Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

On considering the output in space-filling test signal designs for the identification of dynamic Takagi-Sugeno models

Matthias Gringard and Andreas Kroll

Department of Measurement and Control, University of Kassel,
D-34109 Kassel, Germany
{*matthias.gringard,andreas.kroll*}*@mrt.uni-kassel.de*

Abstract: The model-based design of test signals for the identification of dynamical Takagi-Sugeno (TS) fuzzy models is addressed. The multi-model structure is exploited to reduce computational cost. Space-filling designs usually only address the input but the nonlinear behavior of dynamic systems depends on the lagged output in general. This is considered as an additional constraint regarding the test signal design. This contribution investigates whether a control input can be calculated exploiting the local structure and whether considering the output in space-filling designs yields identified models of higher quality.

Keywords: Optimal experiment design, feedforward control, nonlinear system identification, fuzzy modeling, nonlinear systems

1. INTRODUCTION

It is of great interest to design test signals that excite systems in their full operating range. This excitation is important for the identification of high quality models that can be used for simulation and control design. In this contribution, the focus is on the model-based test signal design for the identification of locally affine, dynamic Takagi-Sugeno (TS) fuzzy models. The multi-model structure of TS models can be exploited for control design based on linear approaches which is not possible for other universal approximators like neural networks.

In this approach, an initial model is used to design a test signal such that the output behaves in a certain way. Space-filling designs have been extended from their application on static systems to dynamic systems like in Ebert et al. (2015). For the identification of multi-models, it is important to cover the scheduling space well, since the nonlinear behavior is only described by the superposition that depends on the scheduling variable. In contrast to the identification of static systems, the scheduling variable is likely to contain the output. If a space-filling test signal design is applied just to the input space, the task of achieving a sufficient coverage of the scheduling space is only partly fulfilled or at least not addressed in the design scheme.

Since the input is used to purposefully manipulate the output, the test signal design can be viewed as a control problem. An initial model is required for this method. It is obtained using model-free approaches like in Gringard and Kroll (2016) and Deflorian (2011). The identification then becomes an iterative process in which the success depends on the initial identification. Kroll and Dürrbaum (2015) have shown that data within the center of a partition yield insufficient partition parameters. Therefore a space-filling design that takes into account the output within the scheduling variable is assumed to yield models with higher generalization capability.

For this method, a reference output has to be designed that satisfies conditions on the scheduling variable assumed to be important. The local structure of the TS models will be exploited during the input design. In this contribution, it is investigated whether superposed local designs can achieve the task of controlling the actual system's output. It is also investigated whether the space-filling assumptions result in higher quality models.

In section 2, TS models and the used nomenclature are introduced as well as the identification process. In section 2, the proposed method for the test signal design is discussed. Before this contribution is closed with a summary and an outlook in section 5, a case study is presented in 4.

2. MODEL CLASS AND IDENTIFICATION PROCESS

In this section, the class of the used TS models and the identification process is presented.

2.1 Locally affine Takagi-Sugeno models

Discrete-time, locally affine Takagi-Sugeno (TS) SISO models are considered in this paper. These result from superposing c local models that are weighted by the respective fuzzy basis function (FBF). The prediction equation of such a model is:

$$\hat{y}(k) = \sum_{i=1}^{c} \phi_i\left(\mathbf{z}(k), \boldsymbol{\Theta}_{\mathrm{MF}}\right) \cdot \hat{y}_i\left(\boldsymbol{\varphi}(k), \boldsymbol{\Theta}_{\mathrm{LM},i}\right) \quad (1)$$

The local models are given in ARX (AutoRegressive with eXogenous input) configuration:

⋆ This work was supported by the German Research Foundation (DFG), project code KR 3795/7-1

$$\begin{aligned}\hat{y}_i(k) :=& \hat{y}_i(\boldsymbol{\varphi}(k), \boldsymbol{\Theta}_{\mathrm{LM},i}) = \boldsymbol{\varphi}^\top(k) \cdot \boldsymbol{\Theta}_{\mathrm{LM},i} \\ =& \underbrace{[y(k-1)\cdots y(k-n)\ u(k-1)\cdots u(k-m)\ 1]}_{\boldsymbol{\varphi}^\top(k)} \\ & \cdot \underbrace{[-a_{i,i}\cdots -a_{i,n}\ b_{i,1}\cdots b_{i,m}\ c_i]^\top}_{\boldsymbol{\Theta}_{\mathrm{LM},i}} \end{aligned} \tag{2}$$

In (1) and (2), $\boldsymbol{\Theta}_{\mathrm{LM}_i}$ is the local parameter vector of the i-th local model, $\boldsymbol{\varphi}(k)$ is the regression vector, n and m are the dynamic orders of the output and input, respectively. The FBF depend on the scheduling variable $\mathbf{z}(k)$ as well as the partition parameters $\boldsymbol{\Theta}_{\mathrm{MF}}$. Without a priori knowledge it is assumed that the components of the scheduling variable are a subset of the components of the regression vector. The FBF are normalized membership functions (MF) $\mu_i(\mathbf{z}(k), \boldsymbol{\Theta}_{\mathrm{MF}})$:

$$\phi_i(\mathbf{z}(k), \boldsymbol{\Theta}_{\mathrm{MF}}) = \frac{\mu_i(\mathbf{z}(k), \boldsymbol{\Theta}_{\mathrm{MF}})}{\sum_{j=1}^{c} \mu_j(\mathbf{z}(k), \boldsymbol{\Theta}_{\mathrm{MF}})} \tag{3}$$

with

$$\boldsymbol{\Theta}_{\mathrm{MF}}^\top = [\mathbf{v}_1\ \cdots\ \mathbf{v}_c] \tag{4}$$

In this contribution, the prototype-based MF of the fuzzy-c-means (FCM) cluster algorithm are used:

$$\mu_i(\mathbf{z}(k), \boldsymbol{\Theta}_{\mathrm{MF}}) = \left[\sum_{j=1}^{c} \left(\frac{\|\mathbf{z}(k) - \mathbf{v}_i\|}{\|\mathbf{z}(k) - \mathbf{v}_j\|}\right)^{\frac{2}{\nu-1}}\right]^{-1} \tag{5}$$

Since $\sum_{j=1}^{c} \mu_j(\mathbf{z}(k), \boldsymbol{\Theta}_{\mathrm{MF}}) \equiv 1$, the FBF and MF are identical. The number of local models c as well as the fuzziness parameter $\nu \in \mathbb{R}^{>1}$ are hyperparameters. Different distance norms $\|\cdot\|$ are possible. In this contribution, the Euclidean distance is used. $\mathbf{v}_i$ are the prototypes and make up the vector of partition parameters $\boldsymbol{\Theta}_{\mathrm{MF}}$. If (2) is put into (1), $\hat{y}(k)$ is linear in $\boldsymbol{\Theta}_{\mathrm{LM}}$:

$$\begin{aligned}\hat{y}(k) &= \underbrace{[\mu_1 \boldsymbol{\varphi}^\top(k) \cdots \mu_c \boldsymbol{\varphi}^\top(k)]}_{\boldsymbol{\varphi}_{\mathrm{E}}^\top(k)} \cdot \underbrace{[\boldsymbol{\Theta}_{\mathrm{LM},1} \cdots \boldsymbol{\Theta}_{\mathrm{LM},c}]^\top}_{\boldsymbol{\Theta}_{\mathrm{LM}}} \\ &=: \boldsymbol{\varphi}_{\mathrm{E}}^\top(k)\, \boldsymbol{\Theta}_{\mathrm{LM}} \end{aligned} \tag{6}$$

2.2 Identification process

The identification is conducted in four steps: i) The determination of the structural hyperparameters (ν, c), ii) the calculation of the partition parameters through clustering, iii) the least-squares (LS) estimation of the local model parameters and iv) the parallel optimization of all parameters.

In the first step, the hyperparameters c and ν as well as the dynamic orders n and m are determined. c and ν are determined by cluster validity measures or the prediction error as in Juhász et al. (2006). The scheduling variable $\mathbf{z}(k)$ is often chosen to be identical to the regression vector $\boldsymbol{\varphi}(k)$.

With known hyperparameters, the scheduling space can be partitioned by clustering in the input space by e.g. the FCM algorithm providing the MF (5) in the second step.

In the third step, a global estimation of the local model parameters $\boldsymbol{\Theta}_{\mathrm{LM}}$ is conducted similar to Babuška (1998). For this, the representation of the TS model (6) is evaluated in N datapoints which is for the case of $n \geq m$:

$$\underbrace{\begin{bmatrix}\hat{y}(n)\\ \vdots \\ \hat{y}(N)\end{bmatrix}}_{\hat{\mathbf{Y}}} = \underbrace{\begin{bmatrix}\boldsymbol{\varphi}_{\mathrm{E}}^\top(n)\\ \vdots \\ \boldsymbol{\varphi}_{\mathrm{E}}^\top(N)\end{bmatrix}}_{\boldsymbol{\Phi}_{\mathrm{E}}} \boldsymbol{\Theta}_{\mathrm{LM}} \tag{7}$$

Within the prediction error method (PEM) framework, a quadratic cost function is minimized for the ARX model. It can be written as:

$$J_{\mathrm{PEM}}(\boldsymbol{\Theta}_{\mathrm{LM}}) = \left(\hat{\mathbf{Y}}(\boldsymbol{\Theta}_{\mathrm{LM}}) - \mathbf{Y}\right)^\top \left(\hat{\mathbf{Y}}(\boldsymbol{\Theta}_{\mathrm{LM}}) - \mathbf{Y}\right) \tag{8}$$

$\mathbf{Y} = [y(n)\ \cdots\ y(N)]^\top$ is the vector of output measurements. The argument minimizing $J_{\mathrm{PEM}}(\boldsymbol{\Theta}_{\mathrm{LM}})$ is:

$$\hat{\boldsymbol{\Theta}}_{\mathrm{LM}} = \arg\min_{\boldsymbol{\Theta}_{\mathrm{LM}}} J_{\mathrm{pem}}(\boldsymbol{\Theta}_{\mathrm{LM}}) = \left(\boldsymbol{\Phi}_{\mathrm{E}}^\top \boldsymbol{\Phi}_{\mathrm{E}}\right)^{-1} \boldsymbol{\Phi}_{\mathrm{E}}^\top \mathbf{Y} \tag{9}$$

Since the FCM algorithm converges locally, it is executed multiple times with random initialization of the prototypes followed by the LS estimation of the local model parameters. The best model on the validation data set is selected and used for the succeeding optimization step.

The partition parameters are optimal with respect to the data grouping, not with respect to the prediction error. Since the nonlinear ARX (NARX) estimation often results in an unsatisfying simulation behavior, all model parameters

$$\boldsymbol{\Theta} = \left[\boldsymbol{\Theta}_{\mathrm{LM}}^\top\ \boldsymbol{\Theta}_{\mathrm{MF}}^\top\right]^\top \tag{10}$$

are optimized with respect to the sum of squared simulation errors. In a simulation, previous model outputs are used for predicting the following outputs (recursive model evaluation, nonlinear output error (NOE) configuration):

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \sum_{k=n}^{N} \left(y(k) - \hat{y}(k, \boldsymbol{\Theta})\right)^2 \tag{11}$$

The nonlinear optimization is conducted using the `MATLAB` function `lsqnonlin` which uses a trust region reflective algorithm by default as it can be found in MathWorks (2019).

In this contribution, the simulation error is defined as:

$$e(k) = \hat{y}(k) - y(k) \tag{12}$$

An averaging measure (Normalized Root Mean Square Error, J_{NRMSE}) is used:

$$J_{\mathrm{NRMSE}} = \frac{\sqrt{\frac{1}{N-n}\sum_{k=n}^{N} e(k)^2}}{y_{\max} - y_{\min}} \tag{13}$$

The NRMSE (13) is normalized to the signal range for comparable values in case of different output signal ranges.

3. TEST SIGNAL DESIGN METHOD

The presented method is a model-based approach to achieve desired properties of the scheduling variable $\mathbf{z}(k)$ during the experiment by designing the system's reference signal $y_{\mathrm{ref}}(k)$. Since the output is subject to the system's dynamics a control scheme has to be developed. The local structure of the TS models will be exploited to obtain local control variables that are superposed to build the required

test signal. The test signal is then used to identify the system. The procedure can be divided into three steps: i) Design of the reference signal $y_{\mathrm{ref}}(k)$, ii) calculation of the test signal and iii) subsequent identification.

3.1 Design of the reference signal

In general, the nonlinear behavior of dynamic systems depends not only on the input but also on the lagged output. Therefore, the space-filling input designs have to be adapted and applied to the system's output. The assumption of space-filling designs is that if there is no a priori knowledge about the system, it is a plausible approach to uniformly excite the system. But a uniformly distributed input signal might not excite the system in a way that the scheduling space is covered uniformly which might lead to insufficient identification of the partition parameters. In this contribution, it is assumed that the system's input doesn't determine the nonlinear behavior. Therefore it can be described by the output and its lags alone. So the scheduling variable in general consists of a subset of the lagged outputs. The design itself has to be tailored to the problem.

3.2 Calculation of the test signal

The calculation of the test signal itself can be divided into two steps. First the local models are used to obtain local control variables. In this contribution, a direct calculation of the local control variable is used, since for this specific method the reference signal is known. Secondly, the local control variables are superposed. This method can be used for different approaches of calculating the local control variables where the minimization of a cost function containing nonlinear system dynamics results in high computational cost.

Supposed a fuzzy model (1) is given and the i-th local OE model (2)

$$\hat{y}_i\left(k\right) = c_i - \sum_{l=1}^{n} a_{i,l}\hat{y}\left(k-l\right) + \sum_{l=1}^{m} b_{i,l}u\left(k-l\right) \tag{14}$$

is given in its summation form with the abbreviation $\phi_i\left(\mathbf{z}\left(k\right)\right) =: \phi_i\left(k\right)$. The global model becomes:

$$\hat{y}\left(k\right) = \sum_{i=1}^{c} \phi_i\left(k\right) \cdot \hat{y}_i\left(k\right) \tag{15}$$

In this contribution, the method of determining the local control variables is straightforward, but as mentioned before, it can only be calculated if a reference signal is known at this stage. In (14), the model output $\hat{y}_i(k)$ is substituted by the known reference $y_{\mathrm{ref}}(k)$, resulting in a linear difference equation for the calculation of the desired local control variable $u_j(k)$. This corresponds with the flat output design used by Zeitz (2010) for a more general state space representation. Since a difference equation has to be solved, initial values for the respective control variables $u_j(k)$ have to be specified in accordance with the reference signal. The big advantage using the Input-/Output-(I/O)-representation with TS models is twofold. First the dynamics of the control variable (i.e. the b coefficients and respective lags in $u(k)$) cannot be shifted to the output as it is done for linear systems. It is possible to have the local models in the state space representation but the local states would not be identical for the local models. The affine term c_i is never problematic when using identification in the I/O-representation. The calculation of the j-th local control variable therefore is with $a_{j,0} = 1$:

$$\sum_{l=1}^{m} b_{j,l}u_j\left(k-l\right) = \sum_{l=0}^{n} a_{j,l} \cdot y_{\mathrm{ref}}\left(k-l\right) - c_j \tag{16}$$

The c local control variables $u_j(k)$ are the solutions to the c difference equations (16). A scheme is needed to superpose the local control variables $u_j(k)$. The approach is based on works of the parallel distributed control (PDC) community in which the local state feedbacks are superposed by the forward model's basis functions e.g. in Li et al. (1999).
With given local control variables $u_j\left(k\right)$ from (16) as well as the superposition scheme

$$u(k) = \sum_{j=1}^{c} \phi_j u_j(k) \tag{17}$$

The issue with (17) is it cannot be calculated directly in the case of the scheduling variable containing the input u_k, since the basis functions to be used at every time instance to weight the local control variables would change with the desired result of the calculation itself. In that case (17) has to be solved using nonlinear equation solving methods.
It can be put into (14) which then is put into (15). Since the coeffictients $b_{i,l}$ don't depend on the counting index j the j-sum can be put in front: Also, since the FBF are normalized it is possible to put a sum over all the basis functions in front of every term and pull it out of the bracket as a whole:

$$\hat{y}(k) = \sum_{i=1}^{c}\sum_{j=1}^{c} \phi_i\left(k\right)\phi_j\left(k\right) \cdot \hat{y}_{i,j}\left(k\right) \tag{18}$$

with

$$\hat{y}_{i,j}\left(k\right) = c_i - \sum_{l=1}^{n} a_{i,l}\hat{y}\left(k-l\right) + \sum_{l=1}^{m} b_{i,l}u_j\left(k-l\right) \tag{19}$$

The interpretation of this equation is that the global model is a superposition of every local model excited by every local control variable weighted by the FBF product $\phi_i\phi_j$. Since $\phi_i\phi_j$ only approaches Kronecker's delta in regions around a local model's prototype, the cross terms $\hat{y}_{i,j}(k)$ in the model output have to be taken into account. This will be addressed in subsection 3.3.

To obtain the input $u(k)$ the FBF have to be calculated. This cannot be achieved by using the reference, because deviations from the reference are expected due to the impact of the cross terms. These deviations would accumulate during an experiment. Therefore, the scheduling variable is generated during a simulation of the global model as shown in fig. 1. The grey boxes in fig. 1 contain the FBF and the feedback of the output $\hat{y}(k)$ implies that the scheduling variable is calculated from the output and plugged into the FBF. The local control variables then are superposed to the $u(k)$ that is applied to the fuzzy model. After the completion of the simulation, the resulting $u(k)$ can be used as a test signal to be applied to the system to be identified. The calculation of the test signal follows the following procedure:

(1) Generate reference signal $y_{\mathrm{ref}}\left(k\right)$ (method depends on individual modeling task)

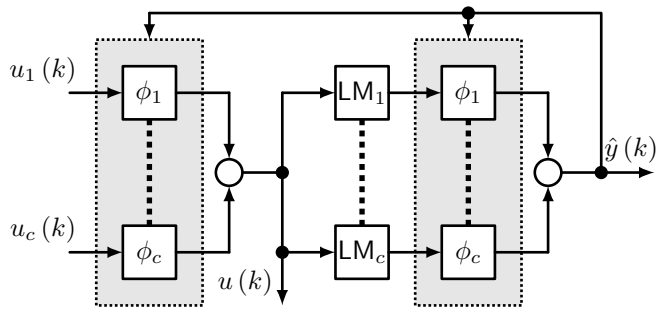

Fig. 1. Superposition scheme for obtaining $u(k)$

(2) Solve the c difference equations in (16)
(3) Calculate scheduling variable current model output
(4) Evaluate FBF at current scheduling variable
(5) Calculate (17)
(6) Calculate the next model output with (15)
(7) Repeat (3) to (6) until the whole input signal is constructed.

3.3 Deviation in regions of model transition

The basic assumption in nonlinear model-based test signal design is that it is possible to identify an initial model that captures the system's beahvior sufficiently well to even attempt model-based design. In a region around a local model's prototype, the activation of cross terms can be neglected. In this section, it is demonstrated that the deviation $e(k) = \hat{y}(k) - y(k)$ is sufficiently small under certain assumptions, even in regions of model transition, where $y(k)$ denotes the reference in this example. These assumptions will be introduced later. The calculations are presented on the example of a first order model to make the result clear, but under the assumptions made the argument holds for higher orders:

$$\hat{y}(k) = \sum_{i=1}^{c} \phi_i(k) \left(a_i \hat{y}(k-1) + b_i u(k)\right) \tag{20}$$

The j-th local input is:

$$u_j(k) = \frac{y(k) - a_j y(k-1)}{b_j} \tag{21}$$

With (21) the global model (19) can be rewritten as:

$$\begin{aligned}\hat{y}(k) = &\sum_{i=1}^{c} \phi_i^2(k)\, y_{i,i}(k) \\ &+ \left(1 - \sum_{i=1}^{c} \phi_i^2(k)\right) \frac{\hat{y}_{i,j}(k) + \hat{y}_{j,i}(k)}{2}\end{aligned} \tag{22}$$

The first assumption is that the past model outputs are approximately equal to the past values of the reference ($\hat{y}(k-1) = y(k-1)$), which is plausible when entering a region of transition. This allows the analysis of the deviation without the accumulation over time which diminishes after leaving transition areas. The cross terms then can be written as:

$$\hat{y}_{i,j}(k) = \begin{cases} y(k), & i = j \\ a_i y(k-1) + \frac{b_i}{b_j}\left(y(k) - a_j y(k-1)\right), & i \neq j \end{cases} \tag{23}$$

The sum of the corresponding cross terms for the case $i \neq j$ is:

$$\begin{aligned}\hat{y}_{i,j}(k) + \hat{y}_{j,i}(k) = &\frac{b_i^2 + b_j^2}{b_i b_j} y(k) \\ &+ (b_i - b_j)\left(\frac{a_i}{b_i} - \frac{a_j}{b_j}\right) y(k-1)\end{aligned} \tag{24}$$

The second assumption is that only the local models where the transition takes place are contributing significantly to the model output. The reactivation effect of the used FBF only is an issue in regions far away from the prototypes. The only transition that is addressed is between two models. Therefore the local model parameters of one model can be expressed in terms of the other in order to analyze how the difference in local model parameters impacts the model output: With $a_i = a$, $a_j = \alpha a$, $b_i = b$ and $b_j = \beta b$ the model error can be written as:

$$\begin{aligned}e(k) = &\left(1 - \sum_{i=1}^{c} \phi_i^2\right) \\ \cdot &\left(\frac{(1-\beta)^2}{2\beta} y(k) + \frac{a}{2}(1-\beta)\left(1 - \frac{\alpha}{\beta}\right) y(k-1)\right)\end{aligned} \tag{25}$$

The first term is the activation of all cross terms with $i \neq j$. It is intuitive that the error approaches 0 in a region around a local model's prototype. (25) shows that the error also depends on the difference of the local models. The technical systems in question are considered to be Lipschitz continuous. Therefore, neighboring models are assumed to be sufficiently similar. However, it is interesting that the proper motion of the local models enters the error term absoluteley. It is concluded that the feedforward control of a TS model with a superposition of local control variables is a reasonable approach since the errors don't accumulate and remain small under plausible assumptions. Furthermore it is expected that the deviations based on the fact that the system is not in the model class are larger than the deviations based on the cross terms which will be addressed in the case study.

4. CASE STUDY

The goal of this case study is to demonstrate that the nonlinear superposition of control inputs derived from identified local linear models can be used as a global control variable for the nonlinear system. Furthermore, the ability to generate system outputs with desired properties is used to investigate whether space-filling methods considering the output yield higher quality models.

4.1 Test system and modeling decisions

The system from Narendra and Parthasarathy (1990) is an established benchmark system. Its output $y_{\mathrm{sys}}(k)$ is given by the 2nd order difference equation:

$$y_{\mathrm{sys}}(k) = f\left(y_{\mathrm{sys}}(k-1), y_{\mathrm{sys}}(k-2)\right) + u(k-1) + \eta(k) \tag{26}$$

with

$$\begin{aligned}&f\left(y(k-1), y(k-2)\right) = \\ &\frac{y(k-1) \cdot y(k-2) \cdot \left(y(k-1) + 2{,}5\right)}{1 + y(k-1)^2 + y(k-2)^2}\end{aligned} \tag{27}$$

$\eta(k)$ is chosen to be an i.i.d. normally distributed random variable $\eta(k) \in \mathcal{N}\left(0, 0.1^2\right)$. For the determination of the

hyperparameters c and ν, three datasets are used. Fig. 2 shows the input data in the top row and the corresponding output data in the bottom row. To determine the hyperparameters multiple complete identifications (including the nonlinear optimization) have been conducted on the first dataset (D1) shown in the leftmost column for all combinations of $\nu \in \{1.1, 1.3, 1.7\}$ and $c \in \{3, 4, 5, 6\}$. The resulting models from dataset D1 have been evaluated on the second (D2) and the third dataset (D3), see fig. 2. A time variable $t(k) = k \cdot T_\mathrm{S}$ with $T_\mathrm{S} = 0.01\,\mathrm{s}$ has been added to simplify comparison. The model performance is assessed with the NRMSE (13). The results can be seen in table 1. The first three rows show the NRMSE evaluation

ν/c	3	4	5	6	dataset
1.1	0.9766	0.2977	0.2792	0.5060	
1.3	0.5387	0.5079	0.2798	0.6239	D2
1.7	0.3495	0.2746	0.2775	0.3206	
1.1	1.0593	0.7332	0.1257	1.1794	
1.3	0.7734	0.8370	0.3102	1.4083	D3
1.7	0.5055	0.0622	0.3729	1.0813	

Table 1. NRMSE for model evaluations on datasets D2 and D3

of the initial models on the dataset D2 and the last three rows the evaluation on the dataset D3, respectively. The number of local models is chosen as $c = 5$ and the fuzziness parameter as $\nu = 1.1$. Since the true system is known, the dynamic orders are set to $n = 2$ and $m = 1$. Considering the affine term, the regression vector is:

$$\boldsymbol{\varphi}^\top = [1 \;\; y\,(k-1) \;\; y\,(k-2) \;\; u\,(k-1)] \tag{28}$$

Since it is also known from the true system that just the regressors $y(k-1)$ and $y(k-2)$ determine the nonlinear behavior, the scheduling variable is chosen as:

$$\mathbf{z}\,(k) = [y\,(k-1) \;\; y\,(k-2)] \tag{29}$$

4.2 Design of the reference signal

Since the scheduling space is spanned by two consecutive output variables, it is expected that the data is placed around the main diagonal in the first and third quadrants. It is proposed to assess the coverage of the scheduling space by assessing the scattering of the data points around the main diagonal. To achieve this, the data in the scheduling space is rotated by 45 ° and the empirical standard deviation is calculated. Fig. 3 shows the rotated scheduling space as well as the scheduling variables of D1 (olive cross), the desired reference (cyan circle) and the actual system's output during the refined experiment (magenta triangle). The standard deviation in the rotated scheduling space of the data from D1 is $s_\mathrm{initial} = 0.0223$. The standard deviation of the reference signal is $s_\mathrm{ref} = 0.0484$ and is twice the value from D1. A factor of 2 in the standard deviation means that the same amount of data is scattered in a region 4 times as big. At this point there is no general method of designing the needed reference signal. In this example, a product of a sine wave with a swept sine have been altered by changing the amplitude as well as the frequency and the frequency rate of change randomly until a reference signal met the criteria mentioned before.

4.3 Test signal design

For the test signal design, the best performing model is chosen, the local inputs are calculated from (16) and superposed using the procedure described in section 3.2. Fig. 4 shows the refined test signal in the top plot which doesn't look like a test signal that could result from tweaking standard test signals manually. The bottom plot of fig. 4 shows the responses of the fuzzy model (magenta), the true system (olive) to the refined test signal and the reference (black dashes). The NRMSE values are $J_\mathrm{fuzzy} = 0.0928$ and $J_\mathrm{truesys} = 0.528$ for the fuzzy model and true system, respectively. From fig. 4 two important facts can be derived. Firstly, the response of the fuzzy model has small deviations from the reference, whereas the deviations of the true system are larger. This corroborates the assumption that the deviations from the reference based on the system not being in the model class are more significant than the deviations regarding the impact of the cross terms in this example. Secondly, the deviations coincide. This leads to the assumption that the local behavior is already approximated well by the initial model but not the partition transitions. Since it was not the primary goal to exactly achieve the reference signal but to achieve well-covered scheduling space, the first part of the design task has been achieved. The scheduling variable during the refined experiment is shown in the rotated scheduling space in fig. 3 as magenta triangles. Its standard deviation $s_\mathrm{new} = 0.0564$ is increased with respect to the reference and it can be seen that the scheduling space is covered much more evenly than by the initial test signal.

4.4 Refined identification

The new dataset based on the refined test signal will be used for identification. For validation, the datasets D2 and D3 are used. The reidentification has been initialized with the initial model. Fig. 5 shows the exemplary model evaluations on the dataset D3. In the top plot, the system's output with the two fuzzy model evaluations is shown. In the bottom plot, the error is plotted. It can be seen that the refined model's error is far more consistently around zero whereas the initial model's error also shows peaks. The error measures (NRMSE) on both datasets (table 2) show that the refined model performs significantly better on both validation data sets.

models	D2	D3
initial	0.2792	0.1257
refined	0.0499	0.0304

Table 2. NRMSE for model evaluation on datasets D2 and D3

5. SUMMARY AND OUTLOOK

In this contribution, two problems were investigated: The first result is that a fuzzy superposition of locally generated control variables can be used to approximately force a nonlinear system to follow a desired output. It was demonstrated that the errors that result from the superposition remain small under the assumption that neighboring local models are similar enough in order to reduce the error following the superposition. This is plausible if the modeled system is not ill-conditioned. But the model error based on the system not being in the model class is larger. This result will be addressed in future works. Secondly, it was shown that it is useful to consider the space-filling aspect

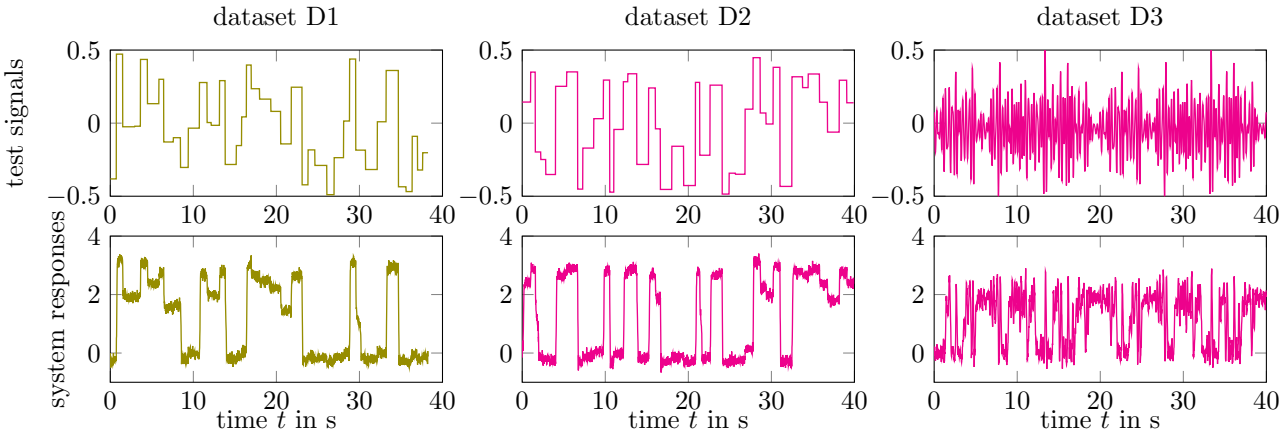

Fig. 2. Datasets used for initial identification (D1) and for validation (D2 & D3)

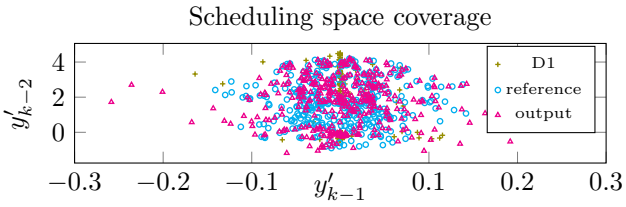

Fig. 3. Assessment of rotated scheduling space coverage

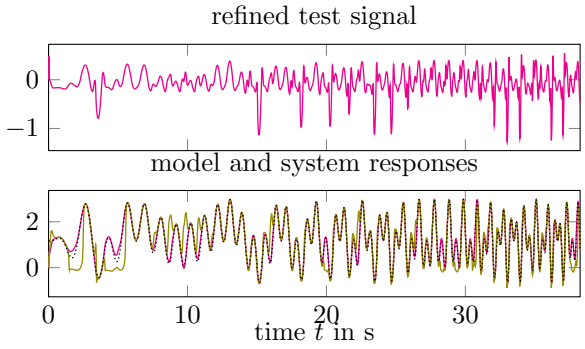

Fig. 4. Refined test signal (top) and responses (bottom): model (magenta), system (olive), reference (dashes)

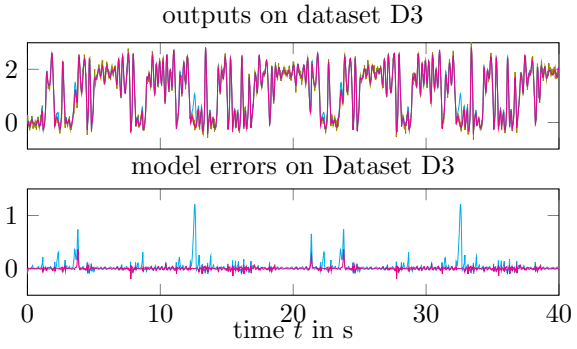

Fig. 5. Validations on dataset D3, initial model (cyan), refined model (magenta), system output (olive)

in the scheduling space which usually contains the output in case of dynamic systems. The method can be applied to other offline control variable designs like optimal control but the implementation gets more complicated if the input cannot be treated as affine and therefore is part of the scheduling variable. Then, nonlinear algebraic equations have to be solved to obtain the test signals. Future studies will include tests with real test beds.

REFERENCES

Babuška, R. (1998). *Fuzzy Modeling for Control*. Springer Netherlands.

Deflorian, M. (2011). Design of experiments for nonlinear dynamic system identification. *IFAC Proceedings Volumes*, 18, 13179–13184.

Ebert, T., Fischer, T., Belz, J., Heinz, T.O., Kampmann, G., and Nelles, O. (2015). Extended deterministic local search algorithm for maximin latin hypercube designs. In *Proc. 2015 IEEE Symposium Series on Computational Intelligence*, 375–382. Cape Town, South Africa.

Gringard, M. and Kroll, A. (2016). On the parametrization of APRBS and multisine test signals for the identification of nonlinear dynamic TS-models. In *Proc. IEEE Symposium Series of Computational Intelligence 2016*, 39–60. Athens, Greece.

Juhász, S., Legány, C., and Babos, A. (2006). Cluster validity measurement techniques. In *Proc. International Conference of Artificial Intelligence, Knowledge Engineering and Data Bases*, 388–393. Madrid, Spain.

Kroll, A. and Dürrbaum, A. (2015). On joint optimal experiment design for identifying partition and local model parameters of Takagi-Sugeno models. In *Proc. 17th IFAC Symposium on System Identification (SysID)*, 1427–1432. Beijing, China.

Li, J., Niemann, D., Wang, H., and Tanaka, K. (1999). Parallel distributed compensation for Takagi-Sugeno fuzzy models: multiobjective controller design. In *Proc. American Control Conference*, 1832–1836. San Diego, California, United States of America.

MathWorks (2019). *Optimization Toolbox User's Guide*.

Narendra, K.S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks*, 1(1), 4–27.

Zeitz, M. (2010). Differential flatness: A useful method also for linear SISO systems (in German). *at - Automatisierungstechnik*, 58(1), 5–13.