# On affine quasi-LPV System Identification with unknown state-scheduling using (deep) Recurrent Neural Networks*

1st Alexander Rehmer
*Department of Measurement and Control*
*University of Kassel*
Kassel, Germany

2nd Andreas Kroll
*Department of Measurement and Control*
*University of Kassel*
Kassel, Germany

***Abstract*—The purpose of this paper is to present a model architecture for the identification of affine linear parameter-varying models in state-space (SS) form. A special case of quasi-LPV identification problem, where the state is part of the scheduling vector but not observed, is considered. In this case the model is always recurrent in the states which demands for an internal dynamics approach. The internal dynamics model will be realized via a Recurrent Neural Network (RNN) with a special structure that allows for transformation into an affine qLPV representation. The proposed approach is based on a structured RNN developed by Lachhab et al. [1]. The original models structures is extended by so-called *gates*, neural network structures which are responsible for the recent success of RNNs on various areas of application in Machine Learning, such as the Long short-term memory (LSTM) network [2] and the Gated Recurrent Unit (GRU) [3]. Through the use of gates the complexity of the models scheduling map and hence its approximation capability is increased while preserving its affine quasi-LPV structure. The performance of the proposed approach is demonstrated by comparison with two other RNN-based approaches on two nonlinear system identification benchmark problems.**

***Index Terms*—System Identification and Modeling; Machine Learning; Nonlinear Systems; LPV**

## I. INTRODUCTION

The linear parameter varying (LPV) framework has proven to be a suitable tool for the modeling and control of highly complex nonlinear systems [4, 1]. Among all synthesis approaches the polytopic approach is the most popular [5]. However, the polytopic synthesis approach requires an affine/polytopic LPV-SS representation of the plant and the implementation complexity of polytopic LPV controllers grows exponentially with the dimension of the time-varying parameters (also referred to as scheduling variables) [5]. Hence, there is a need for LPV representations with as little time-varying parameters as possible. How to arrive at an affine quasi-LPV (qLPV) description of a nonlinear plant is in itself non-trivial. If a physical model is available, procedures for automatically deriving an affine qLPV representation exist [6], but often result in a large number of scheduling variables. This in turn necessitates the application of scheduling dimension reduction (SDR) techniques, for which Deep Learning has recently been applied [7]. Another approach is to identify an affine qLPV-SS model directly from input-output data. Most identification procedures either assume measurable states [8] or full measurement of the scheduling vector [9]. Only few identification methods allow the scheduling vector to depend on unknown states: Verdult [10] proposes a variant of the local linear approach where the weighting functions depend on the unknown state. An RNN with a special strucutre was developed in [11], later a systematic method for deriving an affine qLPV representation was presented by [1]. In this paper both approaches will be evaluated and based on the latter, a deep RNN architecture for affine qLPV system identification will be developed. The proposed deep RNN is potentially capable to approximate nonlinear systems with fewer time-varying parameters compared to the two other approaches, which will be demonstrated on two case studies.

Funded by the federal state of Hesse and the European Regional Development Fund (ERDF 2014-2020), Project: Digital Twin of Injection Molding (DIM) FKZ: 0107/20007409

## II. PRELIMINARIES

An LPV model is a model whose parameters depend on a time-varying parameter vector, or scheduling vector, $\boldsymbol{\rho}_k \in \mathbb{R}^{n_\rho}$. A general LPV plant in SS representation can be written as

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{y}_k \end{bmatrix} = \begin{bmatrix} \mathcal{A}(\boldsymbol{\rho}_k) & \mathcal{B}(\boldsymbol{\rho}_k) \\ \mathcal{C}(\boldsymbol{\rho}_k) & \mathcal{D}(\boldsymbol{\rho}_k) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix} \tag{1}$$

with the state $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$, input $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$, output $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ and matrix-valued functions $\mathcal{A}(\boldsymbol{\rho}_k) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_x \times n_x}$, $\mathcal{B}(\boldsymbol{\rho}_k) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_x \times n_u}$, $\mathcal{C}(\boldsymbol{\rho}_k) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_y \times n_x}$, $\mathcal{D}(\boldsymbol{\rho}_k) : \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_y \times n_u}$, which encode the system matrices dependence on the scheduling vector $\boldsymbol{\rho}_k$. The scheduling vector is a function of $\boldsymbol{\phi}_k$, i.e. $\boldsymbol{\rho}_k = \boldsymbol{\eta}(\boldsymbol{\phi}_k)$ with $\boldsymbol{\eta} : \mathbb{R}^{n_\phi} \to \mathbb{R}^{n_\rho}$ being the scheduling map. If $\boldsymbol{\phi}_k$ contains only exogenous signals, the system is referred to as *pure* LPV, and as *quasi* LPV otherwise [5]. In the general form (1) the system matrices can be arbitrary nonlinear functions of $\boldsymbol{\rho}_k$, which makes

controller synthesis difficult. Hence, usually models with *affine* scheduling dependency are considered:

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{y}_k \end{bmatrix} = \left( \begin{bmatrix} \boldsymbol{A}_0 & \boldsymbol{B}_0 \\ \boldsymbol{C}_0 & \boldsymbol{D}_0 \end{bmatrix} + \sum_{i=1}^{n_\rho} \rho_k^i \begin{bmatrix} \boldsymbol{A}_i & \boldsymbol{B}_i \\ \boldsymbol{C}_i & \boldsymbol{D}_i \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix} \quad (2)$$

with $\boldsymbol{\rho}_k = \mathrm{col}(\rho_k^1, \ldots, \rho_k^{n_\rho})$. In the following, the scheduling map $\boldsymbol{\eta}(\boldsymbol{\phi}_k)$ is assumed to be unknown and depend on the input as well as the unknown state, i.e. $\boldsymbol{\phi}_k = \mathrm{col}(\boldsymbol{x}_k, \boldsymbol{u}_k)$.

## III. RNN-BASED IDENTIFICATION METHODS FOR AFFINE QLPV SYSTEMS WITH SCHEDULING ON UNKNOWN STATES

If the states were observed, the LPV identification problem would amount to a simple regression problem. If not, the model is recurrent in the unknown states. Some identification approaches break the recurrence by first estimating a state sequence and then solving the resulting non-recurrent regression problem, e.g. [9, 12]. However, in these works the estimation of a state sequence relies on knowledge of $\boldsymbol{\rho}_k$. If $\boldsymbol{\rho}_k$ itself is assumed to depend on the unknown state, the model can only be trained in the so-called parallel configuration. This corresponds to identifying a nonlinear output error (NOE) model [13].

In this section two RNN-based approaches for affine qLPV identification which allow for scheduling with unknown states are shortly revisited and their major advantages and shortcomings are pointed out. Based on this analysis a RNN that alleviates some of the identified shortcomings is proposed.

### *A. Radial Basis Function Recurrent Neural Network (RBF-RNN)*

The approach of approximating nonlinear systems via a weighted combination of local linear models has been successfully applied in countless engineering applications, see [10] and the references therein. A local linear SS model can be written as

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{y}_k \end{bmatrix} = \sum_{i=1}^{n_p} p_i(\boldsymbol{\phi}_k) \begin{bmatrix} \boldsymbol{A}_i & \boldsymbol{B}_i \\ \boldsymbol{C}_i & \boldsymbol{D}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix} \quad (3)$$

with $\boldsymbol{A}_i \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B}_i \in \mathbb{R}^{n_x \times n_u}$, $\boldsymbol{C}_i \in \mathbb{R}^{n_y \times n_x}$ and $\boldsymbol{D}_i \in \mathbb{R}^{n_y \times n_u}$. $p_i : \mathbb{R}^{n_\phi} \to \mathbb{R}$ are weighting functions of the local models. The model (3) is similar to the widely used Takagi-Sugeno (TS) fuzzy model, where the weighting functions are interpreted as fuzzy membership functions. It should be pointed out, that in this setting (3) can only be identified using global identification methods due to scheduling with the unknown state. This also implies that the local models can in general not be interpreted as local linearizations. It is common to define $p_i$ as normalized radial basis functions (RBF) with centers $\boldsymbol{c_i}$ and widths $\boldsymbol{w}_i$.

$$p_i = \frac{e^{-(\boldsymbol{\phi}_k - \boldsymbol{c}_i)^T \mathrm{diag}(\boldsymbol{w}_i)(\boldsymbol{\phi}_k - \boldsymbol{c}_i)}}{\sum_{i=1}^{n_\theta} e^{-(\boldsymbol{\phi}_k - \boldsymbol{c}_i)^T \mathrm{diag}(\boldsymbol{w}_i)(\boldsymbol{\phi}_k - \boldsymbol{c}_i)}}. \quad (4)$$

Then (3) can also be considered as an RNN with a special structure and normalized RBFs as activation functions. Defining the weighting functions $p_i$ as scheduling variables $\rho_i$, one arrives at an affine qLPV representation. The local linear approach has been shown to be a universal approximator and can be sensibly initialized: By initializing all local models as a globally identified linear model the initial nonlinear model is identical to the linear model [10]. However, due to the relatively rigid scheduling map many local models, i.e. scheduling variables, might be necessary for modeling highly nonlinear systems. Also the number of model parameters scales badly with $\dim(\boldsymbol{\theta}_k)$, since increasing $\dim(\boldsymbol{\theta}_k)$ by one corresponds to adding a fully parameterized linear SS model.

### *B. Structured RNN (S-RNN) for affine qLPV system identification*

The model equations of the S-RNN for affine qLPV identification developed by Lachhab et al. [1] are given by

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{A}_1 \boldsymbol{f}_1\left(\boldsymbol{E}_1 \boldsymbol{x}_k\right) + \boldsymbol{B}_1 \boldsymbol{f}_2\left(\boldsymbol{E}_2 \boldsymbol{u}_k\right) \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{C}_1 \boldsymbol{f}_3\left(\boldsymbol{E}_3 \boldsymbol{x}_k\right) \end{aligned} \quad (5)$$

with $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B} \in \mathbb{R}^{n_x \times n_u}$, $\boldsymbol{A}_1 \in \mathbb{R}^{n_x \times n_{\theta_1}}$, $\boldsymbol{B}_1 \in \mathbb{R}^{n_x \times n_{\theta_2}}$, $\boldsymbol{E}_1 \in \mathbb{R}^{n_{\theta_1} \times n_x}$, $\boldsymbol{E}_2 \in \mathbb{R}^{n_{\theta_2} \times n_u}$, $\boldsymbol{C} \in \mathbb{R}^{n_y \times n_x}$, $\boldsymbol{C}_1 \in \mathbb{R}^{n_x \times n_{\theta_3}}$, $\boldsymbol{E}_3 \in \mathbb{R}^{n_{\theta_3} \times n_x}$, $\boldsymbol{f}_1 : \mathbb{R}^{n_{\theta_1}} \to \mathbb{R}^{n_{\theta_1}}$, $\boldsymbol{f}_2 : \mathbb{R}^{n_{\theta_2}} \to \mathbb{R}^{n_{\theta_2}}$ and $\boldsymbol{f}_3 : \mathbb{R}^{n_{\theta_3}} \to \mathbb{R}^{n_{\theta_3}}$. $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ and $\boldsymbol{f}_3$ are nonlinear activation functions, which cannot be chosen arbitrarily, as will be shown in the following. In order to rewrite the model (5) in affine qLPV representation, the time-varying parameters $\boldsymbol{\theta}_k$ have to be defined as

$$\begin{aligned} \left[\boldsymbol{\rho}_1\right]_i &= \boldsymbol{\eta}_1\left(\boldsymbol{\phi}_1\right) = \frac{f_1\left(\boldsymbol{e}_{1,i}\boldsymbol{x}\right)}{\boldsymbol{e}_{1,i}\boldsymbol{x}}, \qquad i = 1, \ldots, n_{\theta_1} \\ \left[\boldsymbol{\rho}_2\right]_i &= \boldsymbol{\eta}_2\left(\boldsymbol{\phi}_2\right) = \frac{f_2\left(\boldsymbol{e}_{2,i}\boldsymbol{u}\right)}{\boldsymbol{e}_{2,i}\boldsymbol{u}}, \qquad i = 1, \ldots, n_{\theta_2} \\ \left[\boldsymbol{\rho}_3\right]_i &= \boldsymbol{\eta}_3\left(\boldsymbol{\phi}_3\right) = \frac{f_3\left(\boldsymbol{e}_{3,i}\boldsymbol{x}\right)}{\boldsymbol{e}_{3,i}\boldsymbol{x}}, \qquad i = 1, \ldots, n_{\theta_3} \end{aligned} \quad (6)$$

$\boldsymbol{e}_{j,i}$ denotes the $i$-th row of the matrix $\boldsymbol{E}_j$. With these definitions, (5) can be rewritten in affine qLPV form:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \left( \boldsymbol{A} + \sum_{i=1}^{n_{\theta_1}} \boldsymbol{A}_1 \boldsymbol{e}_{1,i} \left[\boldsymbol{\rho}_{1,k}\right]_i \right) \boldsymbol{x}_k + \\ &\qquad \left( \boldsymbol{B} + \sum_{i=1}^{n_{\theta_2}} \boldsymbol{B}_1 \boldsymbol{e}_{2,i} \left[\boldsymbol{\rho}_{2,k}\right]_i \right) \boldsymbol{u}_k \\ \boldsymbol{y}_k &= \left( \boldsymbol{C} + \sum_{i=1}^{n_{\theta_3}} \boldsymbol{C}_1 \boldsymbol{e}_{3,i} \left[\boldsymbol{\rho}_{3,k}\right]_i \right) \boldsymbol{x}_k \end{aligned} \quad (7)$$

Compared to the local linear approach (5) distinguishes between three sets of scheduling variables $\boldsymbol{\rho}_j$. This adds some flexibility, albeit at the cost of increasing the number of scheduling variables. (5) also scales better in terms of the number of model parameters: Increasing $\dim(\boldsymbol{\rho}_1)$, $\dim(\boldsymbol{\rho}_2)$ or $\dim(\boldsymbol{\rho}_3)$ amounts to merely adding $2n_x$, $n_x + n_u$ or $n_y + n_x$ model parameters, respectively. Also stability conditions can be derived, which to the authors best knowledge is not possible for the general local linear approach [10]. However, the definition of the scheduling maps in (6) is very restrictive and causes some problems. The division by $\boldsymbol{f}_j$'s argument, which

is necessary to preserve the models equivalence to an affine qLPV representation, causes discontinuities at

$$\lim_{\boldsymbol{\phi}_j \to \mathbf{0}} \left[\boldsymbol{\rho}_j\right]_i, \forall j, i.$$

These discontinuities are only removable, if the denominators in (6) approach zero, i.e. $\boldsymbol{f}_j(0) = 0$. This can be verified via L'Hospital's Rule. Choices for $\boldsymbol{f}_{1,2,3}$ are thus restricted to activation functions crossing the origin. The omission of the bias and the division by $\boldsymbol{f}_j$'s own argument severely restricts the scheduling maps that can be represented by (6): A linear activation function results in $\rho_i = 1 \ \forall \ i$ , i.e. an LTI model. A ReLU activation results in $\rho_i$ being either zero or one $\rho_i \in \{0, 1\}0 \ \forall \ i$. This may remind of a piecewise linear approximation (PWA). The difference is, that in this case more than one $\rho_i$ can be active for a given $\boldsymbol{\phi}_i$. A $\tanh$-activation results in all $\rho_i$ becoming radial basis functions (RBF) with different widths, but centered at the origin due to the omission of a bias. In contrast to the RBF-RNN, the time-varying parameters are not normalized, i.e. $\sum_{i=1}^{n_\rho} \neq 1$. Finally, $\boldsymbol{\rho}_j$ depend on different quantities $\boldsymbol{\phi}_j$, e.g. the input matrix can not be scheduled on the state, which is a serious limitation.

## IV. Gated-RNN (G-RNN) for affine qLPV system identification

The goal is to modify the structure of the S-RNN (5) to achieve a more flexible scheduling map, while preserving the S-RNN's equivalence to an affine qLPV representation, its good scaling behavior and the stability results. To this end, so-called *gates*, neural network structures found in Gated Units such as the LSTM and the GRU, are integrated into the S-RNN. Gates typically consist of two single-layered NNs, where the output of one layer is multiplied element-wise by the activation of the other NN. While gates foremost serve the purpose of alleviating the vanishing gradient problem in RNNs [14], they have also shown to increase the capability to approximate quite complex nonlinear functions with comparably few parameters due to the superposition of different activation functions [15] and mitigate the initialization dependence of RNNs [16]. The proposed gated RNN (G-RNN) is depicted in Fig. 1 and follows the equation

$$\begin{aligned} \boldsymbol{x}_{k+1} =& \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{A}_1 \mathcal{NN}_1(\boldsymbol{x}_k, \boldsymbol{u}_k) \circ \boldsymbol{f}_1\left(\boldsymbol{E}_1 \boldsymbol{x}_k\right) \\ &+ \boldsymbol{B}_1 \mathcal{NN}_2(\boldsymbol{x}_k, \boldsymbol{u}_k) \circ \boldsymbol{f}_2\left(\boldsymbol{E}_2 \boldsymbol{u}_k\right) \\ \boldsymbol{y}_k =& \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{C}_1 \mathcal{NN}_3(\boldsymbol{x}_k, \boldsymbol{u}_k) \circ \boldsymbol{f}_3\left(\boldsymbol{E}_3 \boldsymbol{x}_k\right) \end{aligned} \tag{8}$$

The output of each nonlinear layer is now multiplied element-wise with the output of a NN $\mathcal{NN}_j$. The scheduling maps $\boldsymbol{\rho}_j = \boldsymbol{\eta}_j(\boldsymbol{x}_k, \boldsymbol{u}_k)$ are defined as

$$\begin{aligned} \left[\boldsymbol{\rho}_1\right]_i &= \left[\mathcal{NN}_1(\boldsymbol{x}, \boldsymbol{u})\right]_i \frac{\boldsymbol{f}_1\left(\boldsymbol{e}_{1,i}\boldsymbol{x}\right)}{\boldsymbol{e}_{1,i}\boldsymbol{x}}, \qquad i = 1, \ldots, n_{\theta_1} \\ \left[\boldsymbol{\rho}_2\right]_i &= \left[\mathcal{NN}_2(\boldsymbol{x}, \boldsymbol{u})\right]_i \frac{\boldsymbol{f}_2\left(\boldsymbol{e}_{2,i}\boldsymbol{u}\right)}{\boldsymbol{e}_{2,i}\boldsymbol{u}}, \qquad i = 1, \ldots, n_{\theta_2} \\ \left[\boldsymbol{\rho}_3\right]_i &= \left[\mathcal{NN}_3(\boldsymbol{x}, \boldsymbol{u})\right]_i \frac{\boldsymbol{f}_3\left(\boldsymbol{e}_{3,i}\boldsymbol{x}\right)}{\boldsymbol{e}_{3,i}\boldsymbol{x}}, \qquad i = 1, \ldots, n_{\theta_3} \end{aligned} \tag{9}$$

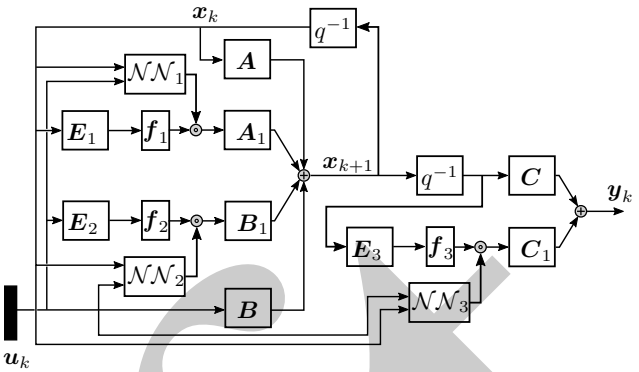

Fig. 1: Structured RNN with *gates* for affine qLPV system identification.

With these definitions the model (8) can be written as in (7). Application of L'Hospital's rule yields

$$\lim_{\boldsymbol{x},\boldsymbol{u} \to \mathbf{0}} \left[\rho_j\right]_i = \mathcal{NN}_j\left(\boldsymbol{x} = \mathbf{0}, \boldsymbol{u} = \mathbf{0}\right) \quad j = 1, 2, 3. \tag{10}$$

I.e. the discontinuities are removable, as long as $\mathcal{NN}_j\left(\mathbf{0}, \mathbf{0}\right)$ exists. This is always the case for a feed-forward NN with bounded activation functions. Besides that, there are no prerequisites regarding $\mathcal{NN}_j$. In particular arbitrary activation functions, biases and any number of layers are admissible. The proposed model also allows scheduling of all SS matrices with the input as well as the unknown state. I.e. (9) is much less restrictive than (7) and hence able to represent a wider range of scheduling maps. Additionally, the G-RNN offers the possibility to increase the approximation capability of the model by adding neurons and/or layers to $\mathcal{NN}_j$, rather than increasing the number of scheduling variables, as is the case with RBF-RNN. This reduces the complexity of the subsequent controller synthesis and implementation step. For these reasons the proposed model structure can offer an advantage in the identification of control-oriented affine qLPV models.

## V. Case Studies

The proposed model structure for quasi LPV system identification will be evaluated on two case studies by comparison.

### A. Silver Box

The Silver Box [17] is an electrical circuit mimicking the behavior of a nonlinear mechanical resonating system with moving mass $m$, viscous damping $d$ and nonlinear spring $k(y)$:

$$m\ddot{y} + d\dot{y} + k(y)y = u.$$

The nonlinear position-dependent spring follows the equation

$$k(y) = a + by^2. \tag{11}$$

The Silver Box was chosen, because it is widely used as benchmark for nonlinear system identification [18] and can be transformed into an affine qLPV representation. Therefore it is possible to gain some notion regarding the parameter

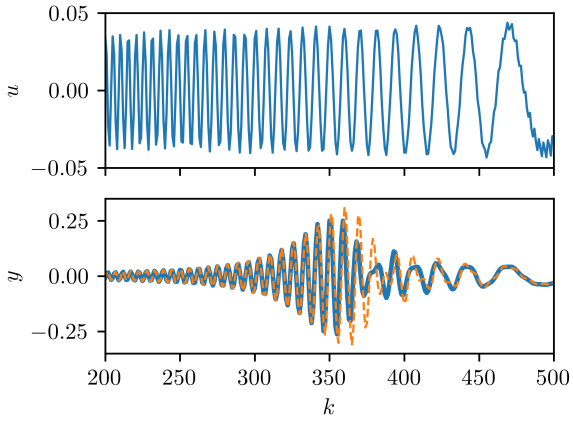


Fig. 2: Multisine signal with Schroeder phases (top) with corresponding system response (—) used for model testing and simulated response of the identified linear subspace model (—) with 46 % BFR (bottom).

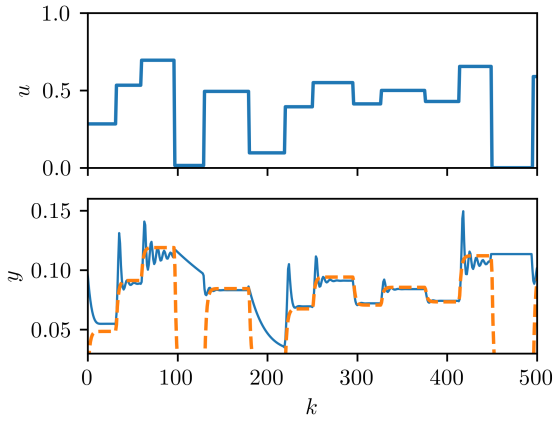


Fig. 3: APRBS signal (top) with corresponding system response (—) used for model testing and simulated response of the identified linear subspace model (—) with 0 % BFR (bottom).

variation, that has to be learned by the RNNs. A affine qLPV representation of the Silver Box is

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \left(\begin{bmatrix}0 & 1\\ -\frac{a}{m} & -\frac{d}{m}\end{bmatrix} + \rho\begin{bmatrix}0 & 0\\ -\frac{b}{m} & 0\end{bmatrix}\right)\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} u \quad (12)$$

with

$$\rho = -x_1^2. \quad (13)$$

Available data were centered and subsequently divided into training, validation and test data in the following manner: One period of a multisine signal with 1342 excited odd harmonics, maximal frequency $f_{\max} \approx 200$ Hz and phases distributed uniformly at random in $[0, 2\pi)$ was used for parameter estimation. The validation signal is white gaussian noise with linearly increasing amplitude. A multisine with Schrœder phases and the corresponding system response, which is partly shown in Fig. 2, was chosen as test signal. The measured data can be considered virtually noise-free.

### B. Bioreactor

A simplified bioreactor model (14) was used as a benchmark for nonlinear system identification and control numerous times, see e.g. [10].

$$\begin{aligned}\dot{x}_1 &= -x_1 u + x_1(1-x_2)e^{x_2/\gamma}\\ \dot{x}_2 &= -x_2 u + x_1(1-x_2)e^{x_2/\gamma}\frac{1+\beta}{1+\beta-x_2}\\ y &= x_1\end{aligned} \quad (14)$$

An exemplary straightforward affine qLPV representation of this plant would have four scheduling variables with $\rho_1 = x_1$, $\rho_2 = x_2$, $\rho_3 = (1-x_2)e^{x_2/\gamma}$ and $\rho_4 = \rho_3(1+\beta)/(1+\beta-x_2)$. The main challenge of this benchmark are the severe nonlinearity and that scheduling with an unobserved state has to be learned. In accordance with common practice parameters were set to $\beta = 0.02$ and $\gamma = 0.48$, the system was simulated using the forward Euler method with a step size of $h = 0.01$ and resulting data were downsampled by a factor of 50. Respectively one data set for training, validation and testing were generated by exciting the system with an Amplitude Modulated Pseudo Binary Random Signal (APRBS). Amplitudes were distributed uniformly at random in [0,0.7], holding times were distributed uniformly at random in [1250,2500]. The signals were designed to prevent pushing the reactor outside the stable region around the equilibrium $\boldsymbol{x}_e = [0.107, 1]^T$, which was also chosen as the initial state for all experiments. Fig. 3 shows the first half of the test data set. No noise was added to the simulated system response, so theoretically a perfect model fit is achievable.

## VI. Identification Procedure and Results

Linear subspace models were identified using the Matlab® implementation of the N4SID-Algorithm. The parameters of the linear model were used to initialize the appropriate parts of the nonlinear models. The remaining parameters were initialized by drawing from a normal random distribution, except for the matrices $\boldsymbol{A}_1$, $\boldsymbol{B}_1$ and $\boldsymbol{C}_1$ of the S-RNN (5) and G-RNN (8). These are required to have a small gain in order for the nonlinear disturbance to be as small as possible and ensure an initially stable model. $\boldsymbol{c}_i$ in (3) were determined from a uniform random distribution over the range of $\boldsymbol{\phi}$ observed during the simulation of the linear model on the training data.

Implementation and gradient-based optimization of all models was performed using CasADi [19] and Ipopt [20]. Each model configuration was initialized 10 times. The optimization procedure was stopped after convergence or a maximum of 1000 iterations. The performance of each model configuration is measured in terms of the Best Fit Rate (BFR) on the test data. The BFR corresponds to the coefficient of determination, but can only take on positive values:

$$\text{BFR} = 100\% \cdot \max\left(1 - \frac{\|\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k\|_2}{\|\boldsymbol{y}_k - \bar{\boldsymbol{y}}\|_2}, 0\right)$$

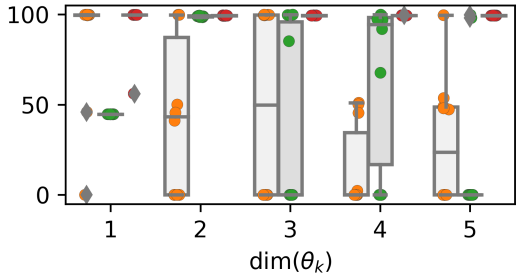


Fig. 4: BFR in % of RBF-RNN (■), S-RNN (■), and proposed G-RNN (■) depending on $\dim(\boldsymbol{\theta}_k)$ on the Silver Box benchmark.

The BFR of the linear subspace model of the Silver Box was 99 % on the training data, but only and 46% on the test data, see Fig. 2. Likewise the BFR for the linear model of the Bioreactor was 64 % on the training data, but 0 % on the test data, see Fig. 3.

Since the input and output matrix of the Silver Box are time invariant $\boldsymbol{\rho}_2$ and $\boldsymbol{\rho}_3$ are omitted in (5) and (8), i.e. $\boldsymbol{\rho} = \boldsymbol{\rho}_1$. $\mathcal{NN}_1$ was designed as a single layer NN with logistic activation functions. Results are summarized in Fig. 4. Since the RBF-RNN with one scheduling variable amounts to a linear model, its performance is equal to the initial linear model. Increasing the number of scheduling variables beyond two, for which the model converges to an almost perfect solution each time, adds unnecessary complexity to the model which impedes convergence to a good local optimum. The S-RNN is, in contrast to the RBF-RNN, able to represent nonlinear behavior with a single scheduling variable. Due to the circumstance that the actual parameter variation of the plant, an inverted centered parabola (13), fits the scheduling map assumed by the model quite well, a centered RBF if tanh is used as activation in (6), the S-RNN achieves an almost perfect fit with only one scheduling variable. However, increasing the number of scheduling variables also hinders convergence to good local optima. The proposed G-RNN on the other hand exhibits exceptional convergence behavior: Out of a total of 50 identified G-RNNs, each had a BFR above 99 %, irrespective of the number of scheduling variables. The property of RNNs with gates to be less sensitive to initialization and choice of hyperparameters has been observed several times, but is still subject of research. Simulation results of the best models are shown in Fig. 5. All three models approximate the plant almost perfectly. At very high amplitudes the simulation error of the G-RNN is slightly higher, a sign that the progressive stiffness of the spring has not been approximated as well. The output map of the Bioreactor (14) is time invariant, therefore $\dim(\boldsymbol{\theta}_3)$ was omitted in (5) and (8). I.e. $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2]$. To enable a fair comparison to the local linear RBF-RNN and present the results in a compact manner, $\dim(\boldsymbol{\theta}_1)$ was chosen to be equal to $\dim(\boldsymbol{\theta}_2)$. $\mathcal{NN}_1$ was chosen to be a two-layered neural network with 5 neurons (tanh-activation) in the hidden layer. $\mathcal{NN}_2$ is a single layer NN with logistic activation. Results

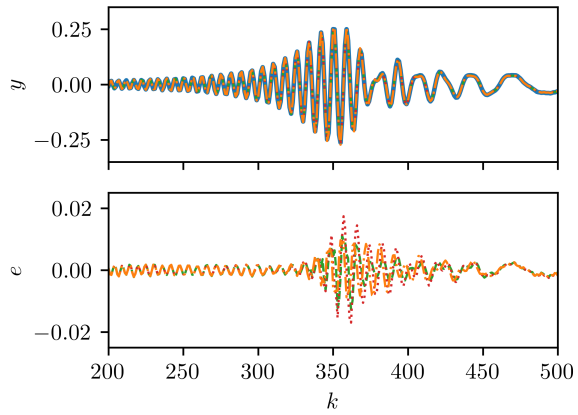


Fig. 5: Top: System response (—) and simulated responses of RBF-RNN (—) with $\dim(\boldsymbol{\theta}) = 3$, S-RNN with $\dim(\boldsymbol{\theta}_1) = 3$ (—) and G-RNN (—) with $\dim(\boldsymbol{\theta}_1) = 1$. Bottom: Simulation error.

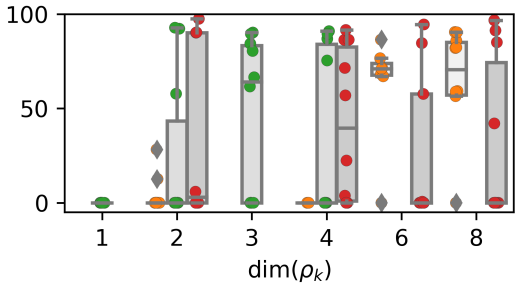


Fig. 6: BFR in % of RBF-RNN (■), S-RNN (■), and proposed G-RNN (■) depending on $\dim(\boldsymbol{\theta}_k)$ Bioreactor benchmark.

are summarized in Fig. 6. The S-RNN fails to find an affine qLPV representation that describes the input-output behavior of the bioreactor with a reasonable number of scheduling variables. Contributing factors are the input entering the state equation of the plant in a nonlinear state-dependent way, which cannot be represented by (6). Also the scheduling maps (6) essentially amount to centered RBFs, which are not suitable to approximate the exponential nonlinearity. Only with a large number of scheduling variables is the S-RNN capable to find an appropriate qLPV representation of the plant. Both the RBF-RNN and G-RNN exhibit high variance in model performance, which reflects the difficulty fo find a good local optimum. However, the proposed G-RNN consistently produces the model with the best fit for any number of scheduling variables. The best identified model is a G-RNN with only two time varying parameters and a BFR of 97 %. Simulation results are shown in Fig. 7. Clearly all models struggle to some extent with the two large downwards steps at $k \approx 100$ and $k \approx 450$. However, while the S-RNNs and the RBF-RNNs simulated responses are considerably off, the proposed G-RNN produces a perfect response for the first step and is also reasonable close to the true response for the second

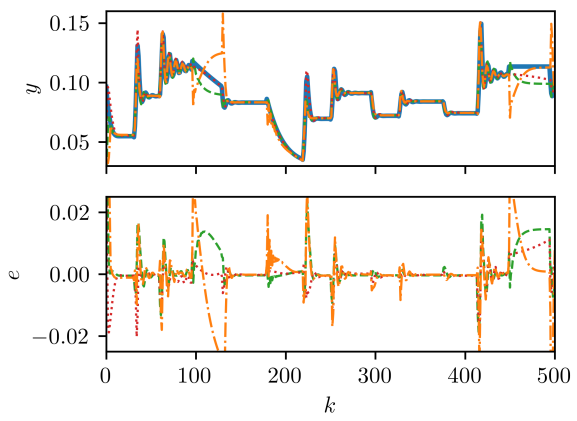


Fig. 7: Top: System response (—) and simulated responses of RBF-RNN (—) with $\dim(\boldsymbol{\theta}) = 2$, S-RNN (—) with $\dim(\boldsymbol{\theta}) = 8$ and G-RNN (—) with $\dim(\boldsymbol{\theta}) = 2$. Bottom: Simulation error.

step.

## VII. Conclusion & Outlook

A (Deep) RNN for affine qLPV identification with scheduling on unknown states was proposed. The proposed RNN achieves a better model fit than comparable approaches with fewer time-varying parameters, which is beneficial for subsequent LPV controller synthesis and implementation. For future research, a systematic way for initializing the nonlinear part of the model should be derived and compared to existing initialization procedures for (Deep) NN. Additionally, a systematic and computationally feasible way for model structure selection, especially with respect to the introduced feed-forward NNs, must be presented to make the procedure applicable for practitioners. Finally, the training procedure assumes the output is disturbed by white measurement noise. In case of correlated noise different identification methods that take into account the correlation of the noise are needed, e.g. see [10].

## References


[1] N. Lachhab, H. Abbas, and H. Werner. "A Neural-Network Based Technique for Modelling and LPV Control of an Arm-Driven Inverted Pendulum". In: *Proc. of CDC*. 2008, pp. 3860–3865.

[2] F.A. Gers, J. Schmidhuber, and F. Cummins. "Learning to forget: Continual prediction with LSTM". In: *Neural computation* 12.10 (2000), pp. 2451–2471.

[3] K. Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *Proc. of EMNLP*. 8. 2014, 1724–1734.

[4] C. Hoffmann and H. Werner. "A Survey of Linear Parameter-Varying Control Applications Validated by Experiments or High-Fidelity Simulations". In: *IEEE Transactions on Control Systems Technology* 23 (2015), pp. 416–434.

[5] C. Hoffmann. "Linear Parameter-Varying Control of Systems of High Complexity". PhD thesis. Technische Universitat Hamburg-Harburg, 2016.

[6] A. Kwiatkowski, M.-T. Boll, and H. Werner. "Automated generation and assessment of affine LPV models". In: *Proc. of CDC*. IEEE. 2006, pp. 6690–6695.

[7] P.J.W. Koelewijn and R. Tóth. "Scheduling Dimension Reduction of LPV Models-A Deep Neural Network Approach". In: *Proc. of ACC*. IEEE. 2020, pp. 1111–1117.

[8] S. Z. Rizvi et al. "A kernel-based approach to MIMO LPV state-space identification and application to a nonlinear process system". In: *IFAC-PapersOnLine* 48.26 (2015), pp. 85–90.

[9] Y. Bao et al. "Identification of state-space linear parameter-varying models using artificial neural networks". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 5286–5291.

[10] V. Verdult. "Nonlinear System Identification: A State-Space Approach". PhD thesis. University of Twente, 2002.

[11] P. Gil et al. "On state-space neural networks for systems identification: Stability and complexity". In: *Proc. of ICORIS*. IEEE. 2006, pp. 1–5.

[12] S. Z. Rizvi et al. "State-space LPV model identification using kernelized machine learning". In: *Automatica* 88 (2018), pp. 38–47.

[13] O. Nelles. *Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes*. Berlin Heidelberg, Germany: Springer, 2020.

[14] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[15] A. Rehmer and A. Kroll. "On the vanishing and exploding gradient problem in Gated Recurrent Units". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 1243–1248.

[16] A. Rehmer and A. Kroll. "On Using Gated Recurrent Units for Nonlinear System Identification". In: *Proc. of ECC*. IEEE, 2019, pp. 2504–2509.

[17] T. Wigren and J. Schoukens. "Three free data sets for development and benchmarking in nonlinear system identification". In: *Proc. of ECC*. Zurich, Switzerland. 2013, pp. 2933–2938.

[18] L. Ljung et al. "Deep learning and system identification". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 1175–1181.

[19] J. A. E. Andersson et al. "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36.

[20] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106.1 (2006), pp. 25–57.