IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Indirect Data-Driven Observer Design Using Neural Canonical Observer Structures

Lukas Ecker[1] and Markus Schöberl[1]

Abstract— **An indirect data-driven observer design approach for nonlinear discrete-time systems based on an input-output injection with neural canonical observer structures is proposed. An artificial neural network auto-encoder structure, trained with recorded state, input, and output data, is used for the identification of a system in a nonlinear Brunovsky observer form with output transformation. The neural approximations of the transformations and the input-output injection can be used to construct an observer with linear error dynamics using methods from linear control theory. The approach is demonstrated on two academic examples and on an industrially-motivated problem with a sampled continuous-time model.**

I. Introduction

The design of observers for the online estimation of state variables is a fundamental aspect of control theory. With the advancement of modern state-space control strategies, see, e.g., [1], the interest in novel observer design approaches, especially for nonlinear systems, has also increased. The design strategies can be distinguished in terms of their basic concepts and ideas, including for instance stochastic, optimization-based, system-theoretical, and differential-geometrical approaches. Among the most popular nonlinear observers are the extensions of the linear Kalman filter, such as the extended and unscented Kalman filter. The Bayesian state observer, as discussed in [2], is a more general and mathematically rigorous state estimator, but can often be computationally intractable. Particle filters, see, e.g., [3], can be considered as approximations to the Bayesian state observer to reduce the computational effort. Moving horizon estimators are optimization-based approaches that generate state estimates by incorporating a series of measurements over a finite horizon, see [4], [5]. System-theoretical approaches, such as energy- and passivity-based observer designs, have been pursued in, e.g., [6], [7], [8], [9].

A differential geometrical nonlinear observer design approach, c.f., [10], which utilizes a canonical system representation with output injection to easily derive a Luenberger-like observer with linear error dynamics, was first proposed by [11] and [12]. What was initially limited in [11] to single output autonomous systems has been extended in [13], [14] to the multi-input multi-output case for nonlinear continuous systems. Extensions of this idea to the discrete-time scenario were presented in, e.g., [15], [16], [17]. Recently, the observer error linearization problem has also been considered in terms of a dynamical extension [18], [19]. However, the necessary and sufficient conditions for the equivalence of a system to a canonical observer form require knowledge of the mathematical model, sophisticated mathematical tools, and often the solution of a set of partial differential equations.

The underlying idea of the data-driven approach proposed in this paper is to identify a system with recorded sensor data in a nonlinear canonical Brunovsky observer form. The necessary state and output transformations are implemented as neural network auto-encoder structures and the input-output injection as separate artificial neural network. With the system represented in this canonical observer form and the transformations approximated by trained neural networks, a Luenberger-like state observer with a linear error dynamic can be designed. It should be noted, that the networks are trained with recordings from the input and output of the system as well as full-state measurements. The final designed observer will rely only on the input and output. The requirement of full-state measurements is in many cases not a hard limitation. For instance, many industrial systems need to be equipped with additional sensors during commissioning in order to identify required model parameters anyway. Furthermore, the proposed data-driven approach intends to overcome the mathematical burden of analytically deriving the necessary transformations, which makes the approach also interesting for complex systems with known governing equations, where full-state recordings can be generated in simulations. A data-driven controller design approach using a neural auto-encoder structure to approximate a feedback linearization for single-input single-output systems has been proposed by the authors in [20]. The present paper considers the observer design for multi-input multi-output systems by approximating a nonlinear Brunovsky observer form with a neural auto-encoder structure.

The paper is organized as follows: Section II recapitulates the basic ideas of representing a nonlinear system with state and output transformations in a nonlinear Brunovsky observer form with input-output injection. Section III addresses the embedding of the system representation in a data-driven framework with the transformations approximated by neural networks. Section IV outlines the subsequent observer design with neural observer structures, whereas Section V demonstrates the capabilities of the method on three examples.

*This work has been supported by the COMET-K2 Center of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.

[1] Lukas Ecker and Markus Schöberl are with Institute of Automatic Control and Control Systems Technology, Johannes Kepler University Linz, Altenberger Str. 69, A-4040 Linz, Austria (e-mail: {lukas.ecker,markus.schoeberl}@jku.at)

II. Nonlinear Brunovsky Observer Form

The addressed data-driven scenario assumes that the governing equations of the considered system are not known. The basic idea is to approximate a model of the system in Brunovsky observer form from collected data sets $\mathcal{X} = \{x_i\}_{i\in\mathcal{Q}}$, $\mathcal{X}^+ = \{x_i^+\}_{i\in\mathcal{Q}}$, $\mathcal{U} = \{u_i\}_{i\in\mathcal{Q}}$, $\mathcal{Y} = \{y_i\}_{i\in\mathcal{Q}}$, of state $x_k = [x_{1,k}, \dots, x_{n_x,k}]^T \in \mathcal{D}_x \subset \mathbb{R}^{n_x}$, successor state $x_k^+ = [x_{1,k}^+, \dots, x_{n_x,k}^+]^T \in \mathcal{D}_x \subset \mathbb{R}^{n_x}$, input $u_k = [u_{1,k}, \dots, u_{n_u,k}]^T \in \mathcal{D}_u \subset \mathbb{R}^{n_u}$ and output $y_k = [y_{1,k}, \dots, y_{n_y,k}]^T \in \mathcal{D}_y \subset \mathbb{R}^{n_y}$. Here, k denotes the time index, i.e., $x_k = x(kT_s)$, with sampling time T_s, and x_k^+ refers to the successor state x_{k+1}. Further, the set of sampled time indices is denoted by $\mathcal{Q} \in \mathbb{N}_0, |\mathcal{Q}| = n_s$. The intention is to represent an unknown multi-input multi-output nonlinear discrete-time system

$$x_{k+1} = f(x_k, u_k), \quad y_k = h(x_k), \tag{1}$$

by means of a smooth state transformation $z_k = \Phi(x_k) : \mathcal{D}_x \to \Phi(\mathcal{D}_x)$ and output transformation $\bar{y}_k = \Xi(y_k) : \mathcal{D}_y \to \Xi(\mathcal{D}_y)$ in the nonlinear Brunovsky observer form [17]

$$\begin{aligned} z_{k+1} &= \Phi \circ f(\Phi^{-1}(z_k), u_k) = A_B z_k + \Psi(u_k, \bar{y}_k), \\ \bar{y}_k &= \Xi \circ h(\Phi^{-1}(z_k)) = C_B z_k, \end{aligned} \tag{2}$$

where $\Psi(u_k, \bar{y}_k) : \mathcal{D}_u \times \Xi(\mathcal{D}_y) \to \Psi(\mathcal{D}_u, \Xi(\mathcal{D}_y)) \subset \mathbb{R}^{n_x}$ is a smooth input-output injection. The pair (A_B, C_B) is assumed to be in canonical Brunovsky observer form with $A_B = \operatorname{diag}(A_{B,1}, \dots, A_{B,n_y}) \in \mathbb{R}^{n_x\times n_x}$ and $C_B = \operatorname{diag}(c_{B,1}^T, \dots, c_{B,n_y}^T) \in \mathbb{R}^{n_y\times n_x}$ defined by the matrices

$$A_{B,i} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{\lambda_i\times\lambda_i},\ 1 \le i \le n_y$$

$$c_{B,i}^T = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{1\times\lambda_i},\ 1 \le i \le n_y.$$

The integers $(\lambda_1, \dots, \lambda_{n_y})$ with $\sum_{i=1}^{n_y} \lambda_i = n_x$ correspond to the observability indices, c.f., [15], [17], with λ_i defined as the least nonnegative integer such that the covector $\mathrm{d}(h_i \circ f^{\lambda_i})$ is included in the codistribution $\operatorname{span}(\{\mathrm{d}(h_j \circ f^{\lambda_s})\}_{1\le j\le n_y, 0\le s<\lambda_i}) \oplus \operatorname{span}(\{\mathrm{d}(h_l \circ f^{\lambda_i})\}_{1\le l<i}) \oplus \operatorname{span}(\{\mathrm{d}u_{j,k+s}\}_{1\le j\le n_u, 0\le s<n_x})$. The operator d denotes the exterior derivative with respect to the coordinates $x_k, u_k, \dots u_{k+n_x-1}$ and f^i the composition defined by

$$\begin{aligned} f^0(x_k, u_k) &= x_k, \quad f^1(x_k, u_k) = f(x_k, u_k), \\ f^i(x_k, u_k) &= f(f^{i-1}(x_k, u_k), u_{k+i}), i \ge 2. \end{aligned}$$

Necessary and sufficient conditions for the existence and construction of the transformations in nonlinear Brunovsky observer form for different system classes can be found in [10], [13], [14], [15], [16], [17]. The benefits of a representation in Brunovsky observer form for an observer design will be discussed later in Section IV. However, without any knowledge of the mathematical model (1), the intention, is to identify the transformations from data. For this purpose, the model (1) is considered in a representation composed of the components of the nonlinear Brunovsky observer form (2). As the state and output transformation Φ and Ξ are assumed to be diffeomorphisms, the desired representation in terms of the previously defined Brunovsky structure is given according to

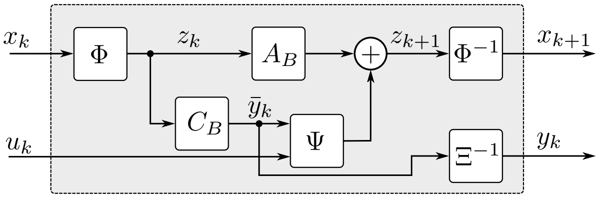

Fig. 1. Representation of the nonlinear discrete-time system (1) in terms of the components of the nonlinear Brunovsky observer form.

$$\begin{aligned} x_{k+1} &= \Phi^{-1} \circ (A_B\Phi(x_k) + \Psi(u_k, C_B\Phi(x_k))), \\ y_k &= \Xi^{-1} \circ C_B\Phi(x_k). \end{aligned} \tag{3}$$

This representation is schematically illustrated in Fig. 1.

III. Neural Observer Structure

The basic idea of the approach proposed in this paper is to circumvent the tedious mathematical derivation of the transformations Φ, Ξ and Ψ by using historically recorded state, input and output trajectories to identify the system (3). The unknown transformations are implemented as artificial neural networks $\hat{\Phi}$, $\hat{\Xi}$ and $\hat{\Psi}$, which are trained on the collected data sets. Note that the approach is in principle not limited to an approximation with artificial neural networks and that other parametric approaches such as radial basis functions or other network topologies, e.g. residual neural networks, are also possible. The matrices (A_B, C_B) can be uniquely determined based on the observability indices $(\lambda_1, \dots, \lambda_{n_y})$. Since the system is not known in the considered data-driven framework, the observability indices can not be determined analytically. Therefore, the indices are treated as hyperparameters and are generally determined by validating all possible combinations in several trainings. Consequently, the question of whether the necessary transformations exist at all and whether the observer design is possible can only be answered after the training process. The transformations for the representation of the system in nonlinear Brunovsky observer form should be solutions of the optimization problem

$$\begin{aligned} &\min_{\Phi,\Xi,\Psi} \|x^+ - \Phi^{-1} \circ (A_B\Phi(x) + \Psi(u, C_B\Phi(x)))\| \\ &\text{s.t.} \quad y = \Xi^{-1} \circ C_B\Phi(x), \quad x \in \mathcal{D}_x, \quad u \in \mathcal{D}_u, \end{aligned}$$

where x^+ corresponds to successor state and y to the output according to (1), such that an identification of the system (3) is obtained. The accuracy of the identified model determines the validity of the transformations. Consequently, a high model accuracy is necessary to design an observer and to achieve high observer performance, since the state $\hat{\Phi}$ and output transformation $\hat{\Xi}$ as well as the input-output injection $\hat{\Psi}$ will be used for the state estimation. Besides the accuracy of the model, the invertibility of the state and output transformation is a crucial factor. The approach followed in this work is to realize the transformations as artificial feed-forward neural networks $\hat{\Phi}$, $\hat{\Xi}$, and $\hat{\Psi}$, with the necessary inversions $\hat{\Phi}^{-1}$ and $\hat{\Xi}^{-1}$ implemented as separate networks. The employment of analytically invertible neural networks would restrict the approach to invertible activation functions and flat network structures with constant number of units in each layer. Although this is not inconsistent with the proposed assumptions, a planned extension of the observer design approach with immersion techniques and dynamic extension where $\dim(z_k) > \dim(x_k)$ would not be feasible. Therefore, an implementation with separate networks was preferred. A comparable discussion was presented in [20]. Upon examining the structure of the model shown in Fig. 1, it becomes evident that the identification problem is comparable to learning an auto-encoder structure. With the encoders being $\hat{\Phi}$ and $\hat{\Xi}$ and the decoders being $\hat{\Phi}^{-1}$ and $\hat{\Xi}^{-1}$, the encoded Brunovsky states undergo a shift through the canonical system in nonlinear Brunovsky observer form (2) and are subsequently decoded back to their original coordinates. In addition to the prediction accuracy of the neural observer structure, the reconstruction performance of the two transformations $x = \Phi^{-1} \circ \Phi(x)$ and $y = \Xi^{-1} \circ \Xi(y)$ is crucial. Therefore penalty terms enforcing these invertibility constraints are included to the cost function, along with the prediction loss. This approach is often used in auto-encoders, as demonstrated in, e.g. , [20], [21]. The considered feed-forward neural networks for the auto-encoder structure and the approximation of the input-output injection are implemented with n_L layers and n_U hidden layer units as

$$\begin{aligned}\hat{\Phi}(x_k; W^i_{\hat{\Phi}}, b^i_{\hat{\Phi}}) &= W^{n_L}_{\hat{\Phi}}(\bar{\phi}^{n_L-1}_{\hat{\Phi}} \circ \cdots \circ \bar{\phi}^1_{\hat{\Phi}}(x_k)) + b^{n_L}_{\hat{\Phi}},\\ \hat{\Xi}(y_k; W^i_{\hat{\Xi}}, b^i_{\hat{\Xi}}) &= W^{n_L}_{\hat{\Xi}}(\bar{\phi}^{n_L-1}_{\hat{\Xi}} \circ \cdots \circ \bar{\phi}^1_{\hat{\Xi}}(y_k)) + b^{n_L}_{\hat{\Xi}},\\ \hat{\Psi}(u_k, \bar{y}_k; W^i_{\hat{\Psi}}, b^i_{\hat{\Psi}}) &= W^{n_L}_{\hat{\Psi}}(\bar{\phi}^{n_L-1}_{\hat{\Psi}} \circ \cdots \circ \bar{\phi}^1_{\hat{\Psi}}(u_k, \bar{y}_k)) + b^{n_L}_{\hat{\Psi}}\end{aligned}$$

with $\bar{\phi}^i_{\hat{\Delta}}(a_k) = \phi(W^i_{\hat{\Delta}} a_k + b^i_{\hat{\Delta}})$, $\hat{\Delta} \in \{\hat{\Phi}, \hat{\Xi}, \hat{\Psi}\}$, where ϕ corresponds to a nonlinear activation function $\phi(s) = [\phi_1(s_1), \ldots, \phi_{n_U}(s_{n_U})]^T \in \mathbb{R}^{n_U}, s_i \in \mathbb{R}$ and $W^i_{\hat{\Delta}}, b^i_{\hat{\Delta}}$ to the neural network parameters $W^i_{\hat{\Delta}} \in \mathbb{R}^{n_U \times n_U}, b^i_{\hat{\Delta}} \in \mathbb{R}^{n_U}$ for $1 < i < n_L$, $W^1_{\hat{\Phi}} \in \mathbb{R}^{n_U \times n_x}, W^{n_L}_{\hat{\Phi}} \in \mathbb{R}^{n_x \times n_U}, b^1_{\hat{\Phi}} \in \mathbb{R}^{n_U}, b^{n_L}_{\hat{\Phi}} \in \mathbb{R}^{n_x}$, $W^1_{\hat{\Xi}} \in \mathbb{R}^{n_U \times n_y}, W^{n_L}_{\hat{\Xi}} \in \mathbb{R}^{n_y \times n_U}, b^1_{\hat{\Xi}} \in \mathbb{R}^{n_U}, b^{n_L}_{\hat{\Xi}} \in \mathbb{R}^{n_y}$ and $W^1_{\hat{\Psi}} \in \mathbb{R}^{n_U \times (n_u + n_y)}, W^{n_L}_{\hat{\Psi}} \in \mathbb{R}^{n_x \times n_U}, b^1_{\hat{\Psi}} \in \mathbb{R}^{n_U}, b^{n_L}_{\hat{\Psi}} \in \mathbb{R}^{n_x}$. The topologies of the single transformations can of course be defined independently of each other. The network weights of the neural observer structure are determined in a training process by means of a gradient decent algorithm with back propagation. The utilized cost function

$$L = \underbrace{\alpha_1 L_{\text{pred},x} + \alpha_2 L_{\text{pred},y}}_{L_{\text{pred}}} + \underbrace{\alpha_3 L_{\text{rec},x} + \alpha_4 L_{\text{rec},y}}_{L_{\text{rec}}},$$

required for the training of the neural auto-encoders and the input-output injection, is composed of two components summarizing the prediction error L_{pred} and the reconstruction error L_{rec}. Scaled with the hyperparameters $\alpha_i \in \mathbb{R}_{>0}, 1 \leq i \leq 4$, the individual loss terms can be synthesized as

$$L_{\text{pred},x} = ||x_i^+ - \Phi^{-1} \circ \sigma(\Phi(x_i), u_i, C_B\Phi(x_i))||_{\text{MSE}}, \quad (4)$$

$$L_{\text{pred},y} = ||y_i - \Xi^{-1} \circ C_B\Phi(x_i)||_{\text{MSE}}, \quad (5)$$

$$L_{\text{rec},x} = ||x_i - \Phi^{-1} \circ \Phi(x_i)||_{\text{MSE}}, \quad (6)$$

$$L_{\text{rec},y} = ||y_i - \Xi^{-1} \circ \Xi(y_i))||_{\text{MSE}}, \quad (7)$$

with the state and output prediction error in (4) - (5) and the reconstruction error (6) - (7), respectively. The abbreviation in (4) is defined as $\sigma(z_k, u_k, \bar{y}_k) = A_B z_k + \Psi(u_k, \bar{y}_k)$ and the utilized loss function norm corresponds to

$$|| \cdot ||_{\text{MSE}} = \frac{1}{n_s} \sum_{i \in \mathcal{Q}_B} || \cdot ||_2^2. \quad (8)$$

with $\mathcal{Q}_B \in \mathcal{Q}$ denoting a batch of sampling time indices.

IV. Observer and Error Dynamics

The system (1) data-driven identified in nonlinear canonical observer form (2) as presented in Section III, allows to trivially derive an observer as a copy of the system imposed with an additional output error-term as

$$\begin{aligned}\hat{z}_{k+1} &= A_B \hat{z}_k + \hat{\Psi}(u_k, \bar{y}_k) + K(\bar{y}_k - \hat{\bar{y}}_k),\\ \hat{x}_k &= \hat{\Phi}^{-1}(\hat{z}_k), \quad \bar{y}_k = \hat{\Xi}(y_k),\end{aligned} \quad (9)$$

where $\hat{\bar{y}}_k = C_B \hat{z}_k$ corresponds to the estimated output of the observer and $\hat{x}_k$ to the state estimation of x_k. In original coordinates the approach results in the nonlinear observer with dynamic $\hat{x}_{k+1} = \hat{\Phi}^{-1} \circ (A_B \hat{\Phi}(\hat{x}_k) + \hat{\Psi}(u_k, \hat{\Xi}(y_k)) + K\hat{\Xi}(y_k - \hat{\Xi}^{-1}(C_B \hat{\Phi}(\hat{x}_k))$. The input-output injection and the state and output transformation are mapped by the previously trained neural auto-encoder structures $\hat{\Phi}$, $\hat{\Xi}$, and the feed-forward neural network $\hat{\Psi}$. The dynamics of the linear observer error $e_k = z_k - \hat{z}_k$ results in the error system $e_{k+1} = (A_B + KC_B)\, e_k$. Since the pair (A_B, C_B) is fully observable and therefore detectable, an asymptotically stable error dynamics can be obtained with a properly determined observer gain $K = \text{diag}(K_1, \ldots, K_{n_y}) \in \mathbb{R}^{n_x \times n_y}$ consisting of the coefficients of the desired characteristic polynomials $K_i = [a_{i,1}, \ldots, a_{i,\lambda_i}]^T \in \mathbb{R}^{\lambda_i}, a_{i,j} \in \mathbb{R}$ of each chain λ_i.

V. Examples

The proposed indirect observer design approach is now demonstrated in two academic examples and in an industrially motivated application. The neural network structures were implemented using the PyTorch framework in Python, and the training process utilized the Adam optimizer. The learning rate of the Adam optimizer is increased linearly during an initial warm-up phase until a specified rate was reached, after which a learning rate scheduler is applied for annealing. It should be noted, that no comprehensive tuning of the hyperparameters, except for the observability indices, has been pursued. The main focus is to demonstrate the principle capabilities of the proposed approach.

A. Academic Examples

1) Example I: The following example with additional additive control input is adapted from [17]. As discussed in the reference, the system is not transformable to nonlinear Brunovsky observer form without output transformation. The two-dimensional nonlinear system with two outputs

$$f = \begin{bmatrix} x_{2,k} \\ x_{1,k} + x_{1,k}x_{3,k} + u_{1,k} \\ x_{2,k} + x_{3,k} \end{bmatrix}, \quad h = \begin{bmatrix} x_{1,k} \\ x_{3,k} \end{bmatrix}, \qquad (10)$$

is considered with the observability indices $(\lambda_1, \lambda_2) = (2, 1)$. The training set for the data-driven approach originates from $n_s = 5000$ uniform randomly drawn samples in the state space $\mathcal{D}_x = [-1, 1]^3 \subset \mathbb{R}^3$ and input space $\mathcal{D}_u = [-1, 1] \subset \mathbb{R}$. The successor states and outputs have been computed according to (10). In addition, separate initial value simulations with random input signals serve as trajectories for the validation process. The neural approximations $\hat{\Phi}$, $\hat{\Xi}$, $\hat{\Psi}$ were implemented as standard feed-forward neural networks with one $n_L = 1$ hidden layer, $n_U = 100$ hidden units and ReLU activation layer. The neural auto-encoder architecture underwent 1000 epochs of training. After the warm-up phase, the model was validated in every epoch by comparing the validation trajectory with the simulated state trajectory of the model. To conduct the simulation and to solve the initial value problem, the initial state and the input signal were inherited from the validation trajectory. The model with the lowest mean-square error (8) between validation and simulation trajectory was chosen as final model. The illustrations of this example in Fig. 2 are focused on the prediction performance of the neural observer structure and the reconstruction performance of the state transformation evaluated on the validation trajectory. As the figure reveals, it is possible to achieve very high prediction and reconstruction accuracy with the neural structure. Although an observer was successfully developed for this example as well, due to space limitations, only the later examples specifically address the observer and its estimation performance.

2) Example II: The second example considers a six dimensional system ($n_x = 6$) with two inputs and outputs ($n_u = n_y = 2$). The nonlinear discrete-time system equation $x_{k+1} = f(x_k, u_k)$ and output equation $y_k = h(x_k)$ are defined according to

$$f = \begin{bmatrix} 10\,\gamma_1(x_k) + u_{1_k} \\ 10\,\gamma_2(x_k) + x_{1,k} + 2\,\frac{\gamma_1(x_k)\psi_4(x_k)}{a_1 b_4} \\ \gamma_3(x_k) + \frac{x_{2,k}}{10} + (\gamma_1(x_k) + \frac{u_{1,k}}{10})(10\,\gamma_3(x_k) + x_{2,k}) \\ \gamma_2(x_k) + \psi_4(x_k) + u_{2,k} + \frac{x_{1,k}}{10} + \frac{1}{5}\frac{\gamma_2(x_k)\psi_5(x_k)}{a_2 b_5} \\ \psi_5(x_k) - \frac{x_{2,k}}{10} + x_{4,k} + \frac{1}{5}\sqrt{x_{6,k} - \frac{x_{1,k}}{10} + 1} \\ \gamma_1(x_k) + \psi_6(x_k) + \frac{u_{1,k}}{10} + x_{5,k} \end{bmatrix},$$

$$h = \begin{bmatrix} 2\,\frac{x_{3,k}}{x_{1,k}+1} \\ -\frac{x_{1,k}}{100} + \frac{1}{5}\frac{x_{3,k}}{x_{1,k}+1} + \frac{x_{6,k}}{10} \end{bmatrix},$$

with abbreviations $\gamma_i(x_k) = a_i \frac{x_{3,k}}{x_{1,k}+1}$ and $\psi_i(x_k) = b_i(x_{6,k} - \frac{x_{1,k}}{10})$ and coefficients $a_1 = a_2 = 0.2\,, a_3 = 0.3\,, b_4 = b_5 = b_6 = 0.05\,.$ The training set consists of $n_s = 20000$ uniform randomly drawn samples in the state space $\mathcal{D}_x = [0, 1]^6 \subset \mathbb{R}^6$ and input space $\mathcal{D}_u = [0, 1]^2 \subset \mathbb{R}^2$. The auto-encoder structures $\hat{\Phi}, \hat{\Xi}$ as well as the input-output injection $\hat{\Psi}$ were implemented as feed-forward neural networks with two hidden layers $n_L = 2$, constant number of hidden units $n_U = 75$, and ReLU activation function. The observability indices (λ_1, λ_2) were assumed not to be known a priori and therefore considered as hyperparameters. The neural canonical observer structure was trained with all possible combinations of observability indices in two scenarios, i.e., once with and once without output transformation. The best pair and final model was determined according to the lowest mean-square error between the predictions of the trained model and a validation trajectory, as in Example I. The final results are visualized in Table 1. It can be observed that the model with index pair (3,3) and trained output transformation yielded the best performance, which is in accordance with the analytical result. All other models failed to achieve a reasonable accuracy, and the observer designs were also unsuccessful. Consequently, the model with indices (3,3) and considered output transformation has been successfully used for the observer design discussed in Section IV. The observer gain coefficients in K_1 and K_2 have been determined by pole placement with the polynomial roots resulting from the poles [0.14, 0.32, 0.43] and [0.40, 0.30, 0.20], respectively. The performance of the resulting state estimator on a validation

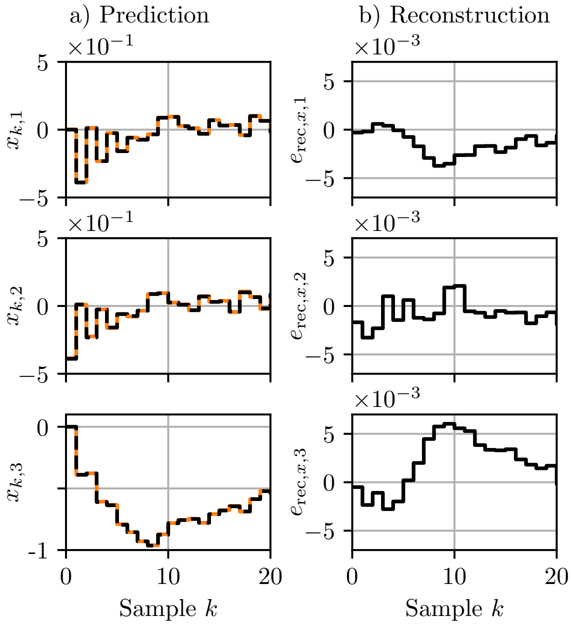

Fig. 2. a) Initial-value simulation of the model with adopted input signal from the validation trajectory. Comparison between the trajectory simulated with the trained model (- - -) and the validation trajectory (—). b) Reconstruction error $e_{\mathrm{rec},x,i} = x_{k,i} - \Phi^{-1} \circ \Phi(x_{k,i})$ (—) of the state transformation evaluated on the validation state trajectory.

TABLE I

VALIDATION OF THE MODEL WITH DIFFERENT OBSERVABILITY INDICES

	with y-transformation			without y-transformation		
(λ_1, λ_2)	$(5, 1)$	$(4, 2)$	$(3, 3)$	$(5, 1)$	$(4, 2)$	$(3, 3)$
MSE/10^{-3}	88.32	83.77	1.321	88.48	87.11	34.26

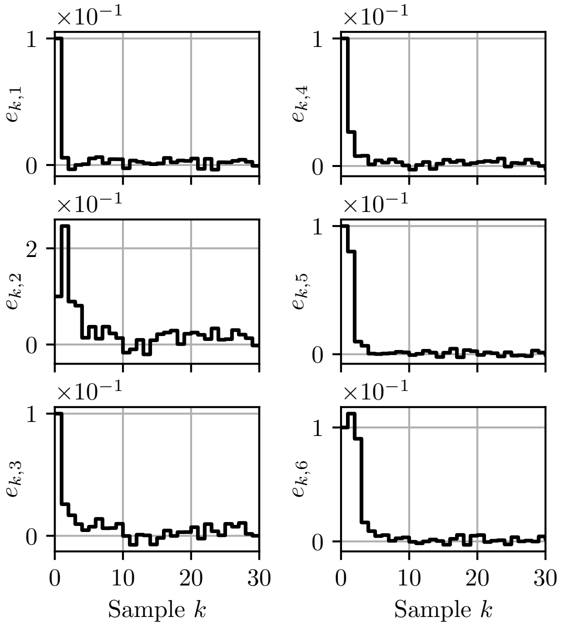

Fig. 3. Observer error $e_k = x_k - \hat{x}_k$ (——) between a validation trajectory x_k and the state estimation $\hat{x}_k$ of the designed observer with trained neural observer structure and indices (3,3).

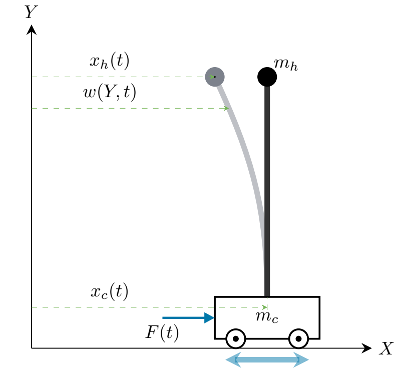

Fig. 4. Schematic of the single-mast stacker crane without lifting unit.

trajectory is shown in Fig. 3. In particular, the figure shows the evolution of the estimation error resulting from an initial observer error $e_0 = [0.1,\, 0.1,\, 0.1,\, 0.1,\, 0.1,\, 0.1]^T$. After a short settling time, the error converges to almost zero, but due to the numerical approximation with the neural networks, slight fluctuations around zero remain.

B. Industrially Motivated Application

The indirect data-driven observer design process is now exemplified using a model of a single-mast stacker crane without lifting unit, which has also been discussed in [20], [6], [22]. The model, as schematically depicted in Fig. 4, is described by the position x_c of the rigid driving cart with mass m_c, the horizontal deflection x_h of the tip mass m_h, and the spatial-dependent deflection $w(Y, t)$ of the Euler-Bernoulli beam with length L, mass density ρA, and bending stiffness EI. The purpose of this example is to demonstrate the data-driven observer design process in a sampled-data scenario, where the information of the full state, the input of the system, and the signals of the desired output, originate from recorded trajectory data sets. Note that a discrete-time observer is designed with data sampled from the continuous-time system. The position of the driving unit and an acceleration measurement at the head mass are considered as output signals. The mathematical model of this problem without lifting unit corresponds to a linear system of ordinary and partial differential equations, which have been finite-dimensionally approximated by a first-order Ritz ansatz for the simulation and generation of trajectories. For a comprehensive derivation of the mathematical model using a variational approach, the reader is referred to [6], [22]. The governing equations consist of the partial differential equation describing the Euler-Bernoulli beam, the momentum equations governing the driving unit and tip mass, and the boundary constraints. To approximate the beam deformation $w(Y, t)$, the first-order Rayleigh-Ritz ansatz $w(Y, t) = x_c(t) + \Upsilon(Y)\bar{q}(t)$ with appropriate spatial ansatz function $\Upsilon(Y)$ is utilized, see [6]. Subsequently, the Euler-Lagrange equations are solved to obtain a finite-dimensional mechanical system of the form $M\ddot{q} + Cq = Gu$ where the generalized coordinates are $q = [x_c, \bar{q}]^T$, and the input force is $u = F$. The matrices involved are given by

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ c_2 \end{bmatrix}, \text{ and } G = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

where m_{11}, m_{12},m_{22} and c_2 are defined as $m_{11} = \rho A L + m_c + m_h$, $m_{12} = m_h \Upsilon(L) + \rho A \int_0^L \Upsilon(Y)\mathrm{d}Y$, and $m_{22} = m_c \Upsilon(L)^2 + \rho A \int_0^L \Upsilon(Y)^2 \mathrm{d}Y$ and $c_2 = EI \int_0^L \left(\partial^2 \Upsilon(Y)/\partial Y^2\right)^2 \mathrm{d}Y$. The model of the single-mast stacker crane has been simulated with the parameters $L = 0.53\,\mathrm{m}, m_c = 13.10\,\mathrm{kg}, m_h = 0.32\,\mathrm{kg}, \rho A = 2.10\,\mathrm{kg/m}$ and $EI = 14.97\,\mathrm{Nm}^2$. For the identification of the system in Brunovsky observer form, the state $x = [x_c, x_{h,rel}, \dot{x}_c, \dot{x}_{h,rel}]^T$ with the relative deflection of the beam $x_{h,rel} = \Upsilon(L)\bar{q}$ will be considered. The data set for the training of the neural observer structure includes the desired output signals $\mathcal{Y} = \{(y_{1,i}, y_{2,i})\}_{i \in \mathcal{Q}}$ with $y_{1,k} = x_{c,k}$ and $y_{2,k} = \frac{EI}{m_h} \frac{\partial^3 \Upsilon(Y)}{\partial Y^3}\Big|_L \frac{x_{h,rel,k}}{\Upsilon(L)}$. The training data originates from 200 recorded trajectories with different excitation input signals and initial values. A total of roughly $n_s = 20000$ samples were recorded with a sampling time of $T_s = 50\,\mathrm{ms}$. The observability indices of the final model $(2, 2)$ have been determined by testing all possible combinations, as in the previous example. Based on the trained neural observer structures, the observer has been designed according to Section IV. The coefficients of the characteristic polynomials required for the observer gain were determined with the desired pole positions $[0.2, 0.3]$ and $[0.5, 0.2]$. The performance of the observer is presented in Fig. 5. Note, that for the validation of the observer in Fig. 5, the system has been excited with a primitive input trajectory in order to stimulate beam oscillation. As expected, the initial observer error decreased rapidly and converged to almost zero after a very short settling time.

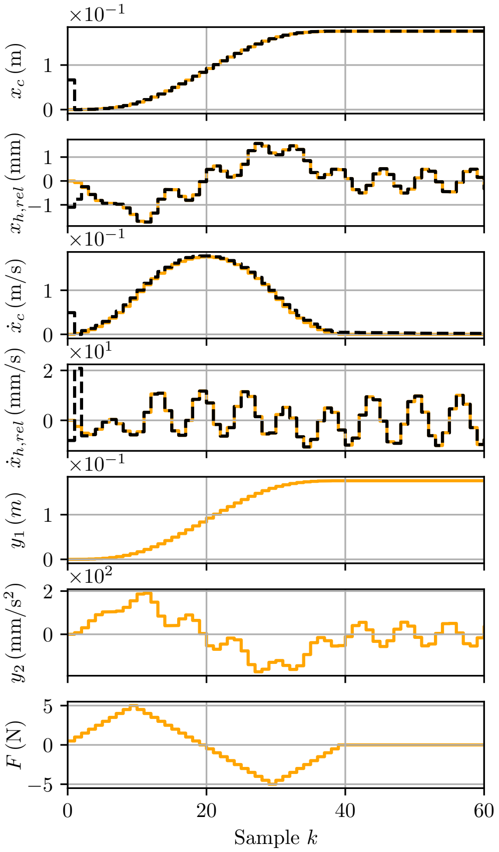

Fig. 5. Demonstration of the observer performance by comparing the state estimation $\hat{x}_k$ (- - -) with a validation trajectory x_k (—) of the single-mast stacker crane model.

VI. Conclusion And Future Works

A. Conclusions

In this paper, a data-driven observer design approach has been proposed and demonstrated on two academic examples and one industrially motivated example. The presented methodology is based on the idea of training an neural autoencoder structure and a neural network approximation of an input-output injection from historical full-state sensor data in order to represent the original system in a Brunovsky observer form. This observer form allows to design an observer as a copy of the plant with an additional output error term. The nonlinearities in the identified input-output injection cancel to a linear error dynamics, which can be adjusted by the output term according to linear system theory.

B. Future Works

Several directions are in the focus of upcoming investigations, such as the extension of the neural observer structures to handle an observer design approach based on immersion techniques and dynamic extension, c.f., [10], [18]. For this purpose, the neural observer structure has to be embedded in an higher-dimensional state-space $\dim(z_k) > \dim(x_k)$. However, this also increases the number of possible chain length constellations, which can be very unfavorable since it also means that the computational effort of the training process increases. Thus, the intention is to derive the observer indices from historical sensor data to avoid them as additional hyperparameters.

References

[1] H. Nijmeijer and A. Schaft, van der, *Nonlinear dynamical control systems*. Germany: Springer, 1991.

[2] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. USA: Wiley-Interscience, 2006.

[3] L. Ecker and K. Schlacher, “An approximation of the bayesian state observer with markov chain monte carlo propagation stage,” *IFAC-PapersOnLine*, vol. 55, no. 20, pp. 301–306, 2022, 10th Vienna International Conference on Mathematical Modelling MATHMOD 2022.

[4] C. V. Rao and J. B. Rawlings, “Nonlinear moving horizon state estimation,” in *Nonlinear Model Predictive Control*, F. Allgöwer and A. Zheng, Eds. Basel: Birkhäuser Basel, 2000, pp. 45–69.

[5] L. Ecker and M. Schöberl, “Data-driven observer design for an inertia wheel pendulum with static friction,” *IFAC-PapersOnLine*, vol. 55, no. 40, pp. 193–198, 2022, 1st IFAC Workshop on Control of Complex Systems COSY 2022.

[6] L. Ecker, T. Malzer, A. Wahrburg, and M. Schöberl, “Observer design for a single mast stacker crane,” *at - Automatisierungstechnik*, vol. 69, no. 9, pp. 806–816.

[7] L. Ecker, K. Schlacher, and M. Schöberl, “Observer design for an inertia wheel pendulum with static friction,” *IFAC-PapersOnLine*, vol. 55, no. 20, pp. 313–318, 2022, 10th Vienna International Conference on Mathematical Modelling MATHMOD 2022.

[8] B. Biedermann, P. Rosenzweig, and T. Meurer, “Passivity-based observer design for state affine systems using interconnection and damping assignment,” in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 4662–4667.

[9] H. Shim, J. Seo, and A. Teel, “Nonlinear observer design via passivation of error dynamics,” *Automatica*, vol. 39, no. 5, pp. 885–892, 2003.

[10] D. Boutat and G. Zheng, *Observer Design for Nonlinear Dynamical Systems Differential Geometric Methods*. Springer, 2021.

[11] A. J. Krener and A. Isidori, “Linearization by output injection and nonlinear observers,” *Systems and Control Letters*, vol. 3, no. 1, pp. 47–52, 1983.

[12] D. Bestle and M. Zeitz, “Canonical form observer design for non-linear time-variable systems,” *International Journal of Control*, vol. 38, no. 2, pp. 419–431, 1983.

[13] A. J. Krener and W. Respondek, “Nonlinear observers with linearizable error dynamics,” *SIAM Journal on Control and Optimization*, vol. 23, no. 2, pp. 197–216, 1985.

[14] X.-H. Xia and W.-B. Gao, “Non-linear observer design by observer canonical forms,” *International Journal of Control*, vol. 47, no. 4, pp. 1081–1100, 1988.

[15] H.-G. Lee, A. Arapostathis, and S. I. Marcus, “Necessary and sufficient conditions for state equivalence to a nonlinear discrete-time observer canonical form,” *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2701–2707, 2008.

[16] C. Califano, S. Monaco, and D. Normand-Cyrot, “On the observer design of multi output systems in discrete time,” *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 7–12, 2005, 16th IFAC World Congress.

[17] H.-G. Lee and H. Hong, “Remarks on discrete-time multi-output nonlinear observer canonical form,” *International Journal of Control, Automation and Systems*, vol. 16, no. 5, pp. 2569–2574, 2018.

[18] D. Bouta and K. Busawon, “On the transformation of nonlinear dynamical systems into the extended nonlinear observable canonical form,” *International Journal of Control*, vol. 84, no. 1, pp. 94–106, 2011.

[19] R. Tami, D. Boutat, and G. Zheng, “Extended output depending normal form,” *Automatica*, vol. 49, no. 7, pp. 2192–2198, 2013.

[20] L. Ecker and M. Schöberl, “Data-driven control and transfer learning using neural canonical control structures.” [Online]. Available: https://arxiv.org/abs/2302.04042

[21] B. Lusch, J. Kutz, and S. Brunton, “Deep learning for universal linear embeddings of nonlinear dynamics,” *Nature Communications*, vol. 9, 2018.

[22] H. Rams, M. Schöberl, and K. Schlacher, “Optimal motion planning and energy-based control of a single mast stacker crane,” *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1449–1457, 2018.