Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

A Neural-Network Based Technique for Modelling and LPV Control of an Arm-Driven Inverted Pendulum

N. Lachhab, H. Abbas and H. Werner

Abstract— **This paper presents a generalization of a recurrent neural-networks (RNNs) approach which was proposed previously in [1], together with stability and identifiability proofs based on the contraction mapping theorem and the concept of sign-permutation equivalence, respectively. A slight simplification of the generalized RNN approach is also proposed that facilitates practical application. To use the RNN for linear parameter-varying (LPV) controller synthesis, a method is presented of transforming it into a discrete-time quasi LPV model in polytopic and linear fractional transformation (LFT) representations. A novel indirect technique for closed-loop identification with RNNs is proposed here to identify a black box model for an arm-driven inverted pendulum (ADIP). The identified RNN model is then transformed into a quasi-LPV model. Based on such LPV models, two discrete-time LPV controllers are synthesized to control the ADIP. The first one is a full-order standard polytopic LPV controller and the second one is a fixed-structure LPV controller in LFT form based on the quadratic separator concept. Experimental results illustrate the practicality of the proposed methods.**

I. INTRODUCTION

Linear parameter-varying (LPV) gain-scheduled control of nonlinear systems has received considerable attention recently as a way of extending linear control techniques to nonlinear systems. However, constructing an LPV model of a nonlinear plant is not trivial. Such a model can often be generated from a nonlinear physical plant model; however, several problems may occur, see [2]. An alternative approach that avoids these problems is to identify an LPV model directly from experimental input/output data. Recurrent Neural-networks (RNNs), as a way to represent a class of nonlinear dynamic systems, are utilized here to identify a nonlinear model from input/output measurements, which is then transformed into an LPV model.

Different structures of dynamic neural-networks have been presented in the literature, see [1], [3], [4] and references therein. They basically contain both feedforward and feedback synaptic connections. We aim in this paper at finding a suitable structure of RNNs that can be easily transformed into discrete-time quasi-LPV models which are suitable for LPV synthesis. The RNNs presented in [1] seem promising in this respect, but somewhat restrictive. For this reason we generalize this class of RNN such that it can represent more general LPV systems. Some fundamental properties of the presented RNNs such as stability and identifiability are investigated, which are essential for practical design of neural-networks models and controllers based on them [4].

Methods for extracting a quasi-LPV model from a feedforward neural-network were proposed e.g. in [5], [2]. On the other hand, only very few references present methods for transforming RNNs into models which can be used for LPV controller synthesis. In [3] RNNs are interpreted as uncertain linear models in LFT representation that can be used for robust controller synthesis, however not for LPV synthesis.

The arm-driven inverted pendulum (ADIP) is a benchmark problem that provides many challenges concerning modeling and control design. Modelling and control of such a plant was considered in [6], where robust and LPV controllers were designed based on a physical model. One of our goals in this work is to obtain from input/output measurements a RNN model of the ADIP and transform it into LPV form.

The main difficulty in the identification stage is the fact that the system is unstable and underactuated. The former means that the system must be identified in closed loop to stabilize it during data collection. Direct and indirect identification are common approaches for system identification in closed loop, see [7]; the latter one is used here. Attractive properties of this identification scheme are that the method does not suffer from bias due to noise correlated with the input signal, as the input signal for identification is taken to be an external reference signal. Several ideas concerning indirect identification have been proposed, see e.g. [7] [8]; some difficulties with this approach are that they typically require stable linear time invariant (LTI) controllers, and that it is based on several impractical assumptions. Here we propose a practical method for extracting a plant model from the closed-loop model, which avoids these difficulties.

Based on the derived quasi-LPV models, two discrete-time LPV controllers have been designed and successfully applied in real-time to the ADIP plant.

This paper is organized as follows. Section II presents the structure of the generalized RNN and considers stability and identifiability issues related to it. In Section III a simplified version of the generalized RNN is transformed into a discrete-time quasi-LPV model in polytopic and LFT representation. The proposed indirect identification approach is illustrated in Section VI. A complete cycle of identification of the ADIP and controller design is presented in Sections V, controller implementation and experimental results are shown in section VI . Finally, conclusions are drawn in Section VII.

N. Lachhab, H. Abbas and H. Werner are with the Institute of Control Systems, Hamburg University of Technology, Eissendorfer Str. 40, 21073 Hamburg, Germany, {nabil.lachhab, hossam.abbas,h.werner}@tu-harburg.de

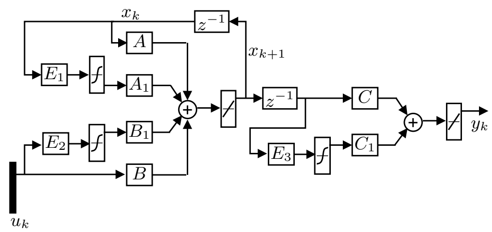

Fig. 1. Proposed general RNN (without the dotted lines: RNN in [1])

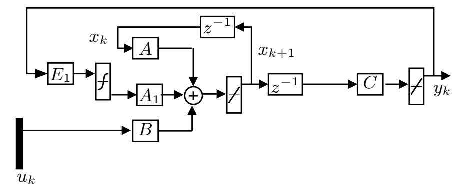

Fig. 2. Modified state-space recurrent neural-network

II. RECURRENT NEURAL-NETWORK MODELS

Our purpose here is to present a neural state-space model that can represent a general class of nonlinear state-space models and that can be easily transformed into a LPV model.

A. Recurrent Neural-Network Topology

In [1] a RNN was presented. The structure of this RNN can be transformed into an LPV model in a systematic way, see Section III, using ideas from [2]. However, it leads to a restricted LPV representation. To see this, write a strictly proper discrete-time LPV model as

$$x_{k+1} = A(\theta_k)x_k + B(\theta_k)u_k, \quad y_k = C(\theta_k)x_k \tag{1}$$

where $A(\theta_k)$, $B(\theta_k)$, $C(\theta_k)$ are continuous mappings (see e.g. [9]), and θ_k is a time-varying parameter. The dependence of the matrices A, B, C in (1) on θ_k represents a general LPV model; imposing that any of these matrices is parameter independent, i.e. fixed, will restrict the generality of the LPV model. The latter is indeed the case when the RNN in [1] is transformed into the LPV model (1): A is parameter dependent but B and C are fixed.

A more general form of an RNN than that presented in [1] is proposed here, see Fig. 1. The main difference between the RNN presented here and that in [1] is that two more hidden layers with sigmoidal activation functions are included in the input and the output paths of the RNN. The discrete-time nonlinear model represented by it is expressed as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + A_1\sigma(E_1x_k) + B_1\sigma(E_2u_k) \\ y_k &= Cx_k + C_1\sigma(E_3x_k) \end{aligned} \tag{2}$$

where $x \in \mathbb{R}^n$ denotes the state vector, $y \in \mathbb{R}^p$ the output and $u \in \mathbb{R}^m$ the input vectors. A, A_1, B, B_1, C, C_1, E_1, $E_2$and E_3, are real valued matrices of appropriate dimensions; they represent the weights which will be adjusted during the training stage of the RNN. The nonlinear activation function $\sigma(.)$, which is applied elementwise in (2), is taken as a continuous, differentiable and bounded function. This RNN leads to a general form of the neural state-space model in the sense that if it is transformed into an LPV model in the form (1), the matrices A, B and C will be parameter dependent.

B. Stability and Identifiability of the Proposed RNN

In this subsection stability and identifiability of the proposed RNN (2) as a dynamic system are investigated.

Based on the contraction mapping theorem [10] and ideas presented in [1], the following sufficient condition for global asymptotic stability of the proposed RNN can be shown:
Theorem II.1: Let $W(A, A_1, B, B_1, C, C_1, E_1, E_2, \ E_3)$ be a given parameterization (weights) for the state-space neural network (2) and define $L = \|E_1\|$. The neural network is contractive [10] and has one equilibrium point if $\|A\| + L\|A_1\| < 1$
Proof: See Appendix A.

Identifiability [11] is concerned with the problem of uniqueness of the weights of the neural network and is related to the question whether two networks with different parameter vectors can produce identical input/output behavior. Work on this topic was done in [11] and [12] for feedforward and recurrent networks, respectively. Based on the concept of sign-permutation equivalence the identifiability of the proposed RNN in (2) is shown in Appendix B.

C. Simplifications and Assumptions

The RNN in (2) can be simplified by removing any of the sigmoidal layers; this can be done according to *a priori* information about the identified system; this is used here in the identification of the ADIP plant.

The LPV model (1) of the ADIP plant, which was derived from a physical model in [6], has the matrices A and B parameter dependent. A specific transformation is used in [6] to have only A parameter dependent. Similarly the general RNN can be simplified: removing the sigmoidal layers from the input and the output paths means that the resulting LPV model has parameter dependence in matrix A only. Moreover, for a practical implementation this simplified RNN is modified as shown in Fig. 2: the outputs instead of the states are taken as input to the sigmoidal layer. This modification facilitates the implementation of LPV controllers designed based on this model.The modified RNN is represented as

$$x_{k+1} = Ax_k + Bu_k + A_1\sigma(E_1Cx_k), \quad y_k = Cx_k \tag{3}$$

It can be shown that the stability condition of the modified RNN in (3) is the same as that one given in Theorem II.1 with Lipschitz constant $L = \|E_1C\|$. Moreover, it is straightforward to check that it is identifiable, Appendix B.

III. A QUASI-LPV MODEL FROM AN RNN

A discrete-time LPV system is defined by the parameter dependent state-space model (1) and the compact set $\mathcal{P} \subset \mathbb{R}^l : \theta_k \in \mathcal{P} \ \forall k > 0$. The time-dependent parameter vector $\theta_k \in \mathbb{R}^l$ depends on a vector of measurable signals $\rho_k \in$

$\mathbb{R}^h$, referred to as the scheduling signals, according to $\theta_k = f_\theta(\rho_k)$ where $f_\theta : \mathbb{R}^h \to \mathbb{R}^l$ is a continuous mapping. The vector of scheduling signals combines external and internal signals. If the parameters depend on external signals only, the system is referred to as LPV system, otherwise as quasi-LPV system. Moreover, the proposed methods here can be extended systematically to the general RNN (2).

A. Derivation of a discrete-time polytopic quasi-LPV model

An LPV system is called polytopic when it can be represented by state-space matrices $A(\theta_k)$, $B(\theta_k)$, $C(\theta_k)$ and $D(\theta_k)$, where the time varying parameter vector $\theta_k \in \mathbb{R}^l$ ranges over a fixed polytope and the dependence of $A(.)$, $B(.)$, $C(.)$ and $D(.)$ on θ_k is affine, where $\theta_k = f_\theta(\rho_k)$ as above. The time-varying parameter θ_k varies in a polytope $\mathcal{Q}$ of vertices $v_1, v_2, \dots, v_r$; that is, $\theta_k \in \mathcal{Q} := Co\{v_1, v_2, \dots, v_r\}$, where Co denotes convex hull.

It is required here to transform the RNN model in (3) into a discrete-time polytopic quasi-LPV model with $y_k = Cx_k$,

$$x_{k+1} = \sum_{i=1}^{r} \alpha_{k,i} A_i x_k + Bu_k = A(\theta_k)x_k + Bu_k \quad (4)$$

where $$A(\theta_k) \in \mathcal{P} := Co\{A_i; i = 1, \dots, r\} \quad (5)$$

Now, define the time-varying parameters

$$\theta_k^j = \sigma(E_1^j y_k)/(E_1^j y_k), \text{if } y_k = 0 \to \theta_k^j = 1 \quad (6)$$

for $1 \le j \le l$, where E_1^j denotes the j[th] row in the hidden layer weight matrix which contains the sigmoid activation functions, l the total number of the neurons in this layer and $y_k = Cx_k$. Then (3) can be rewritten as

$$x_{k+1} = Ax_k + Bu_k + A_1\Theta_k E_1 Cx_k, \quad y_k = Cx_k \quad (7)$$

where $\Theta_k \in \mathbb{R}^{l\times l}$ is a diagonal matrix that contains the variable parameters of the LPV model. By this way the neural-network model is transformed into a quasi-LPV model in the form of (4), where

$$A(\theta_k) = A + A_1\Theta_k E_1 C \quad (8)$$

The time-varying parameters can be collected into a vector $\theta \in \mathbb{R}^l$ whose elements are contained in $[\underline{\theta}^j \ \bar{\theta}^j]$ where

$$\underline{\theta}^j = \min_{0\le k\le T} \theta^j, \quad \bar{\theta}^j = \max_{0\le k\le T} \theta^j \quad (9)$$

for $1 \le j \le l$ where $k \in [0 \ T]$ is the time interval in which the training data have been acquired. Note that $0 \le \underline{\theta}^j, \bar{\theta}^j \le 1$, so no further scaling is required. Thus the neural state-space model (3) is transformed into a quasi-LPV model in a polytope representation (4) that satisfies (5), where $A(\theta)$ is given by (8) and the time varying parameter vector is defined by (6) and its bounds are given by (9).

B. Derivation of a discrete-time LFT quasi-LPV model

The problem considered in this subsection is to represent the nonlinear model (3) in the form of an affine discrete-time quasi-LPV model in LFT form

$$\begin{aligned} x_{k+1} &= A_0 x_k + B_\delta w_k^\delta + B_0 u_k \quad (10)\\ w_k^\delta &= \Delta_k z_k^\delta,\ \Delta_k = diag(\delta_k^1, \dots, \delta_k^l) \in \mathbb{R}^{l\times l},\ \|\Delta_k\| < 1\\ z_k^\delta &= C_\delta x_x + D_{\delta,1}u_k,\ y_k = C_0 x_k + D_{1,\delta}w_k^\delta + D_0 u_k \end{aligned}$$

The parameter matrix Δ_k has a diagonal structure, where the parameters δ_k serve as the scheduling signals in the LPV model. An affine LPV model in an upper LFT representation for system (10) results in the block-matrix representation

$$\begin{bmatrix} A(\delta) & B(\delta) \\ C(\delta) & D(\delta) \end{bmatrix} = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} + \begin{bmatrix} B_\delta \\ D_{1,\delta} \end{bmatrix} \Delta_k \begin{bmatrix} C_\delta & D_{\delta,1} \end{bmatrix} \quad (11)$$

With the time-varying parameters defined in (6), (3) can be rewritten as (7). By comparing (7) and (11) we find that $A_0 = A$, $B_0 = B$, $C_0 = C$, $B_\delta = A_1$, $C_\delta = E_1 C$ $D_0 = 0$, $D_{\delta,1} = 0$, $D_{1,\delta} = 0$ and $\Theta_k = \Delta_k$. Thus (7) is equivalent to (10) with $\|\Theta_k\| < 1$. This is the case because each element of Θ_k is contained in $[0 \ 1]$, i.e. $\theta_k^j \in [0 \ 1]$, see (6). But this bound on Θ_k leads to a conservative LFT representation in (7). To remove that conservativeness, Θ_k can be rescaled such that its elements are contained in $[-1 \ 1]$, and this rescaling can be done as follows. Let us define a new parameter $\delta_k^j \in [-1 \ 1]$, then δ_k^j in terms of θ_k^j and its bounds $\underline{\theta}^j$ and $\bar{\theta}^j$ and θ_k^j in terms of δ_k^j, respectively, for neuron j, can be expressed as

$$\delta_k^j = \frac{2\theta_k^j}{\bar{\theta}^j - \underline{\theta}^j} - \frac{\bar{\theta}^j + \underline{\theta}^j}{\bar{\theta}^j - \underline{\theta}^j}, \quad \theta_k^j = \delta_k^j \frac{\bar{\theta}^j - \underline{\theta}^j}{2} + \frac{\bar{\theta}^j + \underline{\theta}^j}{2} \quad (12)$$

By constructing a new diagonal matrix $\Delta_k \in \mathbb{R}^{l\times l}$ that contains the new time-varying parameters $\delta_k^1, \dots, \delta_k^l$ and substituting in (7) we get

$$x_{k+1} = Ax_k + Bu_k + A_1\underline{\Theta}(\Delta_k\underline{\Theta} + \bar{\Theta})\underline{\Theta}^{-1}E_1 Cx_k \quad (13)$$

and $y_k = Cx_k$, where $\underline{\Theta}$ and $\bar{\Theta}$ are fixed diagonal matrices that contain $(\bar{\theta}^1 - \underline{\theta}^1)/2, \dots, (\bar{\theta}^l - \underline{\theta}^l)/2$ and $(\bar{\theta}^1 + \underline{\theta}^1)/2, \dots, (\bar{\theta}^l + \underline{\theta}^l)/2$, respectively. Now letting $A_0 = A + A_1\underline{\Theta}\bar{\Theta}\underline{\Theta}^{-1}E_1 C$, $B_0 = B$, $B_\delta = A_1\underline{\Theta}$, $C_0 = C$, $C_\delta = E_1 C$, $y_k = Cx_k$, model (10) becomes an LFT representation of the nonlinear RNN with $D_0 = 0$, $D_{\delta,1} = 0$ and $D_{1,\delta} = 0$.

IV. CLOSED-LOOP IDENTIFICATION USING A STRUCTURED RECURRENT NEURAL-NETWORK

Closed-loop system identification is required when the system to be identified is unstable. In indirect identification the closed-loop system from the reference input to the output is first identified, then the open-loop system is retrieved, using the knowledge of the regulator that stabilizes the closed-loop system. Its basic advantage is that the dynamics of the model can be correctly estimated without estimating any noise model. Here we propose a novel and simple approach for indirect identification that allows selection of the model order and can be used when the stabilizing controller for the closed-loop identification has a complex

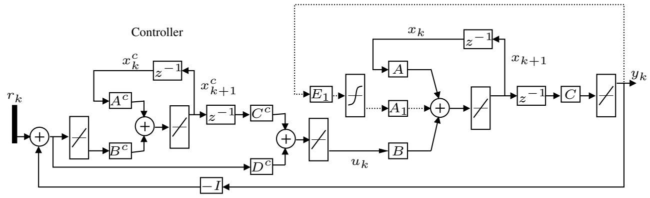

Fig. 3. Structured RNN for indirect identification, the dotted line indicates identification of a nonlinear model

structure and it is possibly unstable. The idea here is based on identifying the closed-loop system using a structured RNN, see Fig. 3, which basically consists of two RNN: the left one is linear with fixed and known weights and represents the controller, whereas the right one is the plant RNN discussed in Section II that has unknown weights. The stability condition for this structured RNN is the same as that in Theorem II.1, namely $\|\tilde{A}\| + L\|\tilde{A}_1\| < 1$. Here the Lipschitz constant $L = \|\tilde{E}_1\|$, $\tilde{A}$, $\tilde{A}_1$ and $\tilde{E}_1$ are given by $\tilde{A} = [A - BD^cC \;\; BC^c; -B^cC \;\; A^c], \tilde{A}_1 = [A_1 \;\; 0]^T, \tilde{E}_1 = E_1[C \;\; 0]$ where A^c, B^c, C^c, D^c are the controller matrices, and A, B, C, A_1 and E_1 are given in (3). Now, assume first that the system to be identified is linear and strictly proper and that it is stabilized by a linear discrete-time controller. Then the closed-loop system, Fig. 3, can be compactly written as

$$x_{k+1}^{cl} = A_{cl}x_k^{cl} + B_{cl}r_k, \quad z_k = C_{cl}x_k^{cl} + D_{cl}r_k \qquad (14)$$

where $A_{cl} = \tilde{A}$, $B_{cl} = [BD^c \;\; B^c]^T$, $C_{cl} = C$, $D_{cl} = 0$. A^c, B^c, C^c, D^c are the controller matrices, and A, B, C are the matrices of the linear plant model which will be identified. Then the structured RNN that represents the closed-loop system in (14) can be constructed as shown in Fig. 3 without the dotted line, where the controller matrices are represented by known fixed weights and the matrices of the linear model by unknown weights. By using the representation in Fig. 3 we can extract directly the system matrices A, B and C once the training is completed. A nonlinear model can be identified in the same way by including the dotted lines and the sigmoidal layer in Fig. 3. Furthermore, it is possible to use an LPV controller to stabilize the plant during gathering the data. The method proposed here can be easily extended such that the structured RNN contains the general RNN in (2) instead of the modified one.

V. IDENTIFICATION OF AN ADIP MODEL

The ADIP used here is the planar rotary inverted pendulum manufactured by Quanser Inc. It is an underactuated system that has one input (the current controlling the drive of the arm) and two outputs (the angular-positions of the arm and the pendulum, ϕ_1, ϕ_2 respectively). The control loop shown in Fig. 4 is used to stabilize the ADIP during all stages of the indirect identification procedure. A differentiator (FD in Fig 4) is used to determine the angular velocity $\dot{\phi}_1$. A PI controller shown in Fig 4 is used here as discussed in [6]. The PI controller and the filter FD are considered as part of the plant; the system to be identified is the dashed box in Fig. 4. The physical ADIP model is 4$^{\text{th}}$-order, whereas the plant to be identified is 7$^{\text{th}}$-order.

Generating an input signal to excite all different dynamics of the plant is the initial step of the system identification - a proper choice of this signal is crucial for the quality of the model. The required operating ranges, the sampling frequency and the bandwidth of the system need to be considered to design a sufficiently rich training signal. Two uncorrelated multilevel random signals r and d in Fig. 4, are designed to excite the system. These signals enter the closed-loop system, one as a reference for the first output and the other as a disturbance to the second output, see Fig. 4. In each stage of the successive identification procedure, the input-output data are collected and divided into training and validation sets. The resulting models are validated through simulation of the RNN in closed loop. To stabilize the plant in the required range, the following successive identification procedure was used. Initially a continuous-time LTI $\mathcal{H}_\infty$ optimal controller based on a continuous-time model of the plant - linearized around the zero position - is designed. The following parameters were used to increase the stability range of the ADIP during data gathering: the closed-loop bandwidth, which is tuned by tuning the controller, the frequency and the levels of the signals d and r, Fig. 4. By using these tuning knobs, closed-loop input-output data in a range of $(\pm 45^o)$ could be acquired. Then, applying the indirect approach proposed in Section IV, a discrete-time linear model from the linear RNN with the same order as the overall ADIP system, can be easily found to represent the plant in this range. A new LTI $\mathcal{H}_\infty$ optimal controller is designed - this time in discrete time - for the new discrete-time LTI model that represents the behavior of the plant in a wider range. The same tuning knobs are used here to stabilize the ADIP in a range of $(\pm 51^o)$ during a new data gathering. Again the proposed indirect approach is applied to find a new discrete-time linear model from the latest acquired data, based on which a new LTI discrete-time controller is designed. At this point it turns out that the plant cannot be stabilized for a range wider than $(\pm 51^o)$ with a controller whose design is based on a linear model. A nonlinear model is now required to capture the plant behavior. Such a model can be obtained from a RNN by adding a sigmoid $\tanh$ activation function and using the indirect identification procedure described in the previous section, Fig. 3. Part of the validation plot is shown in Fig. 5.

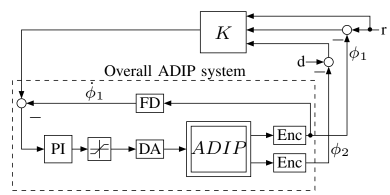

Fig. 4. Overall ADIP system in closed-loop identification

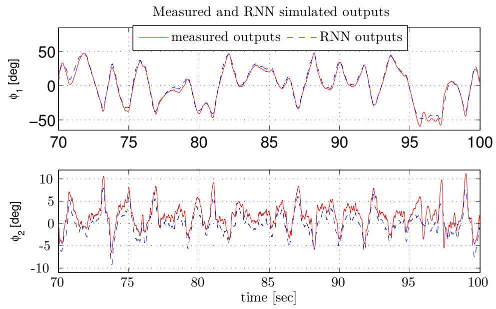

Fig. 5. Part of the validation test of the nonlinear RNN model

VI. LPV SYNTHESIS AND IMPLEMENTATION

After the RNN model of the ADIP is identified, it is transformed into quasi-LPV models in two forms - polytopic and LFT - using the methods proposed in Section III. Both models have one varying parameter, θ_k for the polytopic model and the scaled δ_k for the LFT model, since the RNN contains only one sigmoid activation function.

A. Full-order polytopic discrete-time LPV controller

The resulting polytopic model has two vertices, one at the zero position and the other one at ϕ_t (target angular-position). Two weighting filters were chosen at each vertex, one to shape the sensitivity at the zero and the target position, the other one to shape the control sensitivity. A polytopic discrete-time LPV controller is then designed using the approach in [13]. Note that this approach is based on a fixed Lyapunov function for the whole operating range. The $9th$ order controller achieves an induced $\mathcal{L}_2$-norm of $\gamma = 0.35$. The varying parameter θ is calculated from (6); the bounds are $\bar{\theta} = 1$ and $\underline{\theta} = 0.748$ at the zero and the target position, respectively. The LPV discrete-time controller was experimentally implemented on the ADIP plant; the results are shown in Fig. 6a. One can see that a good performance is achieved over the operating range ($\pm 60^o$). The scheduling parameter θ in Fig. 6 shows that the controller is scheduled according to the operating position.

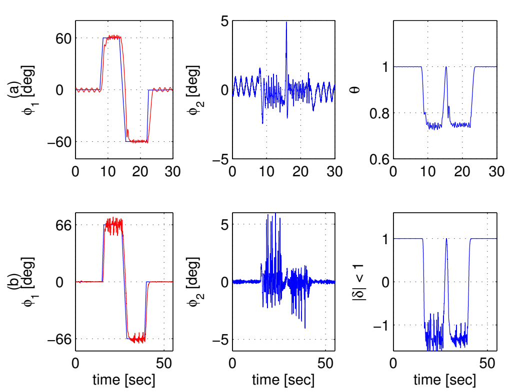

Fig. 6. Experimental results, LPV (a) Full-order, (b) Fixed-structure

B. Fixed-structure LFT discrete-time LPV controller

Next a fixed-structure controller is designed. The design is now based on the idea of quadratic separators, as discussed in [14] - the existence of which is equivalent to the existence of a parameter-dependent Lyapunov function [15]. This allows to take bounds on the rate of parameter variation into account. The method in [14] uses a hybrid evolutionary-algebraic approach for solving the non-convex problems of low-order and fixed structure controller design.

The generalized plant, which includes the model of the plant and the weighting filters for performance, depends as well as the controller on the parameter matrix Δ_k in an LFT manner as given in (10). The closed-loop system can be described by the following LFT structure

$$\begin{aligned} x_{k+1} &= Ax_k + B_\Delta w_k^\Delta + B_p w_k^p, \quad z_k^p = C_p x_k + D_p w_k^p \\ z_k^\Delta &= C_\Delta x_k + D_\Delta w_k^\Delta, \quad w_k^\Delta = \Delta_k z_k^\Delta \end{aligned} \tag{15}$$

where w_k^Δ, $z_k^\Delta \in \mathbb{R}^l$, $w_k^p \in \mathbb{R}^d$, $z_k^p \in \mathbb{R}^v$, and $\Delta_k = diag(\Delta_k, \Delta_k)$, [9], where each $\Delta_k \in \mathbb{R}^{l\times l}$. The dimension of Δ_k is equal to the number of neurons in the sigmoidal layer of the RNN. In [14] a condition on the induced $\mathcal{L}_2$-norm was derived as well as a synthesis approach to design fixed structure LPV controller, which has been used here.

The structure of the controller is selected based on the frequency response of the full-order controller which was designed in the previous subsection. It turns out that the controller shows essentially PD behaviour. Hence, a discrete-time LPV-PD structure is used. The same weighting filters were used here an in the full-order case. The synthesis procedure in [14] is then applied to find and tune the coefficients of the parameter dependent controller. The smallest induced $\mathcal{L}_2$-norm that could be achieved was 0.636, which indicates that stability as well as the required performances can be guaranteed for the whole parameter set. Then the designed discrete-time LPV-PD controller (3rd order) is experimentally tested on the real ADIP plant; the results are shown in Fig. 6b. The LPV-PD controller is able to stabilize the plant in a range of up to ($\pm 66^o$) as shown in Fig. 6b. This range is wider than the one achieved one with the full-order controller (compare Fig. 6a and b), which was based on a fixed Lyapunov function. This indicates the benefit of a design with a parameter dependent Lyapunov function.

VII. CONCLUSION

A general RNN model is presented here along with stability and identifiability proofs. For practical application to the ADIP plant the RNN has been simplified. An indirect identification approach for closed-loop identification based on a structured RNN is presented. The structured RNN in this approach is trained to identify the closed-loop system from the reference to the output signals, where the controller parameters are represented as fixed weights and the parameters of the plant model as unknown weights. The open-loop model can then be easily extracted from the identified closed-loop model. Methods for transforming an RNN into discrete-time polytopic and LFT quasi-LPV models are presented and illustrated with experimental results from application to an arm-driven inverted pendulum (ADIP). Based on the transformed LPV models, full-order and fixed-structure discrete-time LPV controllers have been designed and implemented in real-time successfully on the ADIP plant.

APPENDIX

A. Proof of theorem II.1

Based on the concepts of Lipschitz continuity, the mean value theorem, the contraction mapping theorem, (see e.g. [10]) and ideas presented in [1], Theorem II.2 can be shown as follows. Let us start by transforming the neural network (2) into a new coordinate system where the origin is moved to an equilibrium point. The new state vector is thus rewritten as $z_k = x_k - x^e$. Let $u_k = \bar{u}$ correspond to the equilibrium point x^e. Taking into account z_k, the RNN (2) becomes

$$\begin{aligned} z_{k+1} = Az_k + B(u_k - \bar{u}) + A_1\{\sigma\big(E_1(z_k + x^e)\big) \\ - \sigma(E_1 x^e)\} + B_1\{\sigma(E_2 u_k) - \sigma(E_2 \bar{u})\} \end{aligned} \quad (16)$$

Without loss of generality, let us assume a constant input $u_k = \bar{u}$. Then (16) can be written as

$$z_{k+1} = Az_k + A_1\{\sigma\big(E_1(z_k + x^e)\big) - \sigma(E_1 x^e)\} \quad (17)$$

Now defining a continuous nonlinear function $\Phi : \mathbb{R}^n \to \mathbb{R}^n$, $\Phi(z) = \sigma\big(E_1(z_k + x^e)\big) - \sigma(E_1 x^e)$ such that $\Phi(0) = 0$, (17) can be rewritten as $z_{k+1} = Az_k + A_1\Phi(z)$. This expression describes an autonomous system for which $z = 0$ represents the equilibrium point. Now by applying the mean value theorem to the nonlinear part of (2) we have $\|\sigma(E_1 x) - \sigma(E_1\bar{x})\| \leq L\|x - \bar{x}\|$. The Lipschitz constant $L = \max_y \|\sigma'(E_1 y)\|$ where y is a point on the straight line between x and $\bar{x}$. With $\sigma(.) = \tanh(.)$ as defined above, one can conclude that $L \equiv \|E_1\|$. (This is the main difference between the proof here and the one given in [1], where $L \equiv \max|e_{ij}|$, e_{ij} is an entry of the matrix E_1). For two arbitrary vectors z and $\bar{z}$ the following inequality holds: $\|\Phi(z) - \Phi(\bar{z})\| = \|\sigma\big(E_1(z + x^e)\big) - \sigma\big(E_1(\bar{z} + x^e)\big)\| \leq L\|z - \bar{z}\|$ Defining $\Psi(z) \equiv z_{k+1} = Az_k + A_1\Phi(z_k)$ yields $\|\Psi(z) - \Psi(\bar{z})\| = \|A(z - \bar{z}) + A_1\big(\Phi(z) - \Phi(\bar{z})\big)\|$ $\leq \big(\|A\| + L\|A_1\|\big)\|z - \bar{z}\|$. From the contraction mapping theorem the RNN has one asymptotic equilibrium point if $\|A\| + L\|A_1\| < 1$ □

B. Proof of identifiability of the proposed RNN

Based on the concepts of sign-permutation, permutation equivalent, irreducible neural networks and theorems given in [11] and [12] we have the following

Corollary 1 [16]: The two feedforward networks: $y = \tanh(x)$ and $y = T\tanh(T^{-1}x)$, where $T = PJ$ with P a permutation matrix and $J = diag\{\pm 1\}$, are input/output equivalent.

Starting from this equivalence it is straightforward to apply the results in [12] to the proposed RNN here to check the identifiability.

Corollary 2: Following the same principle as in corollary 1 we have the following input/output equivalent form of the presented RNN in (2)

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k + A_1\tanh(E_1 z_k) + B_1\tanh(E_2 u_k) \\ y_k &= Cx_k + C_1\tanh(E_3 z_k) \end{aligned} \quad (18)$$

Using the equivalence in Corollary 1, one obtains

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k + A_1 T_1 \tanh(T_1^{-1}E_1 z_k) \\ &\quad + B_1 T_2 \tanh(T_2^{-1}E_2 u_k), \\ y_k &= Cx_k + C_1 T_3 \tanh(T_3^{-1}E_3 z_k) \end{aligned} \quad (19)$$

with $T_i = P_i J_i$ $(i = 1, 2, 3)$, P_i a permutation matrix and $J_i = diag\{\pm 1\}$ and $T_1 \in \mathbb{R}^{l_x \times l_x}$, $T_3 \in \mathbb{R}^{l_u \times l_u}$ and $T_3 \in \mathbb{R}^{l_y \times l_y}$ $(l_u, l_y, l_x$ are the number of the sigmoidal activation functions in the sigmoidal layers in the input, output paths and the state of the proposed RNN, respectively, see Fig. 1). Then putting $z_k = Sx_k$ with $S \in \mathbb{R}^{n\times n}$ full rank, one has

$$\begin{aligned} x_{k+1} &= \hat{A}x_k + \hat{B}u_k + \hat{A}_1\tanh(\hat{E}_1 x_k) + \hat{B}_1\tanh(\hat{E}_2 u_k) \\ y_k &= \hat{C}x_k + \hat{C}_1\tanh(\hat{E}_3 x_k) \end{aligned} \quad (20)$$

with $\hat{A} = S^{-1}AS$, $\hat{B} = S^{-1}B$, $\hat{A}_1 = S^{-1}A_1T_1$, $\hat{E}_1 = T_1^{-1}E_1S$, $\hat{B}_1 = S^{-1}BT_2$, $\hat{E}_2 = T_2^{-1}E_2$, $\hat{C} = SC$, $\hat{C}_1 = C_1T_3$ and $\hat{E}_3 = T_3^{-1}E_3S$. This is the sign-equivalence in [12] and shows that the RNN model is unique up to a similarity transformation and sign reversal. This shows that it is identifiable.

REFERENCES

[1] P. Gil, J. Henriques, A. Dourade, H. Duarte-Ramos, "On state space neural networks for systems identification: Stability and complexity," in *IEEE Conference on Cybernetics and Intelligent Systems*, 2006.

[2] H. Abbas and H. Werner, "Polytopic quasi-LPV model based on neural state-space models and application to air charge control of a SI engine," in *Proc. IFAC World Congress*, 2008, pp. 6466–6471.

[3] J. A. K. Suykens, J. P. L. Vandewalle, and B. L. R. D. Moor, *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1996.

[4] Z. Guan, G. Chen, and Y. Qin, "On equilibria, stability, and instability of hopfield neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 534–540, 2000.

[5] J. Bendtsen and K. Trangbæk, "Roubst quasi-LPV control based on neural state space models," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 355–368, 2002.

[6] H. Kajiwara, P. Apkarian, and P. Gahinet, "LPV techniques for control of an inverted pendulum," *IEEE Control Systems Magazine*, vol. 19, no. 1, pp. 44–54, 1999.

[7] L. Ljung, *System Identification, Theory for the User*, 2nd ed., ser. Prentice Hall Information and System Sciences Series. Prentice-Hall Inc. USA, 1999.

[8] P. M. V. d. Hof and R. A. d. Callafon, "Multivariable closed-loop identification: From indirect identification to dual-youla parametrization," in *Proc. of the 35th IEEE Conference on Decision and Control*, 1996.

[9] P. Apkarian and P. Gahinet, "A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controller," *IEEE Transactions on Automatic Control*, vol. 40, no. 5, pp. 853–864, 1995.

[10] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed., ser. Classics in Applied Mathematics, U. o. W. Robert E. O'Mallex, Jr., Ed. 3600 University City Science Center, Philadelphia: SIAM, 2002.

[11] H. Sussmann, "Uniqueness of the weights for minimal feedforward nets with a given input-output map," *Neural Networks*, vol. 5, pp. 589–593, 1992.

[12] F. Albertini and E. Sontag, "For neural networks, function determines form," *Neural Networks*, vol. 6, pp. 975–990, 1993.

[13] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H_∞ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.

[14] S. S. Chughtai and H. Werner, "Synthesis of low-order controllers for discrete LPV systems using LMIs and evolutionary search," in ., Proceedings of ECC07, 2007.

[15] T. Iwasaki and G. Shibata, "LPV system analysis via quadratic separator for uncertain implicit systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 8, pp. 1195–1208, 2001.

[16] J. Suykens, J. Vandewalle, and B. Moor, *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Kluwer, 1996.