulated. For the code and other details, please refer to our repository[4].

The matrices of the KKL observer are chosen as follows:

$$A = -\text{diag}(1, 2, \dots, n_z), \quad B = 1_{n_z}$$

where $n_z = n_y(2n_x + 1)$, diag() denotes a diagonal matrix, and 1_{n_z} is a vector of ones with dimensions $n_z \times 1$. Notice that $n_y = 1$ for both (20) and (21).

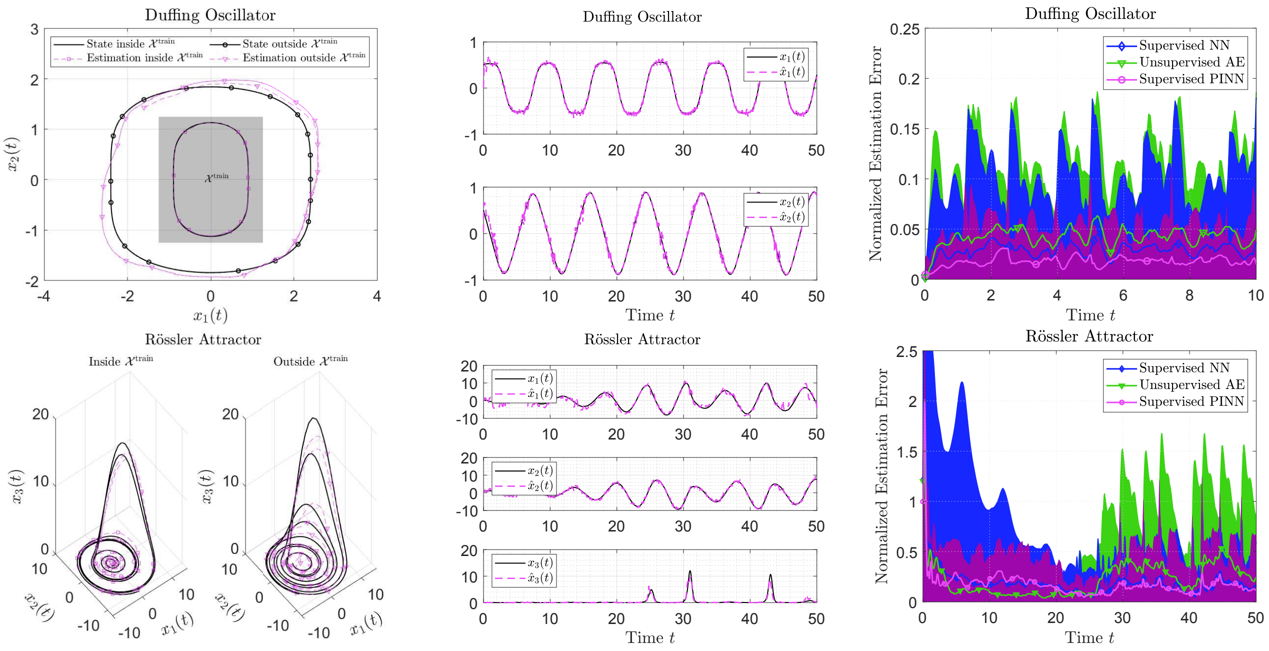
B. Experimental Results

In the following, we present several experimental results and compare our method *supervised PINN* with *supervised NN* [14]–[16] and *unsupervised AE* [16].

1) Testing the supervised PINN-based KKL observer outside the training region: We train the supervised PINN inside the training regions for both (20) and (21). We test it outside the training region. Figure 2a demonstrates the estimation performance of the learned KKL observer when the true system is initialized inside the training region $\mathcal{X}^{\text{train}}$ and outside the training region. Despite an expected deterioration of the state estimation outside the training region, the observer's performance is satisfactory as it is able to follow the true state with a small error.

2) Testing the supervised PINN-based KKL observer under model uncertainties and sensor noise: We randomly initialize the state trajectories inside the training region $\mathcal{X}^{\text{train}}$, where the initial point is different from the initial points in the training dataset X. We consider $w(t) \sim \mathcal{N}(0, 0.1)$ and $v(t) \sim \mathcal{N}(0, 0.1)$ for (20), $w(t) \sim \mathcal{N}(0, 1)$ and $v(t) \sim \mathcal{N}(0, 1)$ for (21). Figure 2b shows the true and estimated state trajectories, and demonstrates that the learned KKL observer is stable under uncertainties and noise as stated in Proposition 3.

[4] https://github.com/Mudhdhoo/ACC_KKL_Observer



(a) (b) (c)

Fig. 2: (a) Phase portrait of the estimation performance when the system is initialized inside and outside the training region. (b) Estimation performance in the presence of model uncertainties and sensor noise. (c) Comparison of our method with others in terms of the range of normalized estimation errors and their averages for 50 state trajectories initialized outside the training region.

3) Estimation errors for multiple state trajectories initialized outside the training region: We initialize the systems (20) and (21) at 50 points that are randomly generated outside the training region. We run the KKL observers that are learned according to supervised NN, unsupervised AE, and our method supervised PINN. To show the merits of each learning scheme, we compare *normalized estimation error* trajectories

$$e_i(t) = \frac{\|\hat{x}^i(t) - x^i(t)\|}{\|x^i(t)\|}; \quad i = 1, \ldots, 50.$$

Figure 2c demonstrates the error ranges and the average ($\sum_{i=1}^{50} e_i(t)/50$) for each learning scheme. For the reverse Duffing oscillator, our method yields lowest maximum and average error for all times. For the Rössler attractor, the overall performance of our method is better than both the supervised NN and unsupervised AE. The supervised NN performs worse in the beginning, which is before the bifurcation of the Rössler attractor, because it fails to capture some trajectories that are initialized outside the training region. On the other hand, the unsupervised AE performs worse after the bifurcation because it is not very sensitive to changes in the z-domain that correspond to the bifurcation in the x-domain.

4) Comparison of the empirical generalization error for multiple learning schemes: We choose multiple initial points outside the training region for each $\delta_i > 0$ in the testing phase as described in Section V-C. We only consider reverse Duffing oscillator (20) for this experiment. We choose multiple $\delta_i \in \{0.5, 1, 1.5, \ldots, 10\}$, and, for each δ_i, we choose 10 initial points in circular formation centered around $[-1, 1]^2$ outside $\mathcal{X}^{\text{train}}$. Figure 3 illustrates the comparison of different learning schemes in terms of empirical generalization error. For all δ_i, it can be seen that supervised PINN yields smaller generalization errors.

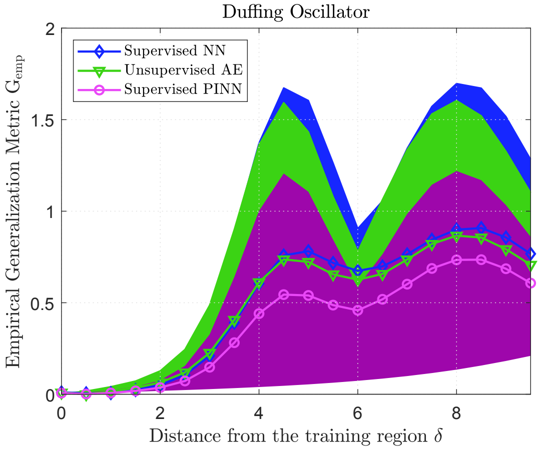


Fig. 3: Comparison of the empirical generalization error as the initial state of reverse Duffing oscillator is at a distance δ from the training region $\mathcal{X}^{\text{train}}$.

VII. Discussion and Future Outlook

We proposed a novel supervised physics-informed learning method to design Luenberger or KKL observers for autonomous nonlinear systems. The proposed method learns the nonlinear transformation map required to transform the system to the observer's coordinates and satisfies a certain PDE constraint. Additionally, the inverse of the transformation map is learned to obtain the state estimate in the original state space. To learn both the transformation map and its inverse, we trained a physics-informed neural network architecture on synthetic data generated by numerically solving both the system and the observer. The PDE constraint acts as a physical invariant that regularizes the neural network, reducing the hypothesis's search space. We demonstrated that the KKL observer designed with our method is robust to neural network's approximation error, model uncertainties, and sensor noise. The proposed method also exhibits better generalization properties than other methods due to the PDE regularization and the regression loss in the observer's coordinates. We validated our results on reverse Duffing oscillator and Rössler attractor.

While we discussed the generalization capability of the proposed learning-based observer design method in detail, theoretical guarantees on its generalizability remain an open problem. Additionally, designing a KKL observer optimally to improve its robustness to model uncertainties and sensor noise is left for future work. We also recognize the potential of alternative methods such as operator learning [33] to learn the non-linear transformation map, which is a prospect to be explored. Furthermore, the proposed method can be extended beyond KKL observers to obtain the triangular form of nonlinear systems required in designing high-gain and backstepping observers. In conclusion, the proposed learning-based observer design method can be a promising solution to address the challenging problem of designing observers for nonlinear systems.

References

[1] D. G. Luenberger, "Observing the state of a linear system," *IEEE Transactions on Military Electronics*, vol. 8, no. 2, pp. 74–80, 1964.

[2] A. Shoshitaishvili, "Singularities for projections of integral manifolds with applications to control and observation problems," in *Theory of Singularities and its Applications*. American Mathematical Society, 1990, pp. 295–333.

[3] ——, "On control branching systems with degenerate linearization," in *IFAC Symposium on Nonlinear Control Systems*, 1992, pp. 495–500.

[4] N. Kazantzis and C. Kravaris, "Nonlinear observer design using lyapunov's auxiliary theorem," *Systems & Control Letters*, vol. 34, no. 5, pp. 241–247, 1998.

[5] A. J. Krener and M. Xiao, "Nonlinear observer design in the siegel domain," *SIAM Journal on Control and Optimization*, vol. 41, no. 3, pp. 932–953, 2002.

[6] G. Kreisselmeier and R. Engel, "Nonlinear observers for autonomous Lipschitz continuous systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 451–464, 2003.

[7] V. Andrieu and L. Praly, "On the existence of a Kazantzis–Kravaris/Luenberger observer," *SIAM Journal on Control and Optimization*, vol. 45, no. 2, pp. 432–456, 2006.

[8] V. Andrieu, "Convergence speed of nonlinear Luenberger observers," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2831–2856, 2014.

[9] R. Engel, "Nonlinear observers for Lipschitz continuous systems with inputs," *International Journal of Control*, vol. 80, no. 4, pp. 495–508, 2007.

[10] P. Bernard, "Luenberger observers for nonlinear controlled systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 3676–3681.

[11] P. Bernard and V. Andrieu, "Luenberger observers for nonautonomous nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 270–281, 2018.

[12] V. Andrieu and P. Bernard, "Remarks about the numerical inversion of injective nonlinear maps," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 5428–5434.

[13] P. Bernard, V. Andrieu, and D. Astolfi, "Observer design for continuous-time dynamical systems," *Annual Reviews in Control*, 2022.

[14] L. d. C. Ramos, F. Di Meglio, V. Morgenthaler, L. F. F. da Silva, and P. Bernard, "Numerical design of Luenberger observers for nonlinear systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 5435–5442.

[15] J. Peralez and M. Nadri, "Deep learning-based Luenberger observer design for discrete-time nonlinear systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 4370–4375.

[16] M. Buisson-Fenet, L. Bahr, and F. Di Meglio, "Towards gain tuning for numerical KKL observers," *arXiv preprint arXiv:2204.00318*, 2022.

[17] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[18] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[19] E. D. Sontag and Y. Wang, "On characterizations of the input-to-state stability property," *Systems & Control Letters*, vol. 24, no. 5, pp. 351–359, 1995.

[20] L. Brivadis, V. Andrieu, P. Bernard, and U. Serres, "Further remarks on KKL observers," *HAL preprint HAL-03695863*, 2022.

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[22] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: Analysis and efficient estimation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[23] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and accurate estimation of Lipschitz constants for deep neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[24] M. Jordan and A. G. Dimakis, "Exactly computing the local Lipschitz constant of ReLU networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7344–7353, 2020.

[25] J. Sokolić, R. Giryes, G. Sapiro, and M. R. Rodrigues, "Robust large margin deep neural networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4265–4280, 2017.

[26] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," *arXiv preprint arXiv:1710.05468*, 2017.

[27] D. Jakubovitz, R. Giryes, and M. R. Rodrigues, "Generalization error in deep learning," in *Compressed sensing and its applications*. Springer, 2019, pp. 153–193.

[28] Y. Cao and Q. Gu, "Generalization bounds of stochastic gradient descent for wide and deep neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[29] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT press, 2018.

[30] A. Zemouche, R. Rajamani, B. Boulkroune, H. Rafaralahy, and M. Zasadzinski, "$\mathcal{H}_\infty$ circle criterion observer design for Lipschitz nonlinear systems with enhanced LMI conditions," in *2016 American Control Conference (ACC)*, 2016, pp. 131–136.

[31] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[32] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, "Conditional image generation with PixelCNN decoders," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[33] G. Kissas, J. H. Seidman, L. F. Guilhoto, V. M. Preciado, G. J. Pappas, and P. Perdikaris, "Learning operators with coupled attention," *Journal of Machine Learning Research*, vol. 23, no. 215, pp. 1–63, 2022.