Neural Network-Based KKL observer for nonlinear discrete-time systems

Johan Peralez, Madiha Nadri, Daniele Astolfi

▶ **To cite this version:**

Johan Peralez, Madiha Nadri, Daniele Astolfi. Neural Network-Based KKL observer for nonlinear discrete-time systems. 2022 IEEE 61st Conference on Decision and Control (CDC), Dec 2022, Cancun, Mexico. 10.1109/CDC51059.2022.9992516 . hal-04005698

HAL Id: hal-04005698
https://hal.science/hal-04005698

Submitted on 27 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Neural Network-Based KKL observer for nonlinear discrete-time systems

Johan Peralez[1], Madiha Nadri[1*] and Daniele Astolfi[1]

***Abstract*— For non-autonomous multivariable discrete-time nonlinear systems, we address the state estimation problem using a Kazantzis-Kravaris-Luenberger (KKL) observer. We aim to build a mapping that transforms a nonlinear dynamics into a stable linear system modulo an output injection and to design an asymptotic observer. However, this mapping is difficult to compute and its numerical approximation may be badly conditioned during the transient phase. We propose an algorithm based on ensemble learning techniques to improve the numerical approximation of the mapping and its extension in the transient phase. This ensures a good asymptotic convergence of the observer and avoids peaking phenomena. The algorithm demonstrates good performance in high-dimensional and multi-input-multi-output examples.**

I. Introduction

The problem of online estimation of the state of nonlinear dynamic systems has been a major research topic in control for many decades. Indeed, it is often the case that nonlinear control laws depend on state information making the observer an essential estimation tool [1]. However, only few results addressing observer design for discrete-time systems can be found in the literature, see, e.g. [2]–[5]. These approaches rely on linearization methods, and thus provide local convergence only.

An alternative strategy dealing with strong non-linearities in observer design is to find a change of coordinates, which makes nonlinear dynamics approximately linear or in canonical forms [4]. Despite the theoretical strength of these approaches, it is still is a challenge in the observer design field to compute these transformations. One of these approaches is the Kazantzis-Kravaris-Luenberger (KKL) observer, which has been transposed to the discrete-time nonlinear case by N. Kazantzis and C. Kravaris in [5] and recently extended in [2]. Based on an immersion technique, the nonlinear system is transformed into a linear system modulo an output injection [6]. Although promising, the difficulty in synthesizing the coordinate transformation remains high, which makes the application of the approach to industrial problems difficult [7].

Data-driven approaches may be an alternative worth exploring to overcome this problem, the objective being to develop a systematic algorithm to construct this mapping and design an asymptotic observer learned from data. A first attempt for the design of a KKL-based neural network observers for nonlinear time-continuous systems has been reported in [8]. Therein, the authors rely on the stability of the observer to eliminate the early stages (the burn-in phase) of the simulation from the dataset. Unfortunately, this procedure does not guarantee a proper exploration of the state space during training and, in the case of non autonomous systems, a time varying mapping is required. More recently, [9] addresses these shortcomings through unsupervised learning, demonstrating significant improvements in term of accuracy.

In this body of work, despite promising results, the proposed neural based KKL observers were only assessed on single-input single-ouput (SISO) systems with low state dimensions (only 2 states). A further drawback are strong assumptions required to ensure asymptotic convergence of the state in the initial coordinates.

In the present paper, we build on the methodology based on unsupervised learning introduced in [9] and improve this approach in several ways: i) we provide an extension relaxing some of the assumptions and address the problem of peaking phenomena; ii) using a neural network based ensemble approach, we improve the convergence accuracy, and iii) to validate our observer design methodologies, we consider two applications where the observer is an important task for process control: a multi-inlet multi-outlet heat exchanger conventionally used in industry and a challenging nonlinear model of Korteweg De Vries (KdV) where, to our knowledge, only local observers exist in the literature.

II. Observer Design

A. Problem Statement and assumptions

In this work, we consider nonlinear discrete-time systems of general form

$$\left\{ \begin{array}{lcl} x_{k+1} & = & F(x_k, u_k) \\ y_k & = & h(x_k) \end{array} \right. \tag{1}$$

where $x \in \mathbb{R}^{d_x}$ is the state, $u \in \mathbb{R}^{d_u}$ is a known control input, $y \in \mathbb{R}^{d_y}$ the measured output, and F and h are suitable functions. In the following, in the particular case where the control input is maintained at a constant value u^0, system (1) is referred to as an *autonomous system*. In order to design an observer for system (1), we require the system to be time reversible backward distinguishable. More formally, we consider the case in which the vector fields F and h satisfy the following assumptions.

Assumption 1. *There exists a compact set $\mathcal{X}$ such that for any solution x to* (1)*, $x_k \in \mathcal{X}$ for all $k \geq 0$.*

Assumption 2. *For all u^0 of interest, $F(., u^0)$ is invertible and F^{-1} and h are of class C^1 and globally Lipschitz.*

[1] Univ Lyon, Université Claude Bernard Lyon 1, CNRS, LAGEPP UMR 5007, 43 boulevard du 11 novembre 1918, F-69100, Villeurbanne France. Johan.peralez@univ-lyon1.fr, madiha.nadri-wolf@univ-lyon1.fr, daniele.astolfi@univ-lyon1.fr

*This research was partially supported by the French Grant ANR DELICIO (ANR-19-CE23-0006)

Assumption 3. *For all u^0 of interest and for all $(x_1, x_2) \in \mathcal{X}^2$, if $x_1 \neq x_2$, then there exists a positive integer i such that $h(F^{-i}(x_1, u^0)) \neq h(F^{-i}(x_2, u^0))$.*

B. Preliminary results

In [2], following the Luenberger-like methodology, the authors have shown that under Assumptions 1-3, for any pair of matrices $A \in \mathbb{R}^{d_z \times d_z}$ and $B \in \mathbb{R}^{d_z \times d_x}$ there exists a continuous mapping $T : \mathcal{X} \to \mathbb{R}^{d_z}$ satisfying

$$T(F(x, u^0)) = AT(x) + Bh(x) \quad \forall x \in \mathcal{X}. \tag{2}$$

Most importantly, it is shown that T is unique and admits a left-inverse on $T(\mathcal{X})$ if $d_z \geq d_y(d_x + 1)$, (A, B) is controllable and A is Schur stable. In the following we denote $\mathcal{T}$ this inverse:

$$\mathcal{T}(T(x)) = x \quad \forall x \in \mathcal{X}. \tag{3}$$

In [9] the transformations T and $\mathcal{T}$ defined by equations (2)-(3) are used to construct an observer for the non-autonomous case of System (1) as follows

$$\left\{ \begin{array}{lcl} \hat{z}_{k+1} & = & A\hat{z}_k + By_k + \Psi(\hat{z}, u_k) \\ \hat{x}_k & = & \mathcal{T}(\hat{z}_k), \end{array} \right. \tag{4}$$

with

$$\Psi(\hat{z}, u_k) := T(F(\mathcal{T}(\hat{z}_k), u_k)) - T(F(\mathcal{T}(\hat{z}_k), u^0)).$$

The eigenvalues of A have to be chosen fast enough. A simple choice, for the single-output case, is to select

$$A = \mathrm{diag}(\lambda_1, \ldots, \lambda_{d_z}), \quad B = (1, \ldots, 1)^\top, \tag{5}$$
$$\max_{i \in \{1,\ldots,d_z\}} |\lambda_i| < \min\left\{1, \frac{1}{c}\right\}, \quad c = \sup_{x \in \mathcal{X}} \left| \left(F^{-1}\right)'(x) \right|.$$

In order to ensure the asymptotic convergence of the state in the initial coordinates it was assumed that the observer state in the latent space $\hat{z}$ remains in $T(\mathcal{X})$. Unfortunately, as highlighted in Section IV, even when $\hat{z}$ is initialized in $T(\mathcal{X})$ the observer state tends to leave this set during the transient phase. This phenomenon is also known as peaking phenomenon (well-known in the context of continuous-time observers [1] and in particular for high-gain observers). To address this issue, an extension of the mapping $\mathcal{T}$ is detailed in the following section.

C. Algorithm extension

The mapping $\mathcal{T}$ is defined in equation (3) by inversion of T on the open set $T(\mathcal{X})$. If the trajectories of the observer state $\hat{z}$ leave this set, it is necessary to extend its domain of definition. Inspired by the solution proposed for continuous time systems in [6], in what follows, we give an extension in discrete time context defined by

$$\widetilde{\mathcal{T}}(z) = \mathrm{argmin}_{x \in \mathcal{X}} |T(x) - z| \quad \forall z \in \mathbb{R}^{d_z}. \tag{6}$$

An observer for the non-autonomous system (1) is then given by

$$\left\{ \begin{array}{lcl} \hat{z}_{k+1} & = & A\hat{z}_k + By_k + \widetilde{\Psi}(\hat{z}, u_k) \\ \hat{x}_k & = & \widetilde{\mathcal{T}}(\hat{z}_k), \end{array} \right. \tag{7}$$

where now $\widetilde{\Psi}$ is defined as

$$\widetilde{\Psi}(\hat{z}, u_k) := T(F(\widetilde{\mathcal{T}}(\hat{z}_k), u_k)) - T(F(\widetilde{\mathcal{T}}(\hat{z}_k), u^0)).$$

Proposition 1. *Let A be designed as in* (5) *and assume there exists a positive constant c_Ψ satisfying*

$$\left|\widetilde{\Psi}(z^1, u) - \widetilde{\Psi}(z^2, u)\right| \leq c_\Psi \left|z^1 - z^2\right|$$

for all $u \in \mathcal{U}$ and for all $z^1, z^2 \in T(\mathcal{X})$, such that the matrix $A + 2c_\Psi I$ is Schur stable. Then any solution to (1)*,* (6)*,* (7) *verifies* $\lim_{k \to +\infty} |x_k - \hat{x}_k| = 0$.

Proof. In [9] it is shown that there exists a map T solution to

$$T(F(x_k, u^0)) = AT(x_k) + By_k.$$

Adding $T(F(x_k, u_k))$ to each side of previous equation, this becomes

$$T(F(x_k, u_k)) = AT(x_k) + By_k + T(F(x_k, u_k)) - T(F(x_k, u^0)).$$

Hence along solutions of System (1), $z_k = T(x_k)$ evolves according to

$$\begin{aligned} z_{k+1} &= T(x_{k+1}) \\ &= Az_k + By_k + \widetilde{\Psi}(z_k, u_k). \end{aligned}$$

Let us denote by e_k the estimation error $e_k = \hat{z}_k - z_k$. From (7) and above equations, we have

$$e_{k+1} = Ae_k + \Delta\Psi(e_k, z_k, u_k), \tag{8}$$

with $\Delta\Psi(e_k, z_k, u_k) := \widetilde{\Psi}(e_k + z_k, u_k) - \widetilde{\Psi}(z_k, u_k)$. Since T is injective, it is sufficient to prove that e_k converges geometrically towards zero. To this end, first show that the following Lipschitz inequality

$$\left|\widetilde{\Psi}(e_k + z_k, u) - \widetilde{\Psi}(z_k, u)\right| \leq 2c_\Psi |e_k| \tag{9}$$

holds for all k. First, note that $z_k = T(x_k) \in T(\mathcal{X})$. So if $\hat{z}_k \in T(\mathcal{X})$, the constraint (9) is satisfied by assumption. Otherwise, let us define $\tilde{z}$ as

$$\tilde{z} \quad := \quad T(\widetilde{\mathcal{T}}(\hat{z})) = \underset{\zeta \in T(\mathcal{X})}{\mathrm{argmin}} |\zeta - \hat{z}|.$$

Note that by definition, $|\tilde{z} - \hat{z}| \leq |z - \hat{z}|$. As a consequence, for all k, $\tilde{z}_k \in T(\mathcal{X})$ we obtain

$$\begin{aligned} |\Delta\Psi(e_k, z_k, u_k)| &\leq c_\Psi |\tilde{z}_k - z_k| \\ &\leq c_\Psi (|\tilde{z}_k - \hat{z}_k| + |\hat{z}_k - z_k|) \leq 2c_\Psi |e_k|, \end{aligned}$$

showing (9). Now, consider the quadratic Lyapunov function $V_k := e_k^\top e_k$, and recall that A is diagonal. We define $a := \max_i |\lambda_i|$, where λ_i denotes the eigenvalues of A.
Using (8) and (9), we compute

$$V_{k+1} = (e_k^\top A^\top + \Delta\Psi_k^\top)(Ae_k + \Delta\Psi_k)$$

We show that $V_{k+1} \leq (a + 2c_\Psi)^2 |e_k|^2$. Then by selecting $A + 2c_\Psi I$ Schur stable, we obtain $a + 2c_\Psi < 1$. Using the previous Lyapunov inequality we obtain $V_{k+1} < V_k$ and this completes the proof by invoking standard Lyapunov discrete-time arguments. □

Notice that solving the minimization of (6) is difficult in general: even if the mapping T is known, the problem may not be convex and the space of search (the compact set $\mathcal{X}$) can be of high dimension. Moreover to implement the observer (7) this optimization problem should be solved on-line at each time step. In Section III, we propose an algorithm – based on deep learning techniques – to learn not only the transformations T and $\mathcal{T}$, but also a method to approximate the extension $\widetilde{\mathcal{T}}$, with the objective to obtain computations which are tractable for a real time implementation.

D. Multi-observer approach: Ensemble

Consider a set of observers $\{\mathcal{O}^m, m = 1, ..., M\}$, where each observer $\mathcal{O}^m$ is defined by equations (7) with (possibly) different design choices for A and B:

$$\begin{cases} \hat{z}^m_{k+1} = A^m \hat{z}^m_k + B^m y_k + \tilde{\Psi}^m(\hat{z}^m_k, u_k) \\ \hat{x}^m_k = \widetilde{\mathcal{T}}(\hat{z}^m_k). \end{cases}$$

In order to provide state estimations using these multiple KKL observers (MKKL), the dynamical equation for MKKL strategy is considered by taking the average of each estimate. Indeed, using the convergence property of each observer established in Proposition 1, it is readily seen that

$$\lim_{k\to\infty} |x_k - \hat{x}_k| = \frac{1}{M} \lim_{k\to\infty} \Big| \sum_{m=1}^{M} x_k - \hat{x}^m_k \Big| = 0.$$

This approach is often used in the control context for state estimation purposes, which allows us to improve the transient response of high-gain observers [10] and to remove the peaking phenomenon [11]. In Section III a deep learning based ensemble method is detailed that takes advantage of this property in order to improve the numerical accuracy.

III. Deep Learning Methodology

In this section, we address the problem of finding the mappings T, $\mathcal{T}$ and $\widetilde{\mathcal{T}}$ (defined in Section II) through deep learning. Indeed, high-capacity deep neural network have shown to be able to learn non-linear mappings and to generalize to unseen data for a diverse range of problems.

In the following, we make use of multi layer perceptrons (MLPs) to approximate the transformations. An MLP is composed of linear functions $f_i(\zeta) = W_i\,\zeta + b_i$ (the *hidden layers*) followed by a nonlinearity (an *activation function*). Resulting MLPs have the following form

$$\begin{aligned} f(\zeta) &= f_p \circ f_{p-1} \circ \cdots \circ f_1(\zeta) \\ &= W_p \phi_p(W_{p-1} ... \phi_2(W_2 \phi_1(W_1 \zeta + b_1) + b_2) ...) + b_p, \end{aligned} \tag{10}$$

where matrices W_i and vectors b_i are the parameters to identify and ϕ_i are the *activation functions*.

A. Preliminary results

Following the methodology introduced in [9], we first learn the mappings T and $\mathcal{T}$ from data. The method is summarized in the following steps:

1) Choose A and B for the observer dynamics such that A is Schur stable and (A, B) controllable as in (5).
2) Sample a set of state values $\{x_k\}$ randomly on $\mathcal{X}$.
3) Compute corresponding outputs $\{y_k\}$ and successor states $\{x_{k+1}\}$ from the dynamical model (1) with a fixed control input u_0.
4) Learn the transformation T which satisfies the observer dynamics given by the equation (2). This dynamics is enforced using the following loss:

$$\mathcal{L}_T = |T(x_{k+1}) - (AT(x_k) + By_k)|. \tag{11}$$

   The learned mapping is then a function of the form (10) minimizing the loss (11) for the collected data $\{(x_k, y_k, x_{k+1})\}$.
5) Perform nonlinear regression to find an approximation of $\mathcal{T}$ by minimization of the following loss:

$$\mathcal{L}_{\mathcal{T}} = |x_k - \mathcal{T}(T(x_k))|. \tag{12}$$

In [9] numerical simulations have shown the ability of the method to learn the mappings T and $\mathcal{T}$ with good accuracy on examples of low dimensions. One of the main advantage of this training algorithm is to allow control over the exploration of the state space: the sampling of the state space (step 2) takes advantage of the unsupervised learning of T (step 4) that does not rely on system trajectories.

B. Algorithm extension

The method described in Section III-A is first used to learn T and $\mathcal{T}$. Knowing that the plant dynamics x evolves in some compact set $\mathcal{X}$, we would like to estimate the image of $T(\mathcal{X})$ in order to compute an (over) approximation of a set $\mathcal{Z}$ in which the observer dynamics should evolve. To identify the extension $\widetilde{\mathcal{T}}$ defined by (6), we use the following loss

$$\mathcal{L}_{\widetilde{\mathcal{T}}} = |z_k - T(\widetilde{\mathcal{T}}(z_k))| + w\; d(\widetilde{\mathcal{T}}(z_k), \mathcal{X})^2, \tag{13}$$

where $d(x_k, \mathcal{X})$ is the distance between $\widetilde{\mathcal{T}}(z_k)$ and the compact set $\mathcal{X}$, and w is a parameter chosen sufficiently large to enforce x_k to remain in $\mathcal{X}$.

To learn the mapping $\widetilde{\mathcal{T}}$ minimizing equation (13) we need to sample some observer values $\hat{z}_k$ to estimate the set in which it evolves.

Assuming its dynamics follows (2), the i^{th} element of vector $\hat{z}_k$ is denoted $\hat{z}_{k,i}$ and follows:

$$\hat{z}_{k+1,i} = \lambda_i \hat{z}_{k,i} + b_i\, y_{k,j},$$

where λ_i is the i^{th} diagonal element of A and $b_i\, y_{k,j}$ the output injection with $j = \lceil i/(d_x+1) \rceil$. Hence an estimate on the bounds of the state vector is obtained from the bounds of the output (for simplicity sake, λ_i and b_i are considered positives): $\bar{\hat{z}}_i = b_i/(1-\lambda_i)\bar{y}_j$ and $\underline{\hat{z}}_i = b_i/(1-\lambda_i)\underline{y}_j$. A set of data $\{z_k\}$ is then sampled randomly inside these bounds.

Notice that the equality $\mathcal{T}(z) = \widetilde{\mathcal{T}}(z)$ stands for all $z \in T(\mathcal{X})$. But in practice, the mapping $\widetilde{\mathcal{T}}$ being learned from data sampled in a larger set, we can expect the learned mapping $\mathcal{T}$ to be of a better accuracy on its domain of definition. We therefore propose to keep using $\mathcal{T}$ when $\hat{z} \in T(\mathcal{X})$. To numerically detect this condition, we introduce

a threshold $\epsilon_{\mathcal{T}}$ and we consider that $\hat{z} \notin T(\mathcal{X})$ when $|\hat{z} - T(\mathcal{T}(\hat{z}))| > \epsilon_{\mathcal{T}}$. We then estimate the state from

$$\hat{x} = \begin{cases} \widetilde{\mathcal{T}}(\hat{z}), & \text{if } |\hat{z} - T(\mathcal{T}(\hat{z}))| > \epsilon_{\mathcal{T}} \\ \mathcal{T}(\hat{z}), & \text{otherwise.} \end{cases}$$

C. Multi-observer approach: Ensemble

Combining predictions from multiple models (referred as *ensemble*) has been shown to be an effective approach to increase the performance of learned models. Ensemble learning has been successfully applied in different areas such as object recognition or reinforcement learning [12]. Among the most popular methods, *fusion*, *bagging* and *gradient boosting* usually demonstrate benefits in term of accuracy. However these methods suffer of high computational cost.

Unlike all these methods where M independent base models are trained, *snapshot ensemble* [13] generates the ensemble by enforcing a single base model to converge to different local minima M times, allows a substantial reduction of training time. For each minima, the parameters of this estimator are recorded, acting as the m^{th} base model in the ensemble. The output of *snapshot ensemble* then takes the average over the predictions from all snapshots.

To obtain base models with good performance, *snapshot ensemble* uses a cyclic annealing schedule on the learning rate. Suppose that the initial learning rate is α_0, the total number of training iterations is K, the learning rate at iteration k is set to:

$$\alpha_k = \frac{\alpha_0}{2}\left(\cos\left(\pi \frac{(k-1) \mod \lceil K/M \rceil}{\lceil K/M \rceil}\right) + 1\right). \quad (14)$$

The global deep learning procedure including *snapshot ensemble* is summarized in Algorithms 1-2:

- Algorithm 1 gives an overview of a base model training, where the mappings T, $\mathcal{T}$ and $\widetilde{\mathcal{T}}$ are trained sequentially. For each one, the learning rate used in the gradient descent is gradually reduced following (14). Starting from a (relatively) high value prevents the base model parameters from being trapped in a local minimum.
- Algorithm 2 depicts the ensemble procedure. Each call of Algorithm 1 is done with an optimized version of the mappings parameters, allowing to reduce the number of iterations (i.e. reduce K_T, $K_{\mathcal{T}}$, $K_{\widetilde{\mathcal{T}}}$) and therefore to reduce the computation time. As shown in [12], diversity in the data helps to improve the ensemble performance; we take advantage of the simulation model to sample a new dataset for each base model.

IV. Simulation Results

In this section we evaluate our approach through numerical simulations applied to two different partial differential equations (PDEs). A multi-input multi-output example (a counter-flow heat exchanger) and the Korteweg-de Vries equation are addressed.

In the rest of the section, we denote with $t \in \mathbb{R}$ the time variable and with $s \in [0, 1]$ the spatial variable. Given a function $w(t, s)$ we denote with $w_t := \partial_t w$, respectively w_s, its partial derivative with respect to t, respectively s.

Algorithm 1: Training a base model.

Input : Initial mappings parameters Θ_T, $\Theta_{\mathcal{T}}$ and $\Theta_{\widetilde{\mathcal{T}}}$. Observer matrices A and B. Datasets D_x and D_z.
Output: Learned mappings parameters Θ_T, $\Theta_{\mathcal{T}}$, $\Theta_{\widetilde{\mathcal{T}}}$.

```
// Learning T
for k = 1, ..., K_T do
    Compute L_T following (11) for a subset (a batch) of D_x.
    Update Θ_T using stochastic gradient descent (SGD) to minimize L_T.
    Reduce the learning rate following (14).
end
// Learning T̃
for k = 1, ..., K_T̃ do
    Compute L_T̃ following (12) for a batch of D_x.
    Update Θ_T̃ using SGD to minimize L_T̃.
    Reduce the learning rate following (14).
end
// Learning T̃̃
for k = 1, ..., K_T̃̃ do
    Compute L_T̃̃ following (13) for a batch of D_z.
    Update Θ_T̃̃ using SGD to minimize L_T̃̃.
    Reduce the learning rate following (14).
end
```

Algorithm 2: Training an ensemble.

Design Parameters: Matrices $\{A^m, B^m, m = 1, ..., M\}$

```
Initialize parameters Θ_T, Θ_T, Θ_T̃ randomly.
for m = 1, ..., M do
    Collect a dataset D_x = {x_k, x_{k+1}, y_k} as detailed in Section III-A.
    Collect a dataset D_z = {z_k} as detailed in Section III-B.
    Call algorithm 1 to update parameters Θ_T, Θ_T, Θ_T̃. Save these parameters as the base model m.
end
```

A. Heat exchanger

In this subsection, we consider a counter-current heat exchanger (HEX) where two liquid streams exchange heat. When no phase change occurs and under standard physical assumptions, a simple PDE model is given by

$$\begin{aligned} &T_t = -v_h T_s - \kappa_h (T - \overline{T}) \\ &\overline{T}_t = v_c \overline{T}_s - \kappa_c (\overline{T} - T) \\ &T(t, 0) = T_{in}, \quad \overline{T}(t, 1) = \overline{T}_{in}, \\ &T(0, s) = T_0(s), \quad \overline{T}(0, s) = \overline{T}_0(s), \end{aligned} \quad (15)$$

where T denotes the temperature of the hot fluid and $\overline{T}$ the temperature of the cold one, and $v_c, v_h, \kappa_c, \kappa_h$ are some positive parameters. We suppose to measure the output temperatures of the fluids, i.e. $T(1, t)$ and $\overline{T}(0, t)$. We refer to [14] and references therein for more details about the

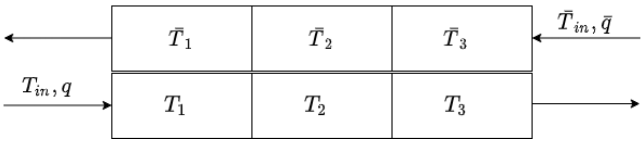


Fig. 1: Counter-current exchanger with inlet and outlet heat flux.

PDE model (15) and its observability properties where an infinite-dimensional observer is proposed. Here we follow a different route and in particular we follow the methodology proposed in Sections II and III. To this end, we first need to discretize the PDE model (15) both in space and time. For the spatial discretization, we follow the representation proposed in [15], in which the heat exchanger is represented as a three homogeneous-cell model, see Figure 1. It is supposed that the temperature fluids are controlled via the mass flow rates q and $\bar{q}$. This representation gives a model which is described by a bilinear ODE. Such an ODE is then discretized in time using an Euler scheme with a time step $\delta_t = 1s$. By using the compact notations $x = (T_1, T_2, T_3, \bar{T}_1, \bar{T}_2, \bar{T}_3)^T \in \mathbb{R}^6$ and $u = (T_{in}, \bar{T}_{in}, q, \bar{q}) \in \mathbb{R}^4$, we obtain the following multi-variable discrete-time model with four inputs and two outputs

$$\begin{cases} x_{k+1} &= x_k + \delta_t\big((e_1 u_{k,1} + E x_k)u_{k,3} + \\ & \quad \bar{e}_1 u_{k,2} + \bar{E} x_k)u_{k,4} + Fx\big) \\ y_k &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} x_k, \end{cases} \tag{16}$$

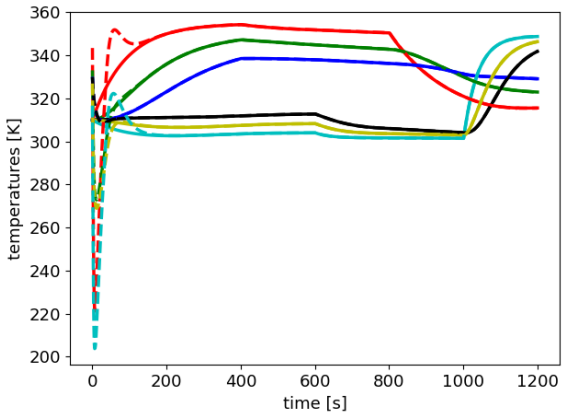
where E, F, $\bar{E}$, e_1 and $\bar{e}_1$ are matrices and vectors that depends on the physical parameters. In the simulations these parameters are selected according to [15, Section 5].

In all simulations, we consider that the inlet temperatures T_{in} and $\hat{T}_{in}$ remain between $280\,K$ and $360\,K$. Hence, from physical consideration, all the state values initialized in $[280, 360]$ remain in this range and we can define $\mathcal{X} = [280, 360]^6$. We also consider inlet mass flows in $]0; 0.05]$.

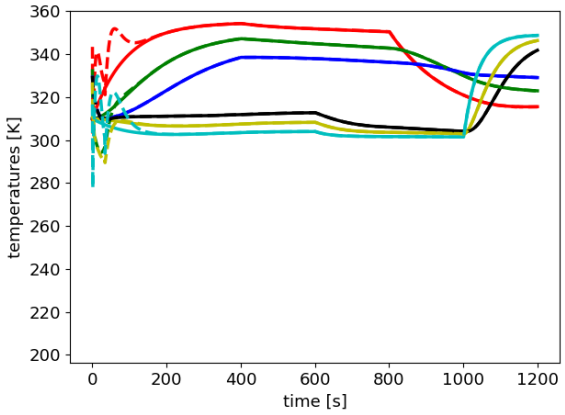
First, we apply the baseline methodology from [9] recalled in Section III-A. Diagonal elements of the observer matrix A are chosen equally spaced in $[0.85, 0.95]$. A result for varying inputs is shown in Figure 2a. Whereas it exhibits good asymptotic performance, a peaking phenomenon is observed during the transient phase.

We extend this baseline as detailed in Section III-B. The switching threshold is fixed to $\epsilon_{\mathcal{T}} = 10^{-4}$. Figure 2b illustrates the improvement for the same initial conditions: during the transient phase the peaking is limited and the estimation accuracy is improved. Figure 3 plots the reconstruction error $|\hat{z} - T(\mathcal{T}(\hat{z}))|$ used to detect if $\hat{z} \in T(\mathcal{X})$.

The state estimation accuracy is further evaluated on a batch of 100 random initial conditions with random values for the inputs. To assess the benefit of using an ensemble model, the method detailed in Section III-C is applied with 5 base estimators ($M = 5$). Results reported in Figure 4 show significant improvements in term of asymptotic accuracy.



(a) Baseline method from [9] (solid lines for x, dash lines for the estimation $\hat{x}$).



(b) Benefits of the extension of $\widetilde{\mathcal{T}}$ during the transient phase (solid lines for x, dash lines for the estimation $\hat{x}$).

Fig. 2: Heat exchanger: a simulation result for varying inputs.

B. A nonlinear observer for the Korteweg-de Vries model

The Korteweg-de Vries (KdV) equation is a nonlinear PDE describing the behavior of long waves in a water channel. To the best of our knowledge, although being a well-known system, only linear local observers exist in the literature [16]. The studied system can be written as follows:

$$\begin{cases} w_t + w_x + w_{xxx} + w w_x = 0 \\ w(t,0) = u(t),\ w_x(t,L) = 0,\ w_{xx}(t,L) = 0 \\ w(0,x) = w_0(x) \\ y(t) = w(t,L). \end{cases}$$

In order to discretize the KdV equation, we use the finite difference scheme detailed in [16, Section 7]. We choose $d_x = 35$ points for the spatial discretization on the interval $[0, L]$ with $L = 2\pi$ and a time step of $0.01\,s$. This spatial discretization was found to be a good trade-off to obtain an accurate model while keeping a tractable state dimension. An example of trajectory is shown in Figure 5.

In order to obtain a meaningful exploration of the state space during the learning procedure, the state space is sampled considering the KdV model (IV-B). State values are

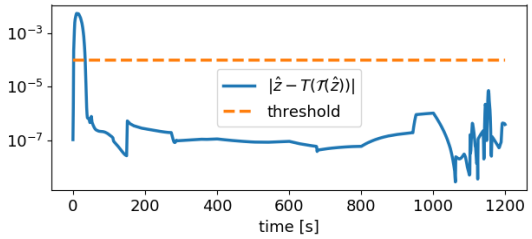


Fig. 3: Heat exchanger - switching condition for extension: a simulation result for varying inputs.

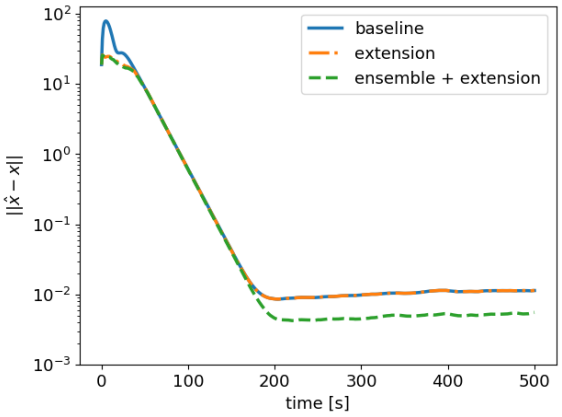


Fig. 4: Heat exchanger - Mean error on 100 trajectories.

sampled from $w(t,x) = a\,\sin(b\,x + c)$ with a, b and c being randomly picked in $[-1.5, 1.5]$ and $[0, 2\pi]$. We follow the methodology proposed in Sections II and III with 5 base estimators and a switching threshold fixed to $\epsilon_{\mathcal{T}} = 10^{-4}$. The simulation results illustrated in Figure 6 show a good convergence of the state estimation in the autonomous case.

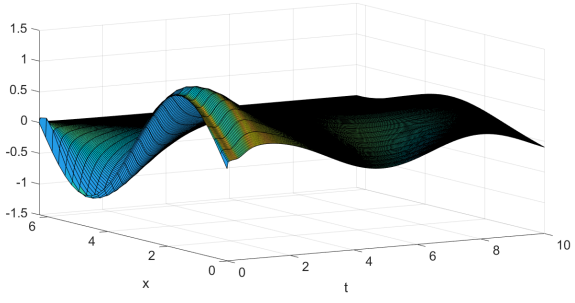


Fig. 5: KdV equation - An example of trajectory.

V. Conclusion

In this work we presented a approach to design an observer for nonlinear non-autonomous discrete-time systems. The new observer consists in a combination of the KKL approach and a deep neural networks, to construct such a transformation. In this work we gave an extension and relax the conditions on the previous version of the observer. We also assess the proposed method on large dimensional nonlinear MIMO systems where the explicit solution is

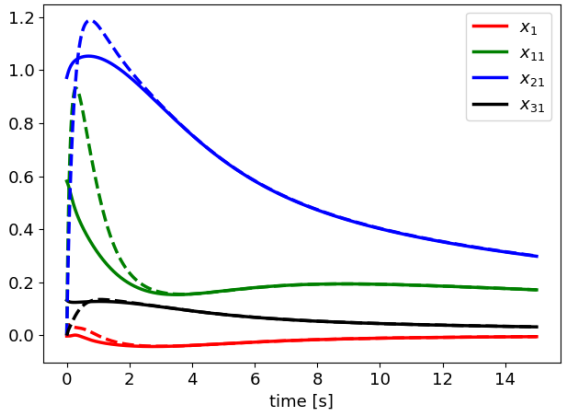


Fig. 6: KdV equation - an estimation result. (solid lines for x, dash lines for the estimation $\hat{x}$)

practically impossible to obtain. The simulation results show an interesting potential of the use of such an approach in industrial applications.

References

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

[2] L. Brivadis, V. Andrieu, and U. Serres, "Luenberger observers for discrete-time nonlinear systems," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019.

[3] A. Alessandri and M. Awawdeh, "Moving-horizon estimation with guaranteed robustness for discrete-time linear systems and measurements subject to outliers," *Automatica*, 2016.

[4] C. Califano, S. Monaco, and D. Normand-Cyrot, "Canonical observer forms for multi-output systems up to coordinate and output transformations in discrete time," *Automatica*, 2009.

[5] N. Kazantzis and C. Kravaris, "Discrete-time nonlinear observer design using functional equations," *Systems & Control Letters*, 2001.

[6] P. Bernard and V. Andrieu, "Luenberger observers for nonautonomous nonlinear systems," *IEEE Transactions on Automatic Control*, 2018.

[7] F. Poulain, L. Praly, and R. Ortega, "An observer for permanent magnet synchronous motors with currents and voltages as only measurements," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008.

[8] 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)*. IEEE, 2020.

[9] 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)*. IEEE, 2021.

[10] K. Esfandiari and M. Shakarami, "Bank of high-gain observers in output feedback control: Robustness analysis against measurement noise," *IEEE Transactions on Systems, Man, and Cybernetics*, 2019.

[11] M. Shakarami, K. Esfandiari, A. A. Suratgar, and H. A. Talebi, "On the peaking attenuation and transient response improvement of high-gain observers," in *Conference on Decision and Control (CDC)*, 2018.

[12] M. Ganaie, M. Hu *et al.*, "Ensemble deep learning: A review," *arXiv preprint arXiv:2104.02395*, 2021.

[13] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," *arXiv preprint arXiv:1704.00109*, 2017.

[14] F. Zobiri, E. Witrant, and F. Bonne, "Pde observer design for counter-current heat flows in a heat-exchanger," *IFAC-PapersOnLine*, 2017.

[15] B. Zitte, B. Hamroun, D. Astolfi, and F. Couenne, "Robust control of a class of bilinear systems by forwarding: Application to counter current heat exchanger," *IFAC-PapersOnLine*, 2020.

[16] S. Marx and E. Cerpa, "Output feedback stabilization of the korteweg–de vries equation," *Automatica*, 2018.