Preprints, 4th IFAC Conference on
Advances in Proportional-Integral-Derivative Control
Almería, Spain | June 12-14, 2024

Automated Tuning of Gain-Scheduled PI Controllers Based on SANARX Models

Fabian Kreutmayr * **Christoph Ament** **

* *HAWE Hydraulik SE, Aschheim, BY 85609 Germany (e-mail: fabian@kreutmayr.de).*
** *University of Augsburg, Augsburg, BY 86159 Germany (e-mail: christoph.ament@uni-a.de)*

Abstract: This paper presents a data-driven method to automatically tune gain-scheduled PI controllers for linear parameter-varying (LPV) systems. First, input-output data from a system is used to train a neural network (NN) based simplified additive nonlinear autoregressive exogenous (SANARX) model. After reformulating this into a state space representation, an H_∞ method is used to obtain the PI parameters for any sampled working point. Throughout the entire pipeline, this work uses data from a real-world hydraulic test rig, which also serves as the system where the resulting gain-scheduled controller is evaluated on. Overall, no prior system knowledge is necessary to utilize this method.

Keywords: LPV system identification, Learning for control, Neural networks, H_∞ synthesis, Gain-scheduled PI, Nonlinear systems.

1. INTRODUCTION

While PID controllers are a standard for controlling industrial processes, they also have some shortcomings. As linear controllers, they are not necessarily the ideal choice for the control of nonlinear systems. Additionally, the quality of the parameters may depend on the experience of the person performing the tuning process. The ideal set of parameters might also change over time, e.g. when the working point of the system changes due to temperature or other external influences.

As shown in Veselý and Ilka (2013), gain scheduling is a widely used option to overcome those challenges. Hereby, a set of PID parameters is designed for each working point of a linear parameter-varying (LPV) system. This approach comes with a greater complexity during the system modeling and controller design process. Most notably, many LPV models contain some uncertainty that need to be recognized during the controller synthesis. An H_∞-based method for this is introduced in Veselý and Ilka (2015) by using linear matrix inequalities (LMIs). The practical applicability of an LMI-based approach is verified in Weiser et al. (2020), where a robust gain-scheduled controller is designed for the purpose of flight control. Furthermore, Yavari et al. (2022) propose a method based on an H_2 approach. A different direction is shown in Rosinová et al. (2022), where a pole placement approach is used to design robust gain-scheduled PID controllers for LPV systems.

However, while all of the mentioned design methods show good results, a common shortcoming is the necessity of having a proper model of the LPV system. Especially when working with real-world systems, e.g. in the field of hydraulics, this is a tough requirement. Very often, analytical descriptions of processes such as friction or magnetic hysteresis are hardly available. Furthermore, hydraulic systems among others often suffer from sparse sensor information in order to reduce costs.

Thus, this work aims to combine an H_∞-based control design method in combination with a data-efficient identification method for LPV systems. In this context, the capabilities of neural networks (NNs) are utilized because they can identify any nonlinear function according to the universal approximation theorem, introduced by Cybenko (1989). Overall, the main efforts of this work are:

- The introduction of simplified additive nonlinear autoregressive exogenous (SANARX) models.
- A formalism to reformulate NN-based SANARX models into an LPV structure.
- The synthesis of robust gain-scheduled PI controllers for those LPV systems by using an H_∞ method.

In order to demonstrate the capabilities of the proposed method, it is evaluated on a hydraulic test rig. Throughout this process, no more than the available input-output data is used to perform both the system identification and the controller synthesis task.

2. SYSTEM IDENTIFICATION

This section describes the used system identification procedure. In the first step, the NN-based SANARX is introduced. In the second step, it is brought into an LPV structure, which is the basis for the controller design.

2.1 SANARX Model

The baseline of the SANARX model is the class of NARX models. These are widely used to identify nonlinear systems and processes by just using previously sampled input-output data. In mathematical terms, this is written as:

$$\hat{y}_t = F\big(y_{t-1}, y_{t-2}, \ldots, y_{t-n}, u_{t-1}, u_{t-2}, \ldots, u_{t-n}\big). \tag{1}$$

Here, $\hat{y}_t$ denotes the predicted value, $y_{t-1}, \ldots, y_{t-n}$ denote the past output measurements, $u_{t-1}, \ldots, u_{t-n}$ denote the past input measurements, $F(.)$ is a nonlinear function approximator, and n is the number timesteps that are fed into the model. To capture all of the relevant dynamics, n should be chosen greater than or equal to the assumed order of the system to be identified.

Through inspection of (1), it becomes evident that the result of the identification process is a highly nonlinear function. This makes it harder to further analyze the system dynamics or even to design a robust controller. As proposed in Kreutmayr and Ament (2023a,b), the more structured SANARX model overcomes this problem:

$$\begin{aligned}\hat{y}_t &= f_1(y_{t-1}) + f_2(y_{t-2}) + \cdots + f_{d-1}\left(y_{t-(d-1)}\right) \\ &\quad + f_{lin,d}(y_{t-d}, u_{t-d}) + \cdots + f_{lin,n}(y_{t-n}, u_{t-n}) \\ &= \sum_{i=1}^{d-1} f_i(y_{t-i}) + \sum_{i=d}^{n} f_{lin,i}(y_{t-i}, u_{t-i}).\end{aligned} \tag{2}$$

Hereby, f_i is a nonlinear function approximator, while $f_{lin,i}$ is a linear function approximator.

Equation (2) shows that the timesteps in the SANARX model are separated into sublayers, which are eventually added up in the output. Thus, each function approximator is only fed with data from the same timestep. Additionally, sublayers 1 to $d-1$ are only fed with output data and sublayers d to n are fed with input and output data. With this, the influence of the system's relative degree is taken into consideration. Moreover, only feeding input data into linear function approximators ensures that the identified system is input affine. While this exact structure is already used in Kreutmayr and Ament (2023a,b), slightly different applications can be found e.g. in Petlenkov (2007); Belikov et al. (2013); Belikov and Petlenkov (2009).

Without any further calculation, the special characteristic of the SANARX model allows to reformulate (2) into a state space model:

$$\begin{aligned} x_{t+1}^{(1)} &= x_t^{(2)} + f_1\left(x_t^{(1)}\right) \\ &\ldots \\ x_{t+1}^{(d-1)} &= x_t^{(d)} + f_{d-1}\left(x_t^{(1)}\right) \\ x_{t+1}^{(d)} &= x_t^{(d+1)} + f_d\left(x_t^{(1)}, u_t\right) \\ &\ldots \\ x_{t+1}^{(n)} &= f_n\left(x_t^{(1)}, u_t\right). \end{aligned} \tag{3}$$

Here, $x_t^{(n)}$ refers to the current states and $x_{t+1}^{(1)} = \hat{y}_{t+1}$ refers to the predicted value for the next timestep.

In the course of this work, it is assumed that the function approximator will always take the form of an NN. However, it should be noted that the SANARX model can be combined with any function approximator that follows the introduced design rules. In the case of NNs, (3) is expressed as follows:

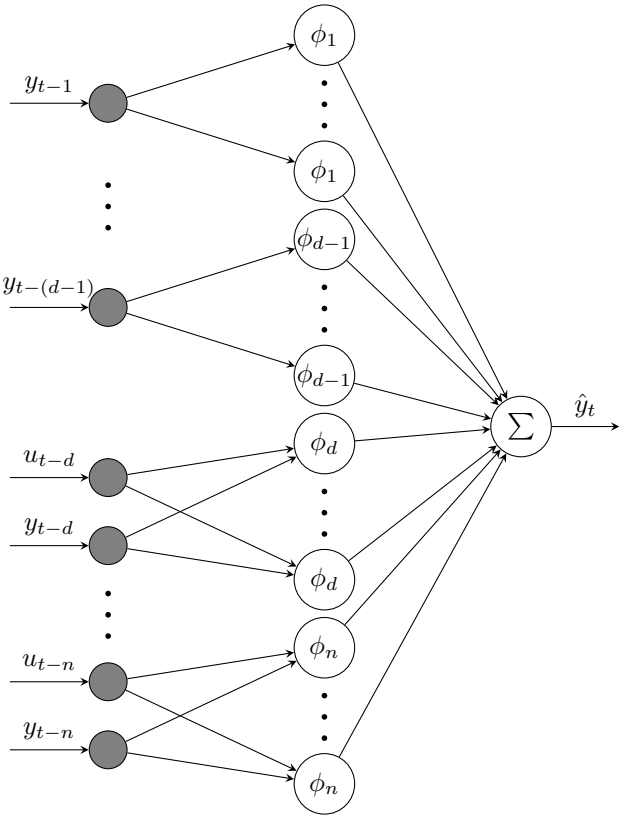

Fig. 1. Example of an NN in SANARX-structure.

$$\begin{aligned} x_{t+1}^{(1)} &= x_t^{(2)} + C_1\phi_1\left(W_1 x_t^{(1)}\right) \\ &\ldots \\ x_{t+1}^{(d-1)} &= x_t^{(d)} + C_{d-1}\phi_{d-1}\left(W_{d-1} x_t^{(1)}\right) \\ x_{t+1}^{(d)} &= x_t^{(d+1)} + C_d\phi_d\left(W_d[x_t^{(1)}, u_t]^\top\right) \\ &\ldots \\ x_{t+1}^{(n)} &= C_n\phi_n(W_n\left[x_t^{(1)}, u_t\right]^\top). \end{aligned} \tag{4}$$

Hereby, $\phi_n(.)$ denotes an activation function, W_n is the input weight matrix, and C_n is the output weight matrix. In order to ensure clarity and readability, no biases are used throughout this work. However, biases can be added without changing the functional principle of this method, as they are just linear additives. Figure 1 displays a visualization of the proposed SANARX structure.

2.2 LPV Model

In order to model systems with changing parameters, LPV models are used to maintain a linear structure. The variable parameters are dependent on the scheduling variable p. Through this, the often nonlinear behavior is embedded into p, while the system itself can be seen as linear in its current working point (see Shamma (2012)). This approach allows to use e.g. a gain-scheduling approach for a proper control of such nonlinear models.

Mathematically, the definition of an LPV model is:

$$\begin{aligned} x_{t+1} &= A\left(p_t\right) x_t + B\left(p_t\right) u_t \\ y_{t+1} &= C\left(p_t\right) x_t + D\left(p_t\right) u_t. \end{aligned} \tag{5}$$

Here, p_t is the scheduling variable, $A\left(p_t\right)$ is the system matrix, $B\left(p_t\right)$ is the input matrix, $C\left(p_t\right)$ is the output matrix and $D\left(p_t\right)$ is the feedthrough matrix.

As explained in Kreutmayr and Ament (2023a), the SANARX structure can be brought into the LPV form. For this, the elliot sigmoid activation function, which is introduced in Elliott (1993), is used:

$$\sigma\left(x\right) = \frac{1}{2}\left(1 + \frac{x}{1+|x|}\right). \tag{6}$$

Hereby, $\sigma\left(x\right)$ denotes the output of the function, which is normalized between $[0, 1]$. It is noteworthy that the usage of this sigmoidal-type activation adheres to the already mentioned universal approximation theorem. Additionally, the scheduling variable is defined as follows:

$$p_t = 2\left|x_t^{(1)}\right|. \tag{7}$$

By combining (4) with (6) and (7), one obtains:

$$\begin{aligned} x_{t+1} = &\begin{bmatrix} \frac{C_1 W_1}{2\left(1+|W_1|\frac{p_t}{2}\right)} & 1 & 0 & \dots & \dots & 0 \\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \frac{C_{d-1} W_{d-1}}{2\left(1+|W_{d-1}|\frac{p_t}{2}\right)} & \vdots & \ddots & \ddots & \ddots & \vdots \\ C_d W_d & \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 1 \\ C_n W_n & 0 & \dots & \dots & \dots & 0 \end{bmatrix} x_t \\ &+ \begin{bmatrix} 0 & \frac{C_1}{2}\mathbf{1}_{v,1} \\ \vdots & \vdots \\ 0 & \frac{C_{d-1}}{2}\mathbf{1}_{v,d-1} \\ C_d W_d & 0 \\ \vdots & \vdots \\ C_n W_n & 0 \end{bmatrix} \begin{bmatrix} u_t \\ 1 \end{bmatrix}. \end{aligned} \tag{8}$$

Hereby, v obtains the number of neurons in a certain sublayer and $\mathbf{1}$ is a vector of ones. Because p_t only depends on the endogenous state variable $x_t^{(1)}$, (8) describes a quasi-LPV system as stated in Kwiatkowski et al. (2006).

3. CONTROLLER DESIGN

A formalism to synthesize a gain-scheduled controller for a SANARX-based LPV system is introduced in the following section. For this purpose, the H_∞ method is combined with a standard PI structure, as it is shown in Kreutmayr and Ament (2023b).

3.1 Structured H_∞ Control

In order to control linear systems, H_∞ controllers are widely used since they guarantee broader stability margins then classical linear quadratic regulators (see Sagliano et al. (2021)). This property is important in cases where systems are not perfectly modeled and when disturbance inputs are not measurable. Even under those circumstances, the H_∞ method ensures a good and robust control performance. Moreover, H_∞ controllers do not require full state feedback, but solely need information from the signal to be controlled.

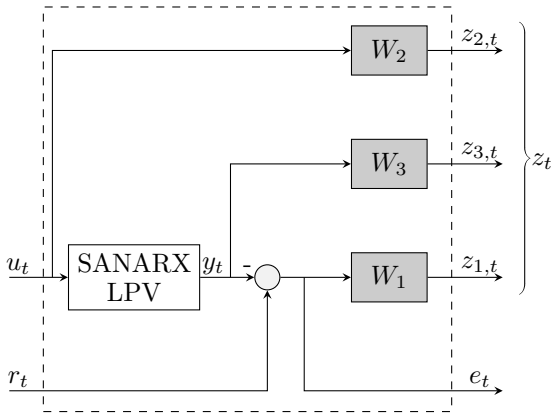

Fig. 2. Block diagram of the augmented plant P used in this work.

As already stated, H_∞ synthesis is a proper way of designing controllers for LPV systems. Furthermore, in Kreutmayr and Ament (2023b), combining H_∞ with a SANARX-based LPV system leads to well-performing controllers. In order to apply the method, the LPV system in (8) is reformulated into the form of an augmented plant P, such that:

$$\begin{bmatrix} z \\ e \end{bmatrix} = P \begin{bmatrix} r \\ u \end{bmatrix}, \tag{9}$$

with z as the error outputs, e as the errors that are provided to the controller, r as the reference signal, and u as the control inputs. A block diagram of the system used in this work is shown in Fig. 2. Here, P contains the weighting functions W_1, W_2, and W_3. They influence the system as follows:

- W_1 is the sensitivity function that affects the reference tracking and disturbance rejection.
- W_2 limits the control effort.
- W_3 is the sensitivity function that affects the robustness and noise attenuation.

When the H_∞ method is applied on the augmented plant P, the result is in the so-called PK structure. A block diagram of this is shown in Fig. 3. Here, controller K can be of any form that is capable of solving the H_∞ control problem for P. On the one hand, this leads to a wide solution space while on the other hand there is a high risk of getting controllers of high order. Those might be hard to implement in a practical application. Moreover, the orders may vary in different working points, when

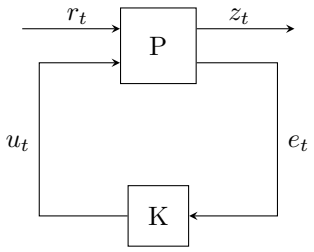

Fig. 3. Block diagram of the PK structure of the H_∞ method.

applying the H_∞ method to an LPV system. Therefore, a gain-scheduling approach could not be used.

Consequently, the H_∞ problem is solved for a fixed control structure in this work. By using the well-known PI controller, the result is expected to be robust and easy to implement.

3.2 PI Controller

Due to its slim structure, good interpretability, and easy-to-implement nature, the PI controller is widely used in academia as well as in industrial applications. In its discrete time form, a PI controller is denoted as:

$$u_t = K_P e_t + K_I \sum_{n=0}^{t} e_n T_s, \tag{10}$$

with K_P as the proportional gain, K_I as the integral gain, and T_s as the sampling time. Furthermore, the control error e_t is defined as:

$$e_t = y_{t,ref} - y_t, \tag{11}$$

with $y_{t,ref}$ as the reference signal. As mentioned above, a gain-scheduled PI controller is necessary to control an LPV system. As a result, the gains in (10) depend on the scheduling variable (7):

$$u_t = K_P\left(p_t\right) e_t + K_I\left(p_t\right) \sum_{n=0}^{t} e_n T_s. \tag{12}$$

As proposed in Gahinet and Apkarian (2011), MATLAB's *hinfstruct* is used to calculate the optimal controller parameters for each of i sampling points. The resulting set of parameters builds the foundation for the gain-scheduled controller, which linearly interpolates between the discrete values to ensure smooth transitions.

4. EXPERIMENTAL EVALUATION

In this section, the experimental evaluation of the proposed method is presented. In a first step, the hydraulic system is introduced briefly. Subsequently, the setup of the augmented plant and the experimental setting is explained. Conclusively, the results of the experiments are shown.

4.1 Hydraulic System

In order to evaluate the method, a hydraulic test system is used. It consists of a hydraulic pump, an oil tank, a differential cylinder, four electromagnetically driven 2/2-way proportional valves, and a combination of hoses and pipes that connect all elements. The interconnection of the components is shown in Fig. 4.

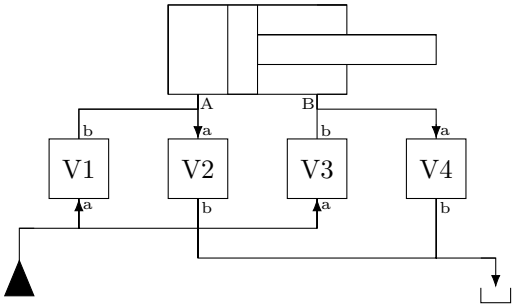

Fig. 4. Diagram of the hydraulic test system.

The valves can be controlled individually by applying an electrical current onto their electromagnet. The goal is to control the pressures in each chamber of the cylinder. Hereby, V1 and V2 are used to directly influence the pressure in the bottom side chamber A, while V3 and V4 modify the pressure in the rod side chamber B.

The cylinder can freely be driven backward and forwards. Moreover, it is possible to force it into a desired position by mechanically fixing the rod.

Due to effects such as frictions in the cylinder and the hydraulic valves, electromagnetic hysteresis, and the capacities of pipes and hoses, the system can be seen as highly nonlinear. Furthermore, it is assumed that only the pressures p_A and p_B in the cylinder chambers and all the electrical currents applied to the valves are measurable. This is of practical relevance, since additional sensors like volume meters are costly and therefore often not used in applications.

The goal is to control the cylinder's chamber pressures p_A and p_B using valves V2 and V4 respectively. Valves V1 and V3 provide varying volume flows to the cylinder, which can be interpreted as a disturbance for the controller.

To complete the control task, both cylinder chambers are interpreted as independent systems. Consequently, this results in two different LPV systems that need to be identified. Thus, only currents I_{V1} and I_{V2} along with p_A are used to identify the bottom side system. Similarly, currents I_{V3} and I_{V4} along with p_B serve as the relevant output data of the rod side system.

4.2 Setup of the Augmented Plant

In order to setup the SANARX model, it is assumed that both subsystems are of order $n = 3$ with a relative degree of $d = 3$. This results in a system where data from the input u appears only in the last layer. Hence, (4) becomes:

$$\begin{aligned} x_{t+1}^{(1)} &= x_t^{(2)} + C_1 \phi_1\left(W_1 x_t^{(1)}\right) \\ x_{t+1}^{(2)} &= x_t^{(3)} + C_2 \phi_2\left(W_2 x_t^{(1)}\right) \\ x_{t+1}^{(3)} &= C_3 W_3\left[x_t^{(1)}, u_1, u_2\right]^\top. \end{aligned} \tag{13}$$

Each of the sublayers contains of five neurons. As a result, (8) becomes:

$$x_{t+1} = \begin{bmatrix} \frac{C_1 W_1}{2\left(1+|W_1|\frac{p_t}{2}\right)} & 1 & 0 \\ \frac{C_2 W_2}{2\left(1+|W_2|\frac{p_t}{2}\right)} & 0 & 1 \\ C_3 W_3 & 0 & 0 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 & \frac{C_1}{2}\mathbf{1}_5 \\ 0 & 0 & \frac{C_2}{2}\mathbf{1}_5 \\ C_{3,1}W_{3,1} & C_{3,2}W_{3,2} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ 1 \end{bmatrix} \tag{14}$$

For the pressure control of both chambers, this LPV system serves as the basis for the augmented plant P as shown in (9). The performance functions for controlling p_A in the bottom side chamber are:

$$\begin{aligned} W_{1,A} &= \frac{0.015z - 0.0050}{z - 0.9997}, \\ W_{2,A} &= \frac{9.995z - 9.995}{z - 0.999}, \\ W_{3,A} &= 0, \end{aligned} \tag{15}$$

with a sampling time T_s of 0.005 s. Furthermore, the performance functions for controlling p_B in the rod side chamber are:

$$\begin{aligned} W_{1,B} &= \frac{0.0085z - 0.0065}{z - 0.9997}, \\ W_{2,B} &= \frac{9.995z - 9.995}{z - 0.999}, \\ W_{3,B} &= 0, \end{aligned} \tag{16}$$

Performance function W_3 is not used in both cases since it is closely related to W_1. As in many practical applications, it is assumed in this work that W_1 is sufficient to also cover robustness and noise attenuation. The exact parameters of the performance functions are the result of a systematic tuning process.

4.3 Experimental Setting

To demonstrate the capabilities of the proposed method, it is evaluated on the aforementioned system. For the training of both NN-SANARX models, two training sets, each consisting of 50,000 data points, are sampled. Hereby, the sampling rate is set to 0.005 s. The pressure of the pump is set to be 300 bar, while the tank pressure is 0 bar. The cylinder is continuously moved back and forth, while varying currents in the interval $[0, 1330]$ mA are applied onto the valves. These are sampled by using an amplitude-modulated pseudo-random binary signal (APRBS). For the first training set, the focus lies on generating data for controlling p_A in the bottom side chamber, while the second training set contains the data for controlling p_B in the rod side chamber.

The Levenberg-Marquardt (LM) algorithm is used for the training of the NN-SANARX models, which takes 2,500 iterations each. In course of this, a min-max normalization is applied onto the data to bring all features into the same range of scale, i.e. $[0, 1]$. As performance metric serves the mean squared error (mse):

$$mse = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{p}_{j,i} - p_{j,i}\right)^2, \tag{17}$$

with N as the number of data points and $j = \{A, B\}$. Since the training is conducted in an open loop setting, the predicted values $\hat{p}_j$ are not fed back into the network.

After the training process is completed, both LPV systems are sampled in ten different working points, which are equidistantly distributed in the interval $[0, 300]$ bar. Afterwards, PI parameters are synthesized for each working point by using MATLAB's *hinfstruct* in combination with the performance functions (15) and (16).

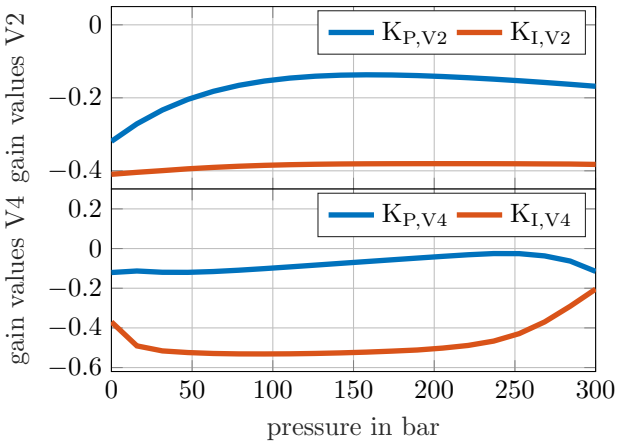

Fig. 5. Gains calculated using the H_∞ procedure.

4.4 Evaluation

Both learning processes end with an $mse < 1 \times 10^{-6}$, where the calculation is done over the whole dataset by using normalized values. The low mse indicates that the models are sufficient to apply the H_∞ synthesis. Figure 5 shows the resulting PI gains for both models. As mentioned earlier, the controllers are applied to valve V2 for controlling the bottom side chamber and valve V4 for controlling the rod side chamber. In both cases, the resulting gains are in the same order of magnitude. The difference in values might be explained by the serial spread of the valves and the different capacities in the cylinder chambers.

To evaluate the performance of the controllers, a test cycle is defined. Again, the pump pressure is 300 bar, while the tank pressure is 0 bar. In this case, the cylinder does not move, but is mechanically fixed in its center position. Both pressures p_A and p_B are supposed to follow a given trajectory through the control of V2 and V4. Moreover, V1 and V3 change the flow into the chambers, which acts like a disturbance. In addition, a changing pressure in one chamber automatically affects the pressure in the other chamber. This is due to the mechanical coupling of the two chambers. The resulting behavior can be seen in Fig. 6. In the bottom graph, the behavior of all of the

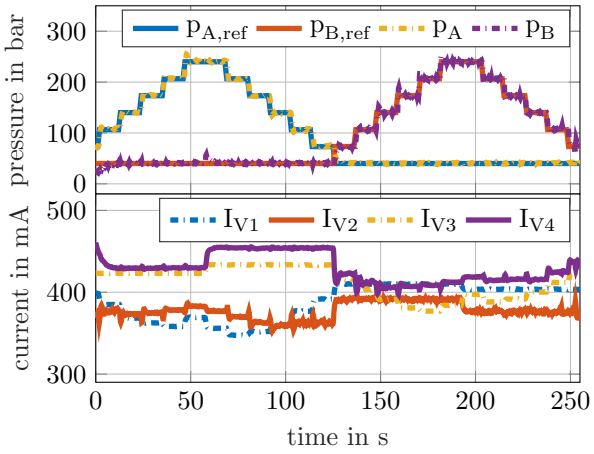

Fig. 6. Result of applying the PI controllers to the real system.

currents in visualized. The upper graph shows that both controllers are able to maintain a constant pressure as well as to track a given profile. Note that p_B is slightly more jittery than p_A. This is due to the lower volume in the cylinder's rod side, which makes this subsystem more aggressive since changes in volume flow result in a higher pressure gradient than on the bottom side. The plot also shows that the pressure changes in the rod side cause more disturbance in the bottom side than vice versa. Eventually, the root mean squared error ($rmse$) is calculated for both controllers. This is done as follows:

$$rmse = \sqrt[2]{mse}, \tag{18}$$

with mse as the result of (17). For the bottom side, the $rmse$ is 22.93, while the $rmse$ of the rod side is 29.13. The slightly higher error value on the rod side can be explained by the jittery behavior of p_B. Overall, this can be seen as a satisfactory result for this control task.

5. CONCLUSION

In this paper, an end-to-end method for automatically tuning gain-scheduled PI controllers is introduced and applied to a real-world system. Here, only input-output data is used to identify an NN-based SANARX model. After reformulation into LPV form, H_∞ synthesis is performed to obtain a robust set of PI parameters for different working points. To illustrate the practicality and robustness of the method, it is evaluated on a real-world hydraulic test rig.

Despite of the advantages of the method, it should be noted that the control result depends strongly on the quality of the identified system and thus the representability of the training data. Hence, an insufficient dataset may lead to an unstable controller even though the H_∞ synthesis is successful.

In future work, this drawback could be addressed by a mechanism that evaluates the quality of the sampled training dataset. In addition, by incorporating the method into a reinforcement learning framework, a retraining process could be introduced. Finally, the use of higher order control structures could help to further improve the control performance.

REFERENCES

Belikov, J., Nõmm, S., Petlenkov, E., and Vassiljeva, K. (2013). Application of neural networks based sanarx model for identification and control liquid level tank system. In *2013 12th International Conference on Machine Learning and Applications*, volume 1, 246–251.

Belikov, J. and Petlenkov, E. (2009). Nn-anarx model based control of nonlinear discrete-time systems with input delay. In *2009 IEEE Control Applications, (CCA), Intelligent Control, (ISIC)*, 1039–1044.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4), 303–314.

Elliott, D. (1993). A better activation function for artificial neural networks. Technical report.

Gahinet, P. and Apkarian, P. (2011). Structured h-infinity synthesis in matlab. *IFAC Proceedings Volumes*, 44, 1435–1440.

Kreutmayr, F. and Ament, C. (2023a). Development of lpv models for nonlinear systems based on simplified additive nonlinear autoregressive exogenous models. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1132–1137.

Kreutmayr, F. and Ament, C. (2023b). Robust gain-scheduled controllers for nonlinear systems based on simplified additive nonlinear autoregressive exogenous models. In *2023 27th International Conference on System Theory, Control and Computing (ICSTCC)*, 215–220.

Kwiatkowski, A., Boll, M.T., and Werner, H. (2006). Automated generation and assessment of affine lpv models. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 6690–6695.

Petlenkov, E. (2007). *Neural Networks Based Identification and Control of Nonlinear Systems : ANARX Model Based Approach.* Ph.D. thesis, Tallinn University of Technology.

Rosinová, D., Veselỳ, V., and Hypiusová, M. (2022). Novel robust gain scheduled pid controller design using dr regions. *Asian Journal of Control*, 24(5), 2062–2073.

Sagliano, M., Tsukamoto, T., Heidecker, A., Hernandez, J.A.M., Fari, S., Schlotterer, M., Woicke, S., Seelbinder, D., Ishimoto, S., and Dumont, E. (2021). Robust control for reusable rockets via structured h-infinity synthesis. In *11th International ESA Conference on Guidance, Navigation & Control Systems.*

Shamma, J.S. (2012). *An overview of LPV systems*, volume 9781461418337, 3–26. Springer, Germany.

Veselý, V. and Ilka, A. (2013). Gain-scheduled pid controller design. *Journal of Process Control*, 23(8), 1141–1148.

Veselý, V. and Ilka, A. (2015). Design of robust gain-scheduled pi controllers. *Journal of the Franklin Institute*, 352(4), 1476–1494.

Weiser, C., Ossmann, D., and Looye, G. (2020). Design and flight test of a linear parameter varying flight controller. *CEAS Aeronautical Journal*, 11(4), 955–969.

Yavari, R., Shamaghdari, S., and Sadeghzadeh, A. (2022). Robust h2 output-feedback bumpless transfer control of polytopic uncertain lpv systems. *European Journal of Control*, 63, 277–289.