Automatica 167 (2024) 111787

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Meta-state–space learning: An identification approach for stochastic dynamical systems[☆,☆☆]

Gerben I. Beintema[a,*], Maarten Schoukens[a], Roland Tóth[a,b]

[a] Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
[b] Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control, Budapest, Hungary

ARTICLE INFO

Article history:
Received 13 July 2023
Received in revised form 17 February 2024
Accepted 1 May 2024
Available online 12 June 2024

Keywords:
System identification
Stochastic nonlinear systems
Hidden Markov models
Deep learning

ABSTRACT

Available methods for identification of stochastic dynamical systems from input–output data generally impose restricting structural assumptions on either the noise structure in the data-generating system or the possible state probability distributions. In this paper, we introduce a novel identification method of such systems, which results in a dynamical model that is able to produce the time-varying output distribution accurately without taking restrictive assumptions on the data-generating process. The method is formulated by first deriving a novel and exact representation of a wide class of nonlinear stochastic systems in a so-called meta-state–space form, where the meta-state can be interpreted as a parameter vector of a state probability function space parameterization. As the resulting representation of the meta-state dynamics is deterministic, we can capture the stochastic system based on a deterministic model, which is highly attractive for identification. The meta-state–space representation often involves unknown and heavily nonlinear functions, hence, we propose an *artificial neural network* (ANN)-based identification method capable of efficiently learning nonlinear meta-state–space models. We demonstrate that the proposed identification method can obtain models with a log-likelihood close to the theoretical limit even for highly nonlinear, highly stochastic systems.

1. Introduction

The framework of stochastic dynamical systems and (hidden) Markov models is a well-respected field of study with many powerful results and application areas (Paul & Baschnagel, 1999). From model predictive control with uncertain stochastic dynamics and probabilistic constraints (Mesbah, 2016), till filtering with Kalman variants up to particle filters (Wills & Schön, 2023), general stochastic models have been extensively applied in many practical applications from the process industry to aerospace engineering. As often the relations governing the system behavior are only partly known in practice, estimation of a reliable model of the dynamics from measured *input–output* (IO) data has been a central question of research (Paul & Baschnagel, 1999).

For the case, when the dynamics are *Linear-time-invariant* (LTI), identification algorithms have been developed to accomplish this by estimating both a process model and a noise model. Examples of such methods include *subspace approaches* (Larimore, 1990; Van Overschee & De Moor, 1994; Verhaegen & Dewilde, 1992) and *prediction error methods*, resulting in IO models with *autoregressive with exogenous input* (ARX), *moving average ARX* (ARMAX), *output-error* (OE), and *Box–Jenkins* (BJ) noise structures or state–space models, e.g., with an *innovation noise* structure (Ljung, 1999). In comparison, most *nonlinear-time-invariant* (NLTI) identification algorithms are highly restrictive on the noise structures which are allowed to be present in the system. The noise models are mostly limited to *nonlinear ARX* (NARX) and *nonlinear OE* (NOE) models in the input–output representation case (Schoukens & Ljung, 2019), and innovation noise structures in the state–space case (Van Wingerden & Verhaegen, 2009). These noise models are commonly considered as one of their advantages is that they allow estimation of the noise corrupting the data, under the assumption that the data-generating system itself falls into these model classes. When considering

[☆] This research has been funded by the European Union (ERC, COMPLETE, 101075836) and also by the Eötvös Loránd Research Network (Grant Number: SA-77/2021). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. The material in this paper was partially presented at The 2023 European Research Network System Identification (ERNSI), September 24–27, 2023, Stockholm, Sweden. This paper was recommended for publication in revised form by Associate Editor Dario Piga under the direction of Editor Alessandro Chiuso.

[☆☆] Implementation of the proposed approach is available in the toolbox: https://github.com/GerbenBeintema/metaSI, while the software for the examples considered in this paper can be found at: https://github.com/GerbenBeintema/metaSS-code-repo.

[*] Corresponding author.
E-mail addresses: g.i.beintema@tue.nl (G.I. Beintema), m.schoukens@tue.nl (M. Schoukens), r.toth@tue.nl (R. Tóth).

https://doi.org/10.1016/j.automatica.2024.111787

highly-structured models, such as block-oriented models, specific structured nonlinear noise models have been also considered (Hagenblad, Ljung, & Wills, 2008; Schoukens, Bai, & Rolain, 2012; Schoukens & Tiels, 2017).

Kernel-based methods such as regularization networks, support vector machines, and Gaussian regression offer another class of system identification approaches, both in the LTI and NLTI cases, with a robust mathematical framework and direct estimation of model uncertainty (Chiuso & Pillonetto, 2019). Such methods have been applied in system identification under various noise assumptions, e.g., white process noise under full-state measurement (Deisenroth & Rasmussen, 2011; Eleftheriadis, Nicholson, Deisenroth, & Hensman, 2017), innovation noise (Shakib, Tóth, Pogromsky, Pavlov, & van de Wouw, 2020), and equation-error noise (Pillonetto, Quang, & Chiuso, 2011). Kernel methods can theoretically be extended to most model and process noise structures by appropriate selection of the involved kernels, however, currently, there exists no general systematic approach for appropriate kernel construction when the noise dynamics are unknown (Chiuso & Pillonetto, 2019). While conceptually kernel selection could be automatized with for instance genetic programming (Khandelwal, Schoukens, & Tóth, 2023), the involved computational effort could be overwhelming for many practical identification problems.

Another method is the identification of probabilistic state–space models through expectation maximization (Schön, Wills, & Ninness, 2011) using particle and smoothing filters. This method does not impose major restrictions on either the noise or the model structure, however, it can have a significant computational cost due to the Monte Carlo nature of such particle-based approaches.

In this paper, we contribute to resolving the current challenges in stochastic dynamical system identification by proposing a so-called meta-state–space representation of the dynamics and an identification algorithm to estimate it from data. In contrast to previous work, the meta-state–space representation is directly applicable to a wide range of nonlinear stochastic systems and it provides an exact representation where the meta-state can be seen as a state probability function space parameterization. An attractive property of the meta-state is that it fully represents the complete distribution of the original state, but its evolution, i.e., the meta-state transition function, is deterministic. Remarkably, the latter allows to capture of the stochastic process representation via a deterministic model, capable of describing the evolution of the complete *probability density function* (PDF) of the state trajectories as a response to an input sequence. Furthermore, since the output PDF is a function of the state PDF, the meta-state–space representation is also able to describe the PDF of the output trajectories. As the meta-state–space representation of a stochastic process often involves unknown and heavily nonlinear functions, we propose an *artificial neural network* (ANN)-based identification method that, by exploiting the universal approximator capabilities of ANNs, is capable of learning such meta-state–space models efficiently directly from data. This provides a general approach for data-driven modeling of stochastic systems well beyond the capabilities of the current state-of-the-art without any severe structural restriction or limiting assumption.

To summarize, the main contributions of the paper are:

- Showing that a wide class of nonlinear stochastic systems have a meta-state–space representation;
- Formulation of a stochastic system identification algorithm based on meta-state–space models and using only measured IO data;
- Solving the identification problem by a computationally efficient ANN-based parameterization and estimation approach.

This paper is structured as follows, Section 2 introduces and proves the existence of the meta-state–space representation. In Section 3, we formulate the identification problem of stochastic systems via meta-state–space models using only IO data and propose a solution to it by an ANN-based approach. This is followed by Section 4, where capabilities of the proposed identification method are demonstrated on a challenging stochastic nonlinear system identification problem where the resulting estimation performance is found to be close to the theoretical limit. Lastly, conclusions on the achieved results and future research directions are provided in Section 5.

2. The meta-state–space representation

Consider a discrete-time nonlinear stochastic system with process and measurement noise described by

$$x_{t+1} = f_{\mathrm{x}}(x_t, u_t, v_t), \qquad y_t = h_{\mathrm{x}}(x_t, u_t, e_t), \tag{1a}$$

where x_t represents the state which is a random variable taking values from $\mathbb{X} \subseteq \mathbb{R}^{n_{\mathrm{x}}}$ with initial condition x_0 described by the PDF $p_0^{\mathrm{x}} : \mathbb{X} \to \mathbb{R}^+$, $u_t \in \mathbb{U} \subseteq \mathbb{R}^{n_{\mathrm{u}}}$ is a known input signal for simplicity of derivation (deterministic sequence or sample-path realization of an input process), y_t represents the output which is a random variable taking values from $\mathbb{Y} \subseteq \mathbb{R}^{n_{\mathrm{y}}}$ and $t \in \mathbb{Z}_0^+$ is the discrete time. The output is corrupted by some i.i.d. stationary measurement noise e with PDF $p^{\mathrm{e}} : \mathbb{R}^{n_{\mathrm{e}}} \to \mathbb{R}^+$. Furthermore, the state transition is also corrupted by some i.i.d. stationary process noise v with PDF $p^{\mathrm{v}} : \mathbb{R}^{n_{\mathrm{v}}} \to \mathbb{R}^+$. Both e and v are considered to be independent of u. Lastly, $f_{\mathrm{x}} : \mathbb{X} \times \mathbb{U} \times \mathbb{R}^{n_{\mathrm{v}}} \to \mathbb{X}$ and $h_{\mathrm{x}} : \mathbb{X} \times \mathbb{U} \times \mathbb{R}^{n_{\mathrm{e}}} \to \mathbb{Y}$ are bounded functions of the state-transition and output functions respectively.

The system described by the *state–space* (SS) representation (1) can be equivalently represented in the form of state-transition probabilities and conditional output probabilities, i.e., a *hidden Markov model* or *probabilistic state–space representation*:

$$\begin{aligned} p^{\mathrm{F}}(x_{t+1}|x_t, u_t) &= \textstyle\int p(x_{t+1}|x_t, u_t, v_t)p(v_t)dv_t, &(2a)\\ &= \textstyle\int \delta(x_{t+1} - f_{\mathrm{x}}(x_t, u_t, v_t))p(v_t)dv_t, &\\ p^{\mathrm{H}}(y_t|x_t, u_t) &= \textstyle\int p(y_t|x_t, u_t, e_t)p(e_t)de_t, &(2b)\\ &= \textstyle\int \delta(y_t - h_{\mathrm{x}}(x_t, u_t, e_t))p(e_t)de_t, & \end{aligned}$$

where δ is the *Dirac delta* function and p denotes the corresponding PDFs. For clarity of the derivation, we will adapt the following notation to indicate state and output probabilities at time $t \in \mathbb{Z}_0^+$:

$$p_t^{\mathrm{x}}(x) \triangleq p(x_t), \qquad p_t^{\mathrm{y}}(y) \triangleq p(y_t).$$

This notation allows us to express all future probability distributions $p_t^{\mathrm{x}}(x)$ and $p_t^{\mathrm{y}}(y)$ given an initial state distribution $p_0^{\mathrm{x}}(x) = p(x_0)$ and input signal using the *Chapman–Kolmogorov equations* (Paul & Baschnagel, 1999):

$$p_{t+1}^{\mathrm{x}}(x) = \int p^{\mathrm{F}}(x|x', u_t)p_t^{\mathrm{x}}(x')dx', \tag{3a}$$

$$p_t^{\mathrm{y}}(y) = \int p^{\mathrm{H}}(y|x', u_t)p_t^{\mathrm{x}}(x')dx'. \tag{3b}$$

We can also use functional operator notation to rewrite this in the following form

$$p_{t+1}^{\mathrm{x}} = F(p_t^{\mathrm{x}}, u_t), \tag{4a}$$

$$p_t^{\mathrm{y}} = H(p_t^{\mathrm{x}}, u_t). \tag{4b}$$

Introduce $u_\tau^d = [u^\top(\tau) \quad u^\top(\tau+1) \quad \cdots \quad u^\top(\tau+d)]^\top$. Using this notation and (4), we can describe the state and output distribution evolution as

$$p_t^{\mathrm{x}} = F^t(p_0^{\mathrm{x}}, u_0^{t-1}), \tag{5a}$$

$$p_t^{\mathrm{y}} = H(F^t(p_0^{\mathrm{x}}, u_0^{t-1}), u_t), \tag{5b}$$

where F^t is defined in a recurrent manner as

$$F^t(p_0^{\mathrm{x}}, u_0^{t-1}) \triangleq F(F^{t-1}(p_0^{\mathrm{x}}, u_0^{t-2}), u_{t-1}), \tag{6a}$$

$$F^0(p_0^{\mathrm{x}}) \triangleq p_0^{\mathrm{x}}. \tag{6b}$$

In terms of (5), we can characterize all possible state distributions that can happen along the solution trajectories of (1):

$$\mathcal{S}^{\mathbb{X}} = \{F^t(p_0^{\mathrm{x}}, u_0^{t-1}) \mid p_0^{\mathrm{x}} \in \mathcal{S}_0^{\mathbb{X}}, u_0^{t-1} \in \mathbb{U}^t, t \geq 0\} \tag{7}$$

where $\mathcal{S}_0^{\mathbb{X}}$ are all initial state distributions of interest and $\mathbb{U}^t \subseteq \{\{u_0, u_1, \ldots, u_t\} \mid u_i \in \mathbb{U} \subseteq \mathbb{R}^{n_{\mathrm{u}}}, t \geq i \geq 0\}$ is the set of all allowed input trajectories. Lastly, $\mathcal{S}^{\mathbb{Y}}$ is defined in a similar way for the output distributions.

To derive the meta-state–space representation, we require that $\mathcal{S}^{\mathbb{X}}$ is parameterizable in terms of the following definition.

Definition 1 (*Uniquely Parameterizable PDF Set*)**.** A set of probability density functions $\mathcal{S}^{\mathbb{X}}$ is called *uniquely parameterizable of order* n_{z} if there exists an injective mapping $M : \mathcal{S}^{\mathbb{X}} \to \mathbb{R}^{n_{\mathrm{z}}}$. Hence, the inverse $M^\dagger$ exists on the co-domain of M given $\mathcal{S}^{\mathbb{X}}$.

Now we have all the ingredients to show the existence of the *Meta-State–Space* (MSS) representation by the following theorem:

Theorem 2 (*Meta-State–Space Representation*)**.** *Assume that the set of probability functions $\mathcal{S}^{\mathbb{X}}$ formed by (1) is uniquely parameterizable of order n_{z} according to Definition 1. Then, there exist $f_{\mathrm{z}} : \mathbb{R}^{n_{\mathrm{z}}} \times \mathbb{R}^{n_{\mathrm{u}}} \to \mathbb{R}^{n_{\mathrm{z}}}$, $h_{\mathrm{z}} : \mathbb{R}^{n_{\mathrm{z}}} \times \mathbb{R}^{n_{\mathrm{u}}} \to \mathcal{S}^{\mathbb{Y}}$ and $z_0 \in \mathbb{R}^{n_{\mathrm{z}}}$ such that*

$$z_{t+1} = f_{\mathrm{z}}(z_t, u_t), \tag{8a}$$

$$p_t^{\mathrm{x}} = M^\dagger(z_t), \tag{8b}$$

$$p_t^{\mathrm{y}} = h_{\mathrm{z}}(z_t, u_t), \tag{8c}$$

for all $t \geq 0$, all $p_0^{\mathrm{x}} \in \mathcal{S}_0^{\mathbb{X}}$, and all $u \in \mathbb{U}^\infty$.

Proof. We provide the proof by induction:

Initial condition: $p_0^{\mathrm{x}} = M^\dagger(z_0)$ by setting $z_0 = M(p_0^{\mathrm{x}})$.

Induction step: If $p_t^{\mathrm{x}} = M^\dagger(z_t)$, then

$$\begin{aligned} p_t^{\mathrm{x}} &= M^\dagger(z_t) && \text{(apply } F\text{)} \\ F(p_t^{\mathrm{x}}, u_t) &= F(M^\dagger(z_t), u_t) && \text{(use (4a))} \\ p_{t+1}^{\mathrm{x}} &= F(M^\dagger(z_t), u_t) && \text{(apply } M^\dagger M \text{ RHS)} \\ p_{t+1}^{\mathrm{x}} &= M^\dagger(\underbrace{M(F(M^\dagger(z_t), u_t))}_{f_{\mathrm{z}}(z_t, u_t)}) \end{aligned}$$

and thus $p_{t+1}^{\mathrm{x}} = M^\dagger(z_{t+1})$ holds with

$$z_{t+1} = f_{\mathrm{z}}(z_t, u_t) \triangleq M(F(M^\dagger(z_t), u_t)).$$

Output case: By applying $M^\dagger$:

$$p_t^{\mathrm{y}} = H(p_t^{\mathrm{x}}, u_t) = H(M^\dagger(z_t), u_t) \triangleq h_{\mathrm{z}}(z_t, u_t). \quad \blacksquare$$

A way to understand this proof is by viewing Fig. 1 which shows that $z_{t+1} = f_{\mathrm{z}}(z_t, u_t) = M(F(M^\dagger(z_t), u_t))$ by the properties of set mappings.

With this, we have shown the existence of an MSS representation of the system described by (1) in the form:

$$z_{t+1} = f_{\mathrm{z}}(z_t, u_t), \tag{9a}$$

$$p(y_t \mid z_t, u_t), \tag{9b}$$

where $p(y_t \mid z_t, u_t)$ is given by $h_{\mathrm{z}}(z_t, u_t)$. A graphical illustration of the evolution of the meta-state and its relation to the time-variation of the original state distribution is given in Fig. 2. The MSS representation is especially suited for system identification since (9) is similar to the nonlinear SS representation of a deterministic system which has been studied extensively in the literature.

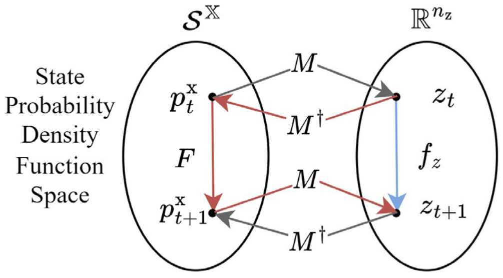

Fig. 1. Both $\mathcal{S}^{\mathbb{X}}$ and the meta-state–space with mapping M and inverse mapping $M^\dagger$ are visualized, showing that a transition from z_t to z_{t+1} can be computed in two ways by following either the blue or the red path and thus $z_{t+1} = f_{\mathrm{z}}(z_t, u_t) = M(F(M^\dagger(z_t), u_t))$.

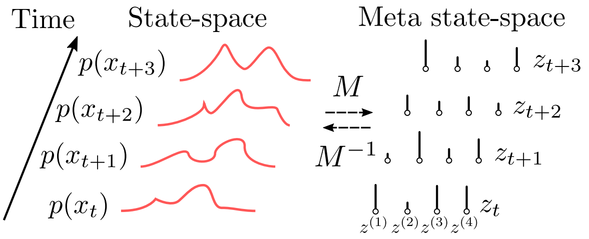

Fig. 2. A graphical representation of the evolution of a PDF of the state according to (3) and the evolution of the meta-state in terms of (9). This figure shows that meta-state vectors z_t can represent the state distribution p_t^{x} through the mapping M.

Remark 3. Existence of an MSS, depends on the assumption that $\mathcal{S}^{\mathbb{X}}$ is uniquely parameterizable of order n_{z} according to Definition 1. Hence it is an important question if such a parameterization exists or not for general nonlinear stochastic systems. It is well known that distributions in general can be uniquely defined in terms of their moments, which means that MSSs with potentially infinite order n_{z} always exist. Higher-order moments have a diminishing role, and hence, often only a subset of the moments and thus finite n_{z} is enough to provide an accurate characterization of $\mathcal{S}^{\mathbb{X}}$. Additionally, there exist many universal approximators which can describe function spaces to arbitrary accuracy with increasing order n_{z}. For example, the difference becomes arbitrarily small with increasing the number of particles (Del Moral, 1997) or increasing the number of components in a Gaussian mixture (Goodfellow, Bengio, & Courville, 2016). These approaches have been exploited in particle filtering (Schön et al., 2011) and Markov-chain Monte-Carlo methods (Chua, Calandra, McAllister, & Levine, 2018) to provide state-filtering and estimation for general stochastic systems. Hence the motivation for the existence of MSS models of (1) with finite n_{z} in an exact or approximative sense is based on the same considerations. However, characterization of the minimal order n_{z} of unique parameterizations of $\mathcal{S}^{\mathbb{X}}$ for a given (1) and its boundedness are open questions and are outside the scope of the current paper.

3. Identification by meta-state–space learning

In this section, we will exploit the existence of MSS representations of stochastic systems in the form of (1) to formulate an efficient data-driven modeling approach of such systems within the meta-state–space setting by using *maximum a posteriori* (MAP) estimation.

3.1. The identification problem

Data set: An input sequence $\{u_t\}_{t=1}^{N}$, either generated as the sampling of an input process or as a deterministic sequence, and unknown initial state x_0 are applied on the system given by (4) and an output realization $\{y_t^*\}_{t=1}^{N}$ is recorded. These two sequences are used to create the input-output dataset:

$$\mathcal{U}_N = \{u_1, u_2, \ldots, u_N\}, \tag{10a}$$

$$\mathcal{Y}_N = \{y_1^*, y_2^*, \ldots, y_N^*\}. \tag{10b}$$

Model structure: To identify (1), estimation of the meta-state dynamics (9) in terms of a parameterized model

$$\hat{z}_{t+1} = f_\theta(\hat{z}_t, u_t), \tag{11a}$$

$$p_\theta(\hat{y}_t | \hat{z}_t, u_t), \tag{11b}$$

is considered, where $f_\theta : \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_z}$ and p_θ is a conditional PDF, both parameterized in terms of $\theta \in \mathbb{R}^{n_\theta}$. Furthermore, the initial state $\hat{z}_1$ is also considered to be a parameter.

Identification criterion: Estimation of $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ and the initial state $\hat{z}_1 \in \mathbb{R}^{n_z}$ are considered in terms of maximization of the posteriori likelihood over $\Theta \times \mathbb{R}^{n_z}$. Since the currently considered formulation of the meta-state–space representation cannot express the joint distribution of the output, e.g., $p(y_1, y_2 | u_1, u_2)$ we aim to maximize the product of the posteriori of each individual term as

$$[\theta, \hat{z}_1]_{\text{MAP}} = \arg\max_{\theta, \hat{z}_1} p(\theta, \hat{z}_1, \mathcal{U}_N) \prod_{t=1}^{N} p_\theta(y_t^* | \hat{z}_t, u_t),$$

where $p(\theta, \hat{z}_1, \mathcal{U}_N)$ is a user-specified prior distribution. By log and mean transformation, the MAP estimation problem of the parameters of a meta-state model can be posed as the following optimization problem:

$$\begin{aligned} \min_{\theta, \hat{z}_1} \quad & -\frac{1}{N} \sum_{t=1}^{N} \log(p_\theta(y_t^* | \hat{z}_t, u_t)) - \frac{1}{N} \log(p(\theta, \hat{z}_1, \mathcal{U}_N)) \\ \text{s.t.} \quad & \hat{z}_{t+1} = f_\theta(\hat{z}_t, u_t), \quad \forall t \in \mathbb{I}_1^N, \end{aligned} \tag{12}$$

where $\mathbb{I}_1^N = [1, N] \subset \mathbb{Z}$. Similar to classical system identification, minimizing a simulation cost can result in a good initial meta-state $\hat{z}_1$ since any error in it can result in a transient error which increases the cost (Forgione, Mejari, & Piga, 2022). An important observation is that this optimization problem neither requires M to be defined nor will result in an estimate of M which can be viewed as both an advantage and disadvantage. It is advantageous since the optimal choice of M is generally unknown and thus would be challenging to choose a priori. However, it is also disadvantageous since after estimation, it is unknown how the meta-state relates to the true hidden state distribution. An exception is when x_t can be directly observed (i.e. $y_t = x_t$), as in this case $p_\theta(x_t | \hat{z}_t)$, which corresponds to $M^\dagger$, is described by the model estimate.

3.2. Neural meta-state–space estimator

This section aims to make the optimization problem given by (12) computationally tractable for gradient-descent optimization algorithms. Moreover, this section will introduce an efficient way of parameterizing f_θ and p_θ using neural networks for general modeling purposes.

Meta-state–space models can be considered under various parameterizations of the meta-state transition function f_θ and output distribution $p_\theta(y|z, u)$. However, these functions can be rather complicated and heavily nonlinear, hence parameterization by artificial neural networks is desirable due to their expressiveness and favorable computational aspects (Goodfellow et al., 2016). We parameterize the mapping $z_+ = f_\theta(z, u)$ as a fully connected feedforward neural network with a linear bypass and n hidden layers which can be expressed recursively as:

$$\xi^{(0)} = [z^\top \ u^\top]^\top, \tag{13a}$$

$$\xi^{(i+1)} = \phi\left(A^{(i)}\xi^{(i)} + b^{(i)}\right), \tag{13b}$$

$$z_+ = A^{(n)}\xi^{(n)} + A_{\text{lin}}\xi^{(0)} + b^{(n)}, \tag{13c}$$

where $\xi^{(i)} \in \mathbb{R}^{n_{\text{hidden}}^{(i)}}$ are the hidden latent variables associated with the layers, $\{A_{\text{lin}}, A^{(0)}, b^{(0)}, \ldots, A^{(n_{\text{layers}})}, b^{(n_{\text{layers}})}\}$ are the real-valued network parameters with appropriate dimensions, and ϕ is a static nonlinear activation function which is applied element-wise. A good standard choice is the tanh activation function for ϕ since it is effective for many deep learning and system identification tasks (Beintema, Schoukens, & Tóth, 2023). However, other activation functions such as ReLU, Gaussian etc. and their combination for different layers can be more effective depending on the problem at hand (e.g., ReLu for piece-wise linear problems), see Goodfellow et al. (2016) for an overview.

Regarding p_θ, a flexible parameterization for non-Gaussian distributions is the mixture of Gaussian distributions (Bishop, 1994) based on which we consider

$$p_\theta(y|\xi) = \sum_{i=1}^{n_p} w_{i,\theta}(\xi) \cdot \mathcal{N}\big(y | \mu_{i,\theta}(\xi), \Sigma_{i,\theta}(\xi)\big), \tag{14}$$

where, $\xi = [z^\top \ u^\top]^\top$, n_p is the number of Gaussian components, $w_{i,\theta} : \mathbb{R}^{n_z+n_u} \to [0, 1]$ with $\sum_{i=1}^{n_p} w_{i,\theta} = 1$, $\mu_{i,\theta} : \mathbb{R}^{n_z+n_u} \to \mathbb{R}^{n_y}$, and $\Sigma_{i,\theta} : \mathbb{R}^{n_z+n_u} \to \mathbb{S}^{n_y}$, where $\mathbb{S}^{n_y}$ is the set of symmetric, positive definite matrices in $\mathbb{R}^{n_y \times n_y}$. The weight w_j, mean μ_j and covariance matrix Σ_j functions are chosen as fully connected feedforward ANNs with a similar structure as f_θ. To improve computational effectiveness we utilize ANNs with n_p outputs such that only three neural networks are required to parameterize the weights w_θ, means μ_θ and covariance terms Σ_θ. The validity of the probability distribution (i.e. $\int p_\theta(y|\xi)dy = 1$ and $p_\theta(y|\xi) \geq 0$, $\forall \xi$) is ensured by the given constraints and enforced by choosing appropriate activation functions on the last layer of each neural network as discussed in Appendix A. This type of distribution parameterization is also called a *Mixture Density Network* (Bishop, 1994). The considered parameterization is sufficient to describe any PDF function under the limit of $n_p \to \infty$ since a weighted sum of normal distributions is a universal approximator (Bishop, 1994; Goodfellow et al., 2016). Using a Mixture Density Network is advantageous since the log-likelihood as in (12) can be computed in an efficient manner. Lastly, mitigation of floating point errors/instabilities is essential and is also discussed in Appendix A.

The computational cost and optimization stability of problem (12) can be greatly enhanced by adopting a multiple shooting formulation (Bock, 1981). Following the multiple shooting approach of Beintema et al. (2023) for conventional nonlinear state–space estimation, we are able to reduce the computational complexity by using independent subsections of the available data. Based on these, we can recast (12) as the following optimization problem:

$$\min_\theta \quad -\sum_{t=1}^{N-T+1} \sum_{k=k_{\text{burn}}}^{T-1} \log(p_\theta(y_{t+k}^* | z_{t+k|t}, u_{t+k})), \tag{15a}$$

$$\text{s.t.} \quad z_{t+k+1|t} = f_\theta(z_{t+k|t}, u_{t+k}), \quad \forall k \in \mathbb{I}_0^{T-1} \tag{15b}$$

$$z_{t|t} = 0, \quad \forall t \in \mathbb{I}_1^{N-T+1} \tag{15c}$$

Table 1
The MSS model, cost and optimization parameters.

n_{layers}	n_{hidden}	n_z	n_p	k_{burn}	T	Learn rate	Batch size
2	64	3	30	10	30	10^{-3}	2048

in which we also assume a uniform prior. For simplicity of the implementation and computational feasibility, we set the initial state $z_{t|t}$ in each section to be fixed to zero in (15). This choice might result in a mismatch with the optimal initial meta-states and thus a transient error can be present. By including a small burn time k_{burn} the effect of this transient error is greatly reduced in the case of fading memory systems. This formulation also allows for the use of powerful batch optimization such as Adam (Kingma & Ba, 2015) by not summing over all possible t.

The proposed model structure and estimation method possess a number of important hyperparameters, which can be chosen based on the following guidelines:

- A key hyperparameter is the order of the meta-state–space model n_z. As mentioned in Remark 3, n_z should be chosen such that Definition 1 is satisfied with the desired level of user-defined accuracy with, for instance, cross-validation.
- The number of Gaussian components n_{p} should be chosen in a similar manner, but our observations suggest that $n_{\text{p}} = 20$ has been a sufficient baseline choice for all the datasets that we have considered.
- The k_{burn} and T can be chosen using n-step-error figures as described in Beintema et al. (2023). Hence, k_{burn} should be chosen larger than the transient observed in the n-step-error figure and T a few times that transient length for stable systems.

Other well-known modeling guidelines, for instance described in Beintema et al. (2023), still hold.

4. Simulation studies

To demonstrate the capabilities of the proposed method, consider the following nonlinear stochastic system

$$x_{t+1} = \alpha(x_t, e_t)x_t + u_t, \tag{16a}$$

$$y_t = x_t, \tag{16b}$$

where $\alpha(x_t, e_t) = 0.3 + 0.7e^{-(x_t+e_t)^2}$ which satisfies $|\alpha(x_t, e_t)| \leq 1$ to ensure stability. The process noise e_t is i.i.d with uniform distribution $p(e_t) = \mathcal{U}(e_t| - 0.5, 0.5)$.

We generate three separate datasets for training, validation and testing, employing a white input sampled from a zero-mean normal distribution with a standard deviation of 2 (i.e. $u_t \sim \mathcal{N}(u_t|0, 2)$). The training and validation set consists of 300k and 10k sample points respectively with $x_0 = 0$ as the initial state. The test set consists of 5000 trajectories of 4100 samples each, where the first 100 samples are discarded to exclude transient effects. These test trajectories all use the same input realization u_t, but different noise realizations e_t from the considered distribution. These trajectories allow us to compare the model and system output distributions.

The MSS model, cost and optimization parameters can be viewed in Table 1 which shows that f_θ is parameterized with a 2 hidden layer neural network with 64 nodes per layer with tanh activation functions as described in (13). To parameterize p_θ, we utilize a Gaussian mixture model as in (14) where $w_{\cdot,\theta}$, $\mu_{\cdot,\theta}$ and $\sigma_{\cdot,\theta}$ are neural networks with the same structure as f_θ. To train the meta-state–space model, we utilize the multiple-shooting-based loss function given by (15) since it scales well to the large training dataset. We use the Adam optimizer and the following hyperparameters, input–output normalization, early stopping using the validation set, $k_{\text{burn}} = 10$, $T = 30$, $n_{\text{p}} = 30$, batch size of 2048 and a learning rate of 10^{-3}. Lastly, using cross-validation we found that the model accuracy expressed in mean log-likelihoods is 1.525, 1.674 and 1.678 for $n_z = 2, 3$ and 4 respectively. Hence, increasing the meta-state dimension beyond $n_z = 3$ does not provide significant increase in model accuracy and thus $n_z = 3$ is chosen.

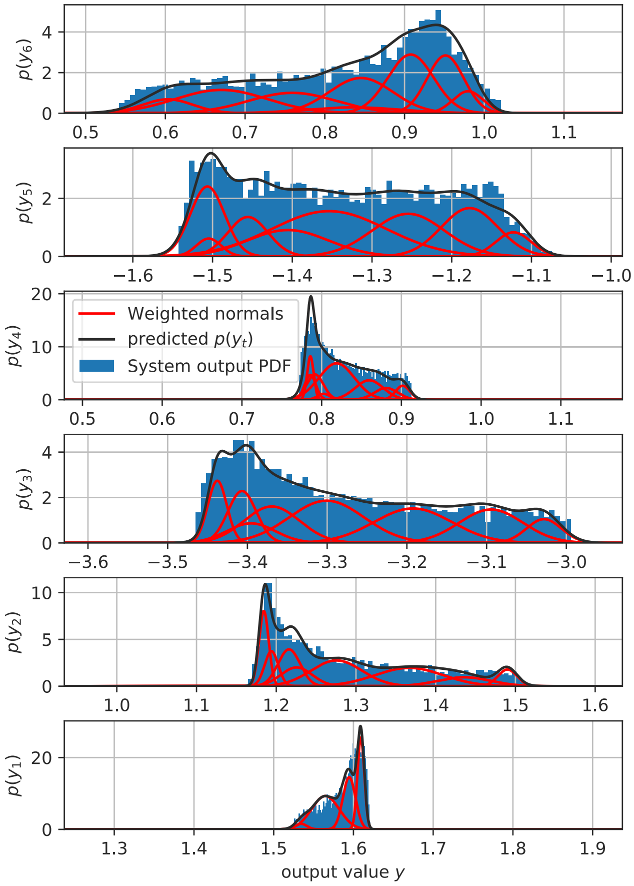

Fig. 3. Probability density histograms of the system output under different noise realizations compared with the predicted output distribution of the meta-state–space model. Since the model output distribution is parameterized as a weighted sum of normal distributions, these weighted components are also displayed.

The resulting meta-state model is analyzed by using both qualitative and quantitative comparisons. The qualitative comparison aims to investigate if the produced probability distribution of the output trajectories of the model well represents the probability distribution of the output trajectories of the considered stochastic system. Using the 5000 test trajectories, we can construct a probability density histogram of the system output over time and compare it to the output distributions given by the model at specific time instances. This comparison can be viewed in Fig. 3 which shows a striking resemblance between the probability density histogram of the test set and the output distributions given by the model. Not only the mean and the variance have been captured by the model, but also smaller features such as bumps as seen in the third row at $y = 0.8$ are present in the histogram. With this, we have shown that the evolution of the meta-state z_t can indeed describe the distributions of y_t using a qualitative comparison.

Numerically quantifying the model quality in a probabilistic setting is done by using the mean log-likelihood on the test set

Table 2
The mean model output log-likelihood (17) over the test set for the nonlinear stochastic system given by (16).

Model/Baseline	Mean log-likelihood
Gaussian $(\mu_{\mathrm{y}}, \sigma_{\mathrm{y}})$	−2.18
Gaussian $(\mu_{\mathrm{y},t}, \sigma_{\mathrm{y,s}})$	1.04
Gaussian $(\mu_{\mathrm{y},t}, \sigma_{\mathrm{y},t})$	1.56
Meta-state–space model	1.67
Upper limit	1.73

with $S = 5000$ test sequences:

$$\frac{1}{NS}\sum_{t=1}^{N}\sum_{i=1}^{S}\log p(y_t^{*(i)}|z_t) \tag{17}$$

where (i) indicates the ith trajectory. To ease the interpretation of this quantity, we included a comparison of 3 baselines and an estimated theoretical upper limit. The following baselines are considered (see Appendix C for details);

(1) Gaussian $(\mu_{\mathrm{y}}, \sigma_{\mathrm{y}})$: has a static mean and a static standard deviation. Comparison with this baseline gives an indication that the results are better than a static Gaussian model.
(2) Gaussian $(\mu_{\mathrm{y},t}, \sigma_{\mathrm{y,s}})$: has a dynamic mean and a static standard deviation. This baseline represents a model which is able to perfectly capture the mean variation of the output, but assumes a static output noise. The performance of this baseline is the upper limit of the performance of conventional output error modeling methods like Fraccaro, Kamronn, Paquet, and Winther (2017).
(3) Gaussian $(\mu_{\mathrm{y},t}, \sigma_{\mathrm{y},t})$; has a dynamic mean and a dynamic standard deviation. This baseline represents a model which is able to perfectly capture the mean variation of the output, but assumes that the output noise is a state-dependent Gaussian.
(4) Upper limit: it can be shown that for any model of the stochastic system, the mean output log-likelihood will be smaller or equal to the negative mean entropy of the output signal (Vasicek, 1976), see Appendix B for details.

The computed mean log-likelihood of all four baselines together with the obtained meta-state–space model are presented in Table 2. This table shows that the obtained model outperforms all three baselines and is very close to the estimated upper limit. This is a remarkable achievement since the meta-state–space model is able to describe the data generated by challenging stochastic dynamics using a deterministic state transition and a stochastic output map. Hence, the existence of the meta-state–space as derived in (9) opens up efficient identification methods that directly estimate meta-state–space models. Furthermore, we observe no major trends in the accuracy of the estimated model with increasing prediction horizon. This identification method and possible future extensions can greatly reduce the complexity of the identification problem of stochastic dynamics.

5. Conclusion

A novel meta-state–space identification method has been introduced which is able to identify general nonlinear stochastic systems with an accuracy close to the theoretical limit as shown in the simulation study. The identification method is formulated based on a meta-state–space representation of the system which can be interpreted as a description of the deterministic evolution of a parameter vector of a state distribution parameterization, called the meta-state. Identification based on this representation is effective since the meta-state transition function is deterministic and in the considered example the proposed method could achieve accuracy close to the theoretical limit.

By the current formulation, the estimated meta-state models allow only to express the output probability distributions of the type $p(y_t|u_0^t)$. However, we suspect that the meta-state can be extended for other prediction objectives. For instance, by the inclusion of a subspace encoder (Beintema et al., 2023) it is potentially possible to obtain accurate n-step ahead predictors such as $p(y_{n+t}|u_0^{n+t}, y_0^n)$ which would be useful for model-based control with chance constraints. Furthermore, by the inclusion of a Kalman measure update in the meta-state–space, it is potentially possible to obtain the joint probability distributions $p(y_0^n|u_0^n)$ which would be useful for filtering and observer tasks.

Appendix A. Mixture density network parameterization

Obtaining a valid mixture of Gaussians given by (14) requires that the weights are $\sum_i w_i = 1$ and $w_i > 0$ and that $\sigma_i > 0$. This is enforced by utilizing proper activation functions as described below. Furthermore, floating-point errors are minimized for a successful implementation.

To compute w_i in (14), it is suggested to use the following relations

$$\tilde{w}_i = \mathrm{ANN}^i_{\theta w}(z), \tag{A.1}$$

$$w_i = \frac{\exp(\tilde{w}_i)}{\sum_j \exp(\tilde{w}_j)} = \frac{\exp(\tilde{w}_i - \max_k(\tilde{w}_k))}{\sum_j \exp(\tilde{w}_j - \max_k(\tilde{w}_k))}. \tag{A.2}$$

Additionally, it is suggested to use

$$\tilde{\sigma}_i = \mathrm{ANN}^i_{\theta\sigma}(z), \tag{A.3}$$

$$\sigma_i = \exp(\tilde{\sigma}_i), \tag{A.4}$$

$$\mu_i = \mathrm{ANN}^i_{\theta\mu}(z), \tag{A.5}$$

where $n_y = 1$ with $\Sigma_i = \sigma_i^2$ is considered for simplicity. Lastly, we compute the log probability as follows

$$r_i \triangleq \log(w_i) + \log\left(N(y|\mu_i, \sigma_i)\right), \tag{A.6}$$

$$\log(p_\theta(y|z)) = \log\left(\sum_i \exp(r_i)\right), \tag{A.7}$$

$$= \max_k(r_k) + \log\left(\sum_i \exp\left(r_i - \max_k(r_k)\right)\right).$$

Here the max outside of the log reduces floating point errors which could prevent convergence of the optimization. Application of the max-operator is well-known to improve numerical floating point stability, for instance, this has been the reason for the introduction of softmax activation functions in machine learning (Karpathy et al., 2016).

Appendix B. Mean log-likelihood upper limit

It is well-known that the *Kullback–Leibler* (KL) divergence is zero if the given distribution $q(y_t)$ is equal to the target distribution $p(y_t)$:

$$D_{\mathrm{KL}}(p, q) = \int p(y_t)\log\left(\frac{p(y_t)}{q(y_t)}\right) dy_t, \tag{B.1}$$

$$= \underbrace{\int p(y_t)\log\left(p(y_t)\right) dy_t}_{\text{negative differential entropy}} - \underbrace{\int p(y_t)\log\left(q(y_t)\right) dy_t}_{\text{cross entropy}},$$

where the cross entropy is equal to our performance measure using sampling of $p(y_t)$. This also shows that the upper bound is the negative mean entropy of $p(y_t)$. However, since an analytical expression for $p(y_t)$ is unavailable in practice, we cannot compute the differential entropy directly and will need to estimate it using the data samples. Many methods exist for computing entropy from samples. For this purpose, we employ the often applied method proposed by Alizadeh Noughabi (2015), Vasicek (1976) to estimate the mean differential entropy.

Appendix C. Estimation of the Gaussian baselines

Parameters of the Gaussian baseline models, discussed in Section 4, are obtained using the following equations:

$$\mu_{\mathrm{y}} = \tfrac{1}{NS}\Sigma_{t=1}^{N}\Sigma_{i=1}^{S} y_t^{*(i)}, \tag{C.1}$$

$$\mu_{\mathrm{y},t} = \tfrac{1}{S}\Sigma_{i=1}^{S} y_t^{*(i)}, \tag{C.2}$$

$$\sigma_{\mathrm{y}}^2 = \tfrac{1}{NS}\Sigma_{t=1}^{N}\Sigma_{i=1}^{S}(y_t^{*(i)} - \mu_{\mathrm{y}})^2, \tag{C.3}$$

$$\sigma_{\mathrm{y,s}}^2 = \tfrac{1}{NS}\Sigma_{t=1}^{N}\Sigma_{i=1}^{S}(y_t^{*(i)} - \mu_{\mathrm{y},t})^2, \tag{C.4}$$

$$\sigma_{\mathrm{y},t}^2 = \tfrac{1}{S}\Sigma_{i=1}^{S}(y_t^{*(i)} - \mu_{\mathrm{y},t})^2. \tag{C.5}$$

References

Alizadeh Noughabi, H. (2015). Entropy estimation using numerical methods. *Annals of Data Science*, *2*(2), 231–241.

Beintema, Gerben I., Schoukens, Maarten, & Tóth, Roland (2023). Deep subspace encoders for nonlinear system identification. *Automatica*, *156*, Article 111210.

Bishop, C. M. (1994). *Mixture density networks*: *Technical report*, Neural Network Research Group, Aston University.

Bock, H. G. (1981). Numerical treatment of inverse problems in chemical reaction kinetics. *Modelling of Chemical Reaction Systems*, *18*, 102–125.

Chiuso, A., & Pillonetto, G. (2019). System identification: A machine learning perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, *2*, 281–304.

Chua, K., Calandra, R., McAllister, R., & Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, *31*.

Deisenroth, M., & Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 465–472).

Del Moral, P. (1997). Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, *325*(6), 653–658.

Eleftheriadis, S., Nicholson, T., Deisenroth, M., & Hensman, J. (2017). Identification of Gaussian process state space models. *Advances in Neural Information Processing Systems*, *30*.

Forgione, M., Mejari, M., & Piga, D. (2022). Learning neural state-space models: do we need a state estimator? arXiv preprint arXiv:2206.12928.

Fraccaro, M., Kamronn, S., Paquet, U., & Winther, O. (2017). A disentangled recognition and nonlinear dynamics model for unsupervised learning. *Advances in Neural Information Processing Systems*, *30*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hagenblad, A., Ljung, L., & Wills, A. (2008). Maximum likelihood identification of Wiener models. *Automatica*, *44*(11), 2697–2705.

Karpathy, A., et al. (2016). *Cs231n convolutional neural networks for visual recognition*. Stanford University.

Khandelwal, D., Schoukens, M., & Tóth, R. (2023). Automated multi-objective system identification using grammar-based genetic programming. *Automatica*, *154*, Article 111017.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR*.

Larimore, W. E. (1990). Canonical variate analysis in identification, filtering, and adaptive control. In *29th IEEE conference on decision and control* (pp. 596–604). IEEE.

Ljung, L. (1999). *System identification-theory for the user*. Upper Saddle River, NJ.

Mesbah, Ali (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, *36*(6), 30–44.

Paul, W., & Baschnagel, J. (1999). *Stochastic processes: From Physics to finance*. berlin: Springer.

Pillonetto, G., Quang, M. H., & Chiuso, A. (2011). A new kernel-based approach for nonlinearsystem identification. *IEEE Transactions on Automatic Control*, *56*(12), 2825–2840.

Schön, T. B., Wills, A., & Ninness, B. (2011). System identification of nonlinear state-space models. *Automatica*, *47*(1), 39–49.

Schoukens, M., Bai, E. W., & Rolain, Y. (2012). Identification of Hammerstein-Wiener systems. *IFAC Proceedings Volumes*, *45*(16), 274–279, Symposium on System Identification.

Schoukens, J., & Ljung, L. (2019). Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, *39*(6), 28–99.

Schoukens, M., & Tiels, K. (2017). Identification of block-oriented nonlinear systems starting from linear approximations: A survey. *Automatica*, *85*, 272–292.

Shakib, MF, Tóth, R, Pogromsky, AY, Pavlov, A, & van de Wouw, N (2020). State-space kernelized closed-loop identification of nonlinear systems. *IFAC-PapersOnLine*, *53*(2), 1126–1131.

Van Overschee, P., & De Moor, B. (1994). N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, *30*(1), 75–93.

Van Wingerden, J. W., & Verhaegen, M. (2009). Subspace identification of bilinear and LPV systems for open-and closed-loop data. *Automatica*, *45*(2), 372–381.

Vasicek, O. (1976). A test for normality based on sample entropy. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *38*(1), 54–59.

Verhaegen, M., & Dewilde, P. (1992). Subspace model identification part 2. Analysis of the elementary output-error state-space model identification algorithm. *International Journal of Control*, *56*(5), 1211–1241.

Wills, Adrian G., & Schön, Thomas B. (2023). Sequential Monte Carlo: A unified review. *Annual Review of Control, Robotics, and Autonomous Systems*, *6*, 159–182.

Gerben I. Beintema is a post-doc researcher at the Control Systems (CS) Group at the Department of Electrical Engineering. His current research is on the intersection of nonlinear system identification and deep learning, under the supervision of Professor Roland Tóth and Assistant Professor Maarten Schoukens. His main research interest is improving and understanding deep nonlinear state–space identification to obtain interpretable, robust, and generally applicable methods. Gerben I. Beintema obtained his Ph.D. degree in the CS group in 2024 on neural state–space identification.

Maarten Schoukens is an Associate Professor in the Control Systems group of the Department of Electrical Engineering at the Eindhoven University of Technology. He received a master's degree in electrical engineering and a Ph.D. degree in engineering from the Vrije Universiteit Brussel (VUB), Brussels, Belgium, in 2010 and 2015 respectively. From 2015 to 2017, he has been a Post-Doctoral Researcher with the ELEC Department, VUB. In October 2017 he joined the Control Systems research group, TU/e, Eindhoven, The Netherlands as a Post-Doctoral Researcher, in 2018 he became an Assistant Professor, and Associate Professor in 2023, in the same group. Maarten was awarded an FWO Ph.D. Fellowship in 2011, a Marie Skłodowska–Curie Individual Fellowship in 2018, and an ERC Starting Grant in 2022. His main research interests include the measurement and data-driven modeling and control of nonlinear dynamical systems using system identification and machine learning techniques. Furthermore, he is one of the organizers of the nonlinearbenchmark.org initiative promoting the use of common datasets in the development of data-driven nonlinear dynamical system modeling approaches.

Roland Tóth received his Ph.D. degree with Cum Laude distinction at the Delft University of Technology (TUDelft) in 2008. He was a post-doctoral researcher at TUDelft in 2009 and at the Berkeley Center for Control and Identification, University of California in 2010. He held a position at TUDelft in 2011–12, then he joined to the Control Systems (CS) Group at the Eindhoven University of Technology (TU/e). Currently, he is a Full Professor at the CS Group, TU/e and a Senior Researcher at the Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control (SZTAKI) in Budapest, Hungary. He is Senior Editor of the IEEE Transactions on Control Systems Technology. His research interests are in identification and control of linear parameter-varying (LPV) and nonlinear systems, developing data-driven and machine learning methods with performance and stability guarantees for modeling and control, model predictive control and behavioral system theory. On the application side, his research focuses on advancing reliability and performance of precision mechatronics and autonomous robots/vehicles with nonlinear, LPV and learning-based motion control. He has received the TUDelft Young Researcher Fellowship Award in 2010, the VENI award of The Netherlands Organization for Scientific Research in 2011, the Starting Grant of the European Research Council in 2016 and the DCRG Fellowship of Mathworks in 2022.